Linux Network Servers

TCP/IP Parte 1

Objetivo: Fazer uma análise de baixo pra cima (bottom-up) esclarecendo a independência das camadas físicas, rede, transporte e aplicação. Usaremos uma conversa telefônica como exemplo. Faremos as analogias entre as partes da conversa telefônica e o TCP/IP. Ao final da leitura. você entenderá o básico do protocolo tcp/ip e conhecerá os comandos básicos de rede do Linux.

Começamos analisando o dia-a-dia.

O que é comunicação?

Se entende por comunicação como o intercâmbio de informação entre sujeitos ou objetos.

E o que precisamos para que a troca de informações, ou seja, a comunicação funcione?

É necessário que haja um entendimento entre quem está enviando a informação e quem está recebendo. Por exemplo, que ambos falem a mesma língua.

E o que seriam meios de comunicação?

Refere-se ao instrumento ou à forma de conteúdo utilizados para a realização do processo de comunicação.

Por exemplo:

* Sonoro: telefone, rádio.
* Escrita: jornais, diários e revistas.
* Audiovisual: televisão, cinema.

Mas agora falando de computadores, como eles se comunicam?

Em primeiro lugar, para que haja comunicação entre os computadores, assim como entre as pessoas, eles têm que estar ligados uns aos outros por algum meio físico.

Por exemplo:

* Cabos de cobre
* Fibra óptica
* Rádio (rede sem fio)

Quando falamos com alguém diante de nós, nossa voz que é a informação trafega até a outra pessoa usando o ar. Quando um computador fala com o outro ele vai utilizar um protocolo pra que a comunicação funcione.

O que seria um protocolo?

Protocolo é uma convenção ou padrão que controla e possibilita uma conexão, comunicação ou

Linux Network Servers

transferência de dados entre dois sistemas computacionais. De maneira simples, um protocolo pode ser definido como "as regras que governam" a sintaxe, semântica e sincronização da comunicação.

Imagine uma ligação telefônica. Você quer se comunicar com outra pessoa. Quando você tira o telefone do gancho e ouve o sinal de discagem, o sistema de telefonia está te dizendo que está pronto para receber o número de destino.

Vai até o telefone e disca um número que representa o aparelho da outra pessoa. O sistema telefônico então então encontra onde está o número e a outra pessoa atende a ligação.

Quando ela atente a ligação, estamos usando os cabos nos postes e o número de telefone para ir até o destino correto.

Quando a conversa começa, a voz é usada para que as pessoas se ouçam. Mas não basta simplesmente ouvir a fala, temos que concordar com a língua.
Diante dessa conversa telefônica, podemos separá-la em 4 camadas:

1. Meio físico: fios telefônicos
2. Rede: Usamos um número para identificar com quem queremos falar
3. Transporte: Usamos nossa voz como transporte. Poderíamos utilizar código morse.
4. Aplicação: Efetivamente a informações que queremos mandar formatada em nossa língua.

Perceba que uma camada é independente da outra. Se a pessoa com quem você queira falar esteja na mesma sala que você, o meio físico seria o ar, a rede seria direcionar sua fala para onde a pessoa estiver, e o transporte ainda continuaria sendo a voz humana e a aplicação a informação formatada em nossa língua.

Voltando a falar de computadores, me deem exemplos de meios e protocolos implantandos no meio físico usados por computadores?

* Ethernet
* ADSL
* PPP (modem)

Qual dispositivo usamos para falar com uma rede Ethernet?

Placa de rede.

E qual é a suíte de protocolos que a internet utiliza?
TCP/IP

O conjunto de protocolos TCP/IP é um conjunto de protocolos de comunicação entre computadores em rede. Seu nome vem dos dois protocolos mais importantes do conjunto: o TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) e o IP (Internet Protocol - Protocolo de Interconexão).

O conjunto de protocolos pode ser visto como um modelo de camadas, onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior.

Linux Network Servers

As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação), e lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

O que é esse IP? O que ele representa?

Na suíte de protocolos para a internet, o IP executa a tarefa básica de levar pacotes de dados da origem para o destino. O protocolo IP pode transmitir dados para diferentes protocolos de níveis mais altos, esses protocolos são identificados por um único número de protocolo IP. O protocolo IP funciona na camada de rede.

Cada host (nó) recebe um número que o identifica, esse número é chamado também de IP. Imagine que é como o telefone. O IP é um número de 32 bits, separado em 4 partes compostas por 8 bits cada, que chamamos de octetos. Nós representamos esse número como 4 números decimais, de 8 bits cada, separados por um ponto.

Ex:

192.168.0.1
11000000.10101000.00000000.00000001

Convertendo um número decimal para binário:

Exemplo:
Número decimal: 192

Para converter esse número basta dividi-lo por 2 e pegando o seu resto, veja como:
192 / 2 = 96
Resto = 0

96 / 2 = 48
Resto = 0

48 / 2 = 24
Resto = 0

24 / 2 = 12
Resto = 0

12 / 2 = 6
Resto = 0

Linux Network Servers

6 / 2 = 3
Resto = 0

3 / 2 = 1
Resto = 1

1 / 2 = 0
Resto = 1

Eu pego o número de trás para frente.
Logo o número decimal 192 é em binário o número: 11000000

Fazendo o inverso (converter um número binário para decimal): 11000000

Pegue a posição do algarismo setado como 1.
Temos dois números 1 nesse número binário.
A posição começa com zero.

A posição começa com zero (da direita para esquerda).
A posição do primeiro 1 é 7;

A posição do segundo 1 é 6;

2 elevado a 7 = 128
2 elevado a 6 = 64
128 + 64 = 192

O protocolo IP (internet protocol) atua na camada de rede, e dentro dele temos outros procotolos. Um deles é o ICMP (Internet Control Message Protocol) que serve para uma básica comunicação entre hosts.

Um comando comum que utilizamos comummente é o ping.

Rode esse comando em sua máquina.

```
$ ping -c 4 localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.057 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.047 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.048 ms
```

Linux Network Servers

-c determina a quantidade de repetições.

Qual dispositivo do computador é configurado com um número IP?
Placa de rede

E como faço para identificar minha placa de rede?

No Linux as placas de redes são identificadas como ethX, onde X é um número que começa em 0. Caso você tenha apenas uma placa de rede, ela será identificada como eth0. Caso tenho duas, eth0 e eth1.

Rode o seguinte comando:

```
$ dmesg| grep eth
eth0: RealTek RTL8139 at 0xdc00, 00:e0:7d:f0:97:bf, IRQ 18
eth0:  Identified 8139 chip type 'RTL-8100B/8139D'
eth1: link down
e100: eth0: e100_watchdog: link up, 100Mbps, full-duplex
```

E como eu configuro um IP na placa de rede?

Usamos o o comando ifconfig. Temos que estar como root.

Rode o comando ifconfig.

```
eth0      Link encap:Ethernet  HWaddr 00:15:c5:32:e0:4b
  inet addr:192.168.1.100  Bcast:192.168.1.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1492  Metric:1
        RX packets:250850 errors:0 dropped:0 overruns:0 frame:0
        TX packets:167495 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:329805594 (314.5 MiB)  TX bytes:18542787 (17.6 MiB)
```

O endereço MAC (do inglês Media Access Control) é o endereço físico da estação, ou melhor, da interface de rede. Uma máscara de subrede também conhecida como subnet mask ou netmask é um número de 32 bits usada para separar em um IP a parte correspondente à rede pública, à subrede e aos hosts.

Uma subrede é uma divisão de uma rede de computadores - é a faixa de endereços lógicos reservada para uma organização. A divisão de uma rede grande em menores resulta num tráfego de rede reduzido, administração simplificada e melhor performance de rede. No IPv4 uma subrede é identificada por seu endereço base e sua máscara de subrede.

Embora aparentem ser uma coisa só, os endereços IP incluem duas informações: o endereço da rede e o endereço do host dentro dela. Em uma rede doméstica, por exemplo, você poderia utilizar os endereços "192.168.1.1", "192.168.1.2" e "192.168.1.3", onde o "192.168.1." é o endereço da rede (e por isso não muda) e o último número (1, 2 e 3) identifica os três micros que fazem parte dela.

Linux Network Servers

	Endereço decimal	Binário
Endereço completo	192.168.1.10	11000000.10101000.00000001.00001010
Máscara da subrede	255.255.255.0	11111111.11111111.11111111.00000000
Porção da rede	192.168.1.0	11000000.10101000.00000001.00000000
Broadcast	192.168.1.255	11000000.10101000.00000001.11111111

Como configurar a interface de rede?

Cuidado pois isso vai trocar o IP da interface e você pode perder conectividade!

Configurando:

```
ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

Derrumando:

```
ifconfig eth0 down
```

Subindo:

```
ifconfig eth0 up
```

Eu posso usar qualquer número IP?

As faixas de endereços começadas com "10", "192.168" ou de "172.16" até "172.31" são reservadas para uso em redes locais e por isso não são usadas na Internet. Os roteadores que compõe a grande rede são configurados para ignorar pacotes provenientes destas faixas de endereços, de forma que as inúmeras redes locais que utilizam endereços na faixa "192.168.0.x" (por exemplo) podem conviver pacificamente, sem entrar em conflito.

No caso dos endereços válidos na Internet, as regras são mais estritas. A entidade global responsável pelo registro e atribuição dos endereços é a IANA (http://www.iana.org/), que delega faixas de endereços às RIRs (Regional Internet Registries), entidades menores, que ficam responsáveis por delegar os endereços regionalmente. Nos EUA, por exemplo, a entidade responsável é a ARIN (http://www.arin.net/) e no Brasil é a LACNIC (http://www.lacnic.net/pt/). Estas entidades são diferentes das responsáveis pelo registro de domínios, como o Registro.br.

Dica de segurança!
Nunca tente usar em redes locais IPs que não sejam de faixas reservadas, você pode ter sérios problemas.

Como ter acesso a outras máquinas em outras redes? Quem nos leva até outras redes?

Um Gateway, ou porta de ligação, é uma máquina intermediária geralmente destinada a interligar redes, separar domínios de colisão, ou mesmo traduzir protocolos. Exemplos de gateway podem ser os routers (ou roteadores) e firewalls, já que ambos servem de intermediários entre o utilizador e a rede. Um proxy também pode ser interpretado como um gateway (embora em outro nível, aquele da camada em que opera), já que serve de intermediário também.

Como configuro o gateway?

```
ebl:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
```

Linux Network Servers

E se precisar trocar o gateway padrão?

Apagando:
route del default
Configurando:
route add gw 192.168.0.1

Agora temos nossa rede configurada, mas como funciona o transporte entre um computador e outro? Quais são os protocolos de transportes e suas diferenças?

Temos basicamente o TCP e o UDP.

As características fundamentais do TCP são:

- Orientado à conexão - A aplicação envia um pedido de conexão para o destino e usa a "conexão" para transferir dados;
- Ponto a ponto - uma conexão TCP é estabelecida entre dois pontos;
- Confiabilidade - O TCP usa várias técnicas para proporcionar uma entrega confiável dos pacotes de dados, que é a grande vantagem que tem em relação ao UDP, e motivo do seu uso extensivo nas redes de computadores. O TCP permite a recuperação de pacotes perdidos, a eliminação de pacotes duplicados, a recuperação de dados corrompidos, e pode recuperar a ligação em caso de problemas no sistema e na rede;
- Handshake - Mecanismo de estabelecimento e finalização de conexão a três e quatro tempos, respectivamente, o que permite a autenticação e encerramento de uma sessão completa. O TCP garante que, no final da conexão, todos os pacotes foram bem recebidos;
- Entrega ordenada - A aplicação faz a entrega ao TCP de blocos de dados com um tamanho arbitrário num fluxo (ou stream) de dados, tipicamente em octetos. O TCP parte estes dados em segmentos de tamanho especificado pelo valor MTU. Porém, a circulação dos pacotes ao longo da rede (utilizando um protocolo de encaminhamento, na camada inferior, como o IP) pode fazer com que os pacotes não cheguem ordenados. O TCP garante a reconstrução do stream no destinatário mediante os números de sequência.

No TCP, os dados são transmitidos através de conexões.

Esquema de uma comunicação entre um cliente e um servidor:

Cliente: Envia um SYN para solicitar a abertura da conexão;
Servidor: Envia um SYN para confirmar o recebimento e avisar que a porta está disponível;
Servidor: Envia um ACK para iniciar a conexão;

Linux Network Servers

Cliente: Envia um ACK para confirmar a conexão;
Cliente: Envia um DATA que contém o pacote;
Servidor: Envia um OK para confirmação, depois de analisar a integridade do pacote;
Cliente: Envia um FYN para solicitar o fechamento da conexão;
Servidor: Envia um FYN para confirmar;
Cliente: Envia um FYN para confirmar que recebeu a confirmação;

As características fundamentais do UDP são:

O protocolo UDP não é confiável. Caso garantias sejam necessárias, é preciso implementar uma série de estruturas de controle, tais como timeouts, retransmissões, acknowlegments, controle de fluxo, etc. Cada datagrama UDP tem um tamanho e pode ser considerado como um registro indivisível, diferentemente do TCP, que é um protocolo orientado a fluxos de bytes sem inicio e sem fim.

Também dizemos que o UDP é um serviço sem conexão, pois não há necessidade de manter um relacionamento longo entre cliente e o servidor. Assim, um cliente UDP pode criar um socket, enviar um datagrama para um servidor e imediatamente enviar outro datagrama com o mesmo socket para um servidor diferente. Da mesma forma, um servidor poderia ler datagramas vindos de diversos clientes, usando um único socket.

O que são portas?

Imagine que as duas partes do endereço IP (a parte referente à rede e a parte referente ao host) correspondem ao CEP da rua e ao número do prédio. Um carteiro só precisa destas duas informações para entregar uma carta. Mas, dentro do prédio moram várias pessoas. O CEP e número do prédio só vão fazer a carta chegar até a portaria. Daí em diante é preciso saber o número do apartamento. É aqui que entram as famosas portas TCP. Existem 65.536 portas TCP, numeradas de 0 a 65535. Cada porta pode ser usada por um programa ou serviço diferente, de forma que em teoria poderíamos ter até 65536 serviços diferentes ativos simultaneamente em um mesmo servidor, com um único endereço IP válido. O endereço IP contém o CEP da rua e o número do prédio, enquanto a porta TCP determina a que sala dentro do prédio a carta se destina.
Relação de serviços e portas:

http: 80
https: 443
smtp: 25
ssh: 22
telnet: 23

Dica!
Consulte o arquivo /etc/services

Quanto temos um programa que precisa deixar uma porta sempre aberta, dizemos que ele está "ouvindo", do inglês listing.

Dica de segurança!
Deixar daemons ativados que manteem portas abertas é um prato cheio para CRACKERS. Procure manter somente o que é estritamente necessário.

Linux Network Servers

Como saber quais são as portas abertas em minha máquina? E as conexões ativas?

Usamos o comando netstat.

```
ebl:~# netstat -pntl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:139           0.0.0.0:*               LISTEN      3089/smbd
tcp        0      0 192.168.1.100:139       0.0.0.0:*               LISTEN      3089/smbd
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      2996/apache2
tcp        0      0 10.0.1.1:53             0.0.0.0:*               LISTEN      2984/named
tcp        0      0 192.168.1.100:53        0.0.0.0:*               LISTEN      2984/named
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2742/sshd
```

Parâmetros do netstat:

* -a all
* -t tcp
* -u udp
* -n somente números
* -p mostrar o processo responsável
* -l mostra o que está em estado listen (ouvindo)