Administração de Redes de Computadores

Firewall (Pacotes/Aplicação)

Alexssandro C. Antunes

(Alexssandro.Antunes@unisul.br)

Foco

- Discutir pontos relevantes sobre segurança de sites que utilizam o sistema operacional Linux em seus servidores.

Site

- Qualquer organização que possua computadores ou recursos relacionados a rede “ativos”(routers, links, servidores de acesso) conectados a Internet.
- Iremos assumir que um site possui autoridade de definir suas próprias políticas de segurança com o conhecimento e colaboração de quem efetivamente detém propriedade sobre estes recursos.

Perigos - Protocolos TCP/IP

- Packet Sniffing
  - Espiando o tráfego da rede.
- Spoofing e source routing
  - Fazendo-se passar por outra máquina.
- Tráfego falso injetado na conexão
  - Previsão de “sequence number” ->opção do TCP.
- Ataques “denial of service”.

Packet Sniffing

- Necessário acesso a uma máquina com conexão física a rede.
- Requer placa de rede em modo promíscuo.

Spoofing e Source Routing

- Métodos para fazer-se passar por outros hosts com pacotes TCP/IP forjados.
- Controle de acesso baseado em nome ou endereço pode ser comprometido.
- Filtros no roteador e via IPChains/IPTables/IPFwAdm podem bloquear tentativas mais óbvias.
- Pacotes “source routed” devem ser filtrados no roteador.

Seqüestro de seção e Previsão de “sequence number”

- Os números seqüenciais controlam as seções TCP.
- Segurança depende dos números iniciais (ISN) serem imprevisíveis.
- Sniffing ou testes podem tentar determinar os ISN.

Firewall

- Estrutura que previne o “fogo de se espalhar”.
- Internet Firewall
  - evita que chamas do “inferno da arquitetura internet” atinjam a sua LAN (rede local) privada.
  - Mantém os membros de sua LAN “puros” ao negar a estes acesso as tentações da internet.

Firewalls e Filtros de Pacotes

- O que é um firewall?
  - Uma ferramenta ou conjunto de ferramentas de segurança.
  - Modelo de rede formado por vários elementos.
  - Firewalls bem configurados podem fortalecer a segurança da rede.
  - Pode manter os de fora para fora e também os de dentro para dentro.

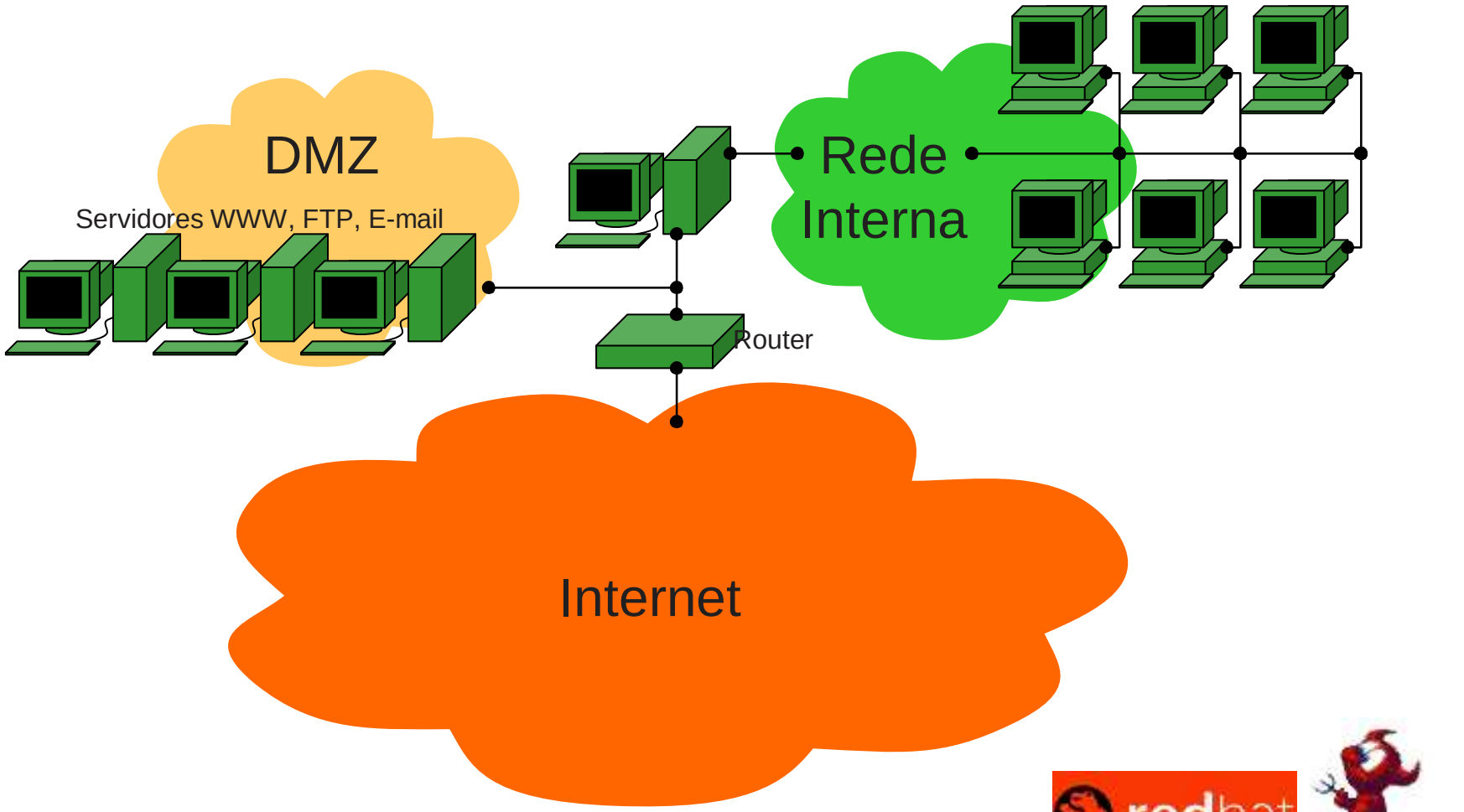
Firewalls e Filtros de Pacotes

- O que NÃO é um firewall?
  - Firewall não garante a segurança.
    - Não pode monitorar tráfego que não passa através dele.
  - Firewall não autentica dados.
    - Proteção menor contra o spoofing.
  - Firewall não garante integridade dos dados.
    - Não protege contra vírus e cavalos-de-tróia.
  - O fato de um produto ser chamado de firewall não garante que ele realmente seja um.

Elementos de um Firewall

- Duas ou mais redes distintas.
- Zona desmilitarizada.
- Router “untrusted” com filtros de pacotes.
- Máquinas “untrusted” provendo serviços expostos.
- Serviços de proxy por “bastion hosts”.

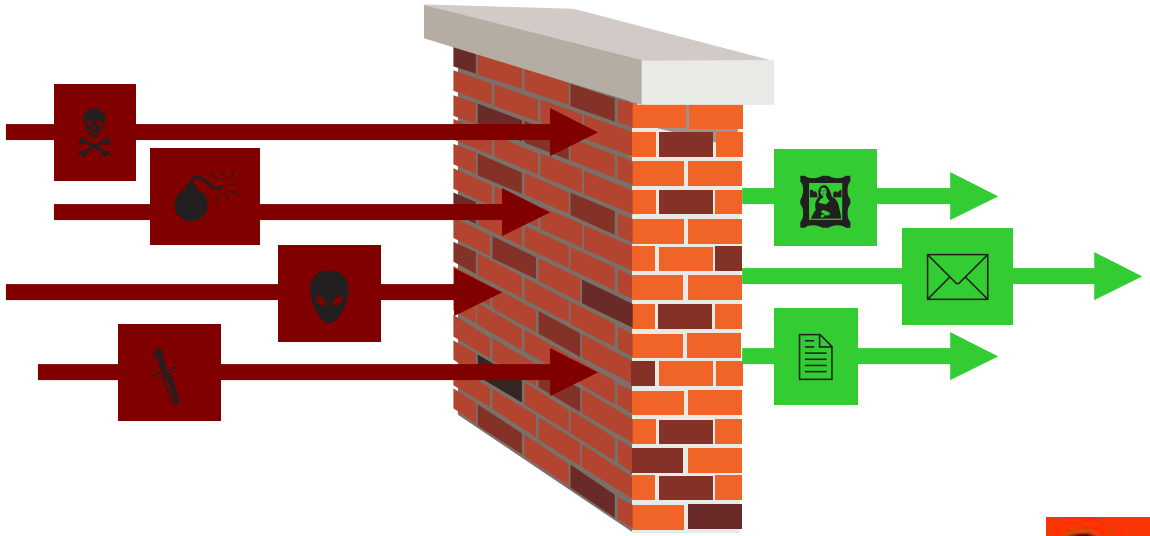
Elementos de um Firewall



Linux como parte de um Firewall

- Proxy (Squid)
- Filtro de pacotes
- NAT

} IPFwAdm/IPTables/IPChains



Tipos de Firewall

- Firewall de filtragem (packet filtering firewall)
  - boqueia pacotes selecionados de rede.

- Servidores Proxy (Proxy server firewall)
  - faz a conexão de rede para o usuário final.

Packet Filtering Firewall

- Funcionam no nível de camada de rede.
- Permite que o dado saia (ou entre) no sistema se as **regras** permitirem.
- Filtragem se dá pelo tipo de pacote (tcp, ip), endereço fonte e destino, e portas.
- Muitos roteadores agem como *firewalls* de filtragem de pacotes
  - "Lista de Controle Acesso - ACl´s".

Packet Filtering Firewall

- Não disponibilizam controle de senha.
- Não identifica usuários (o único identificador é o endereço IP da máquina na qual está).
- Transparente para o usuário final. Ele não precisa configurar regras em seus próprios aplicativos (configuração do Netscape, por exemplo).
- “Redirecionamento de tráfego”.

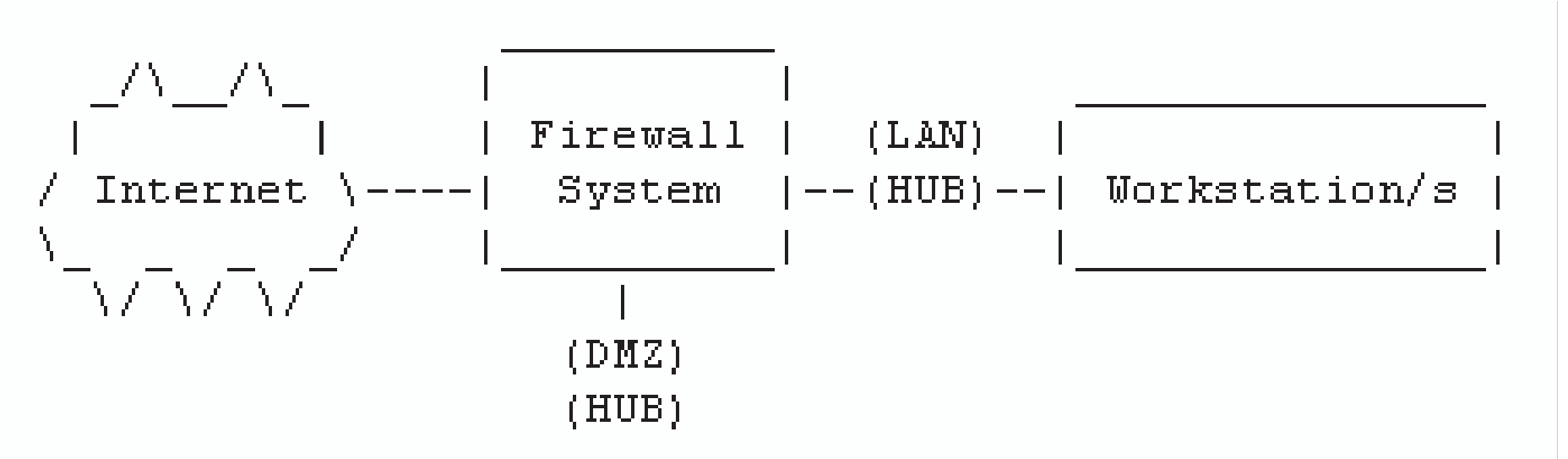
Servidor Proxy (proxy server firewall)

- Não permite roteamento de pacotes até o usuário final. Todo dado de chegada é manipulado pelo firewall por programas chamados proxies que cuidam da passagem dos dados.
- Exemplo: De “A” faz-se telnet para “B” e, de lá, para “C”. O processo pode ser automatizado com um proxy server.

Servidor Proxy (proxy server firewall)

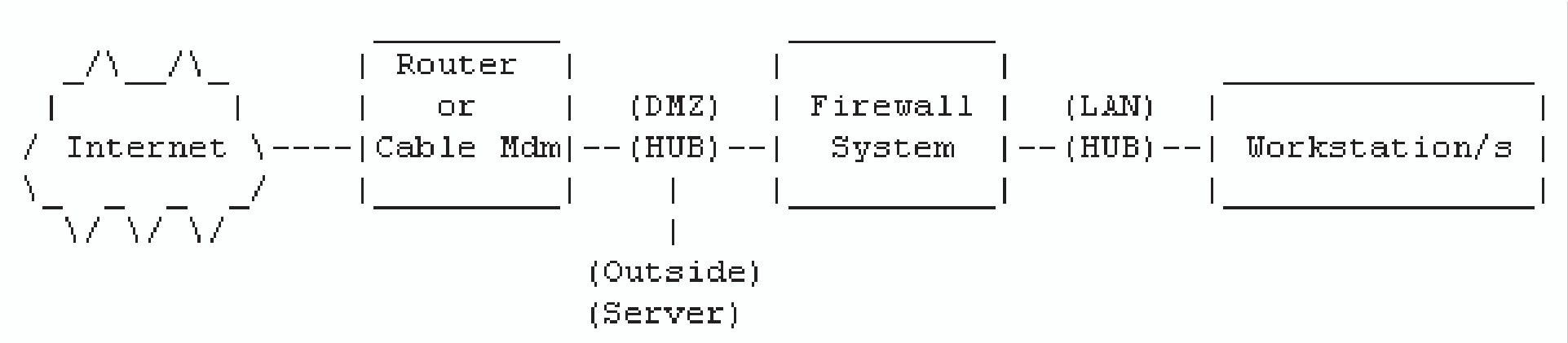
- O cliente telnet em “A” manda a solicitação para um servidor proxy telnet que, por sua vez, faz a solicitação em nome do cliente e retorna a informação a este.
- Esse tipo de servidor proxy é conhecido como proxy de *aplicação*.
  - Squid é um servidor proxy para HTTP com capacidade de filtrar URLs e palavras “impróprias”.

Arquitetura de um Firewall



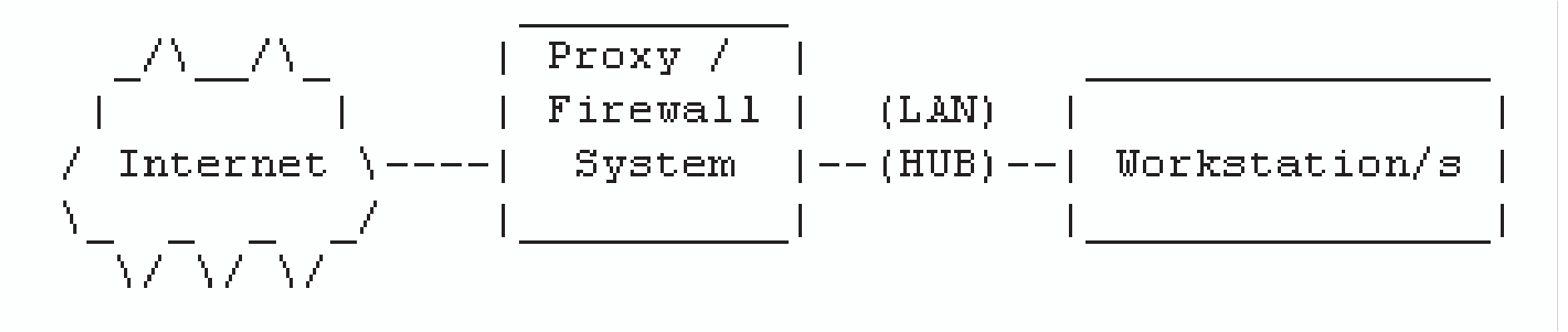
- Arquitetura tipo Dial-up.
- DMZ: Demilitarised Zone (onde ficam os servidor WEB, ftp, e-mail,etc)

Arquitetura firewall com Roteador



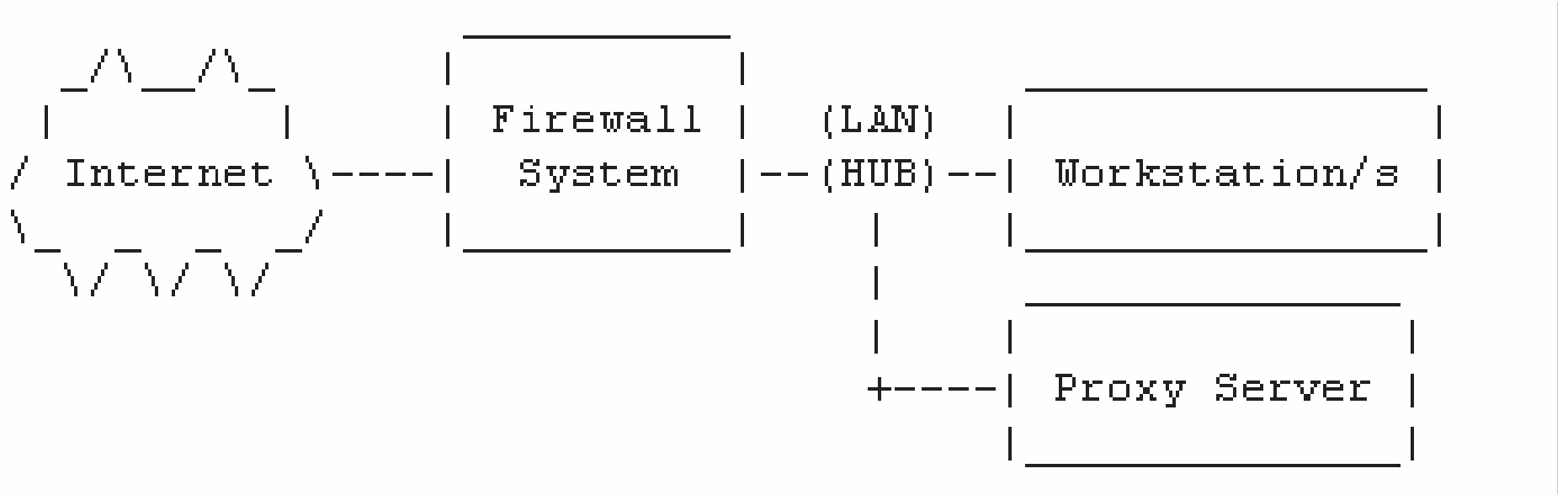
- Há um roteador na sua rede (que pode conter regras de firewall).
- Se não tiver, faça o seu firewall com suas regras.

Firewall com servidor proxy



- Integração de um firewall e servidor proxy. Assim é possível manter um “log” de todas as preferências “acessos” dos usuários.

Firewall com servidor proxy (2)



- Configure regras no firewall de forma a somente aceitar conexões com o mundo externo pelo proxy server. Assim, o usuário somente poderá acessar a Internet via proxy.
- Configuração “recomendada”.

Configuração do Kernel (Linux)

- É preciso ter o fonte do linux instalado.
- /usr/src/linux.
- make menuconfig.
- Selecionar algumas opções para firewall.

Configurações do kernel para firewall

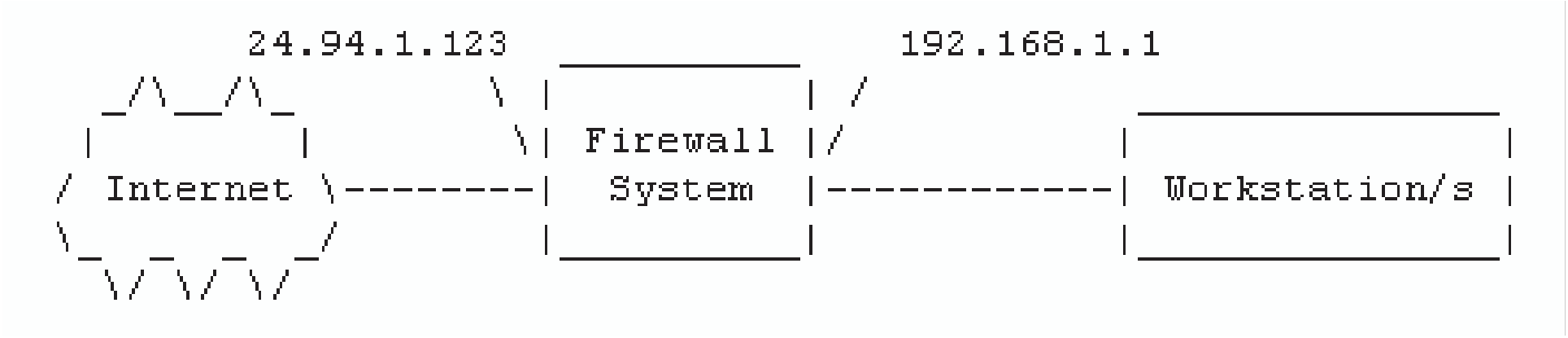
<*> Packet socket
    [ ] Kernel/User netlink socket
    [*] Network firewalls
    [ ] Socket Filtering
    <*> Unix domain sockets
    [*] TCP/IP networking
    [ ] IP: multicasting
    [*] IP: advanced router
    [ ] IP: kernel level autoconfiguration
    [*] IP: firewalling
    [?] IP: always defragment (required for masquerading)
    [?] IP: transparent proxy support
    [?] IP: masquerading

--- Protocol-specific masquerading support will be built as modules.
[?] IP: ICMP masquerading
--- Protocol-specific masquerading support will be built as modules.
[ ] IP: masquerading special modules support
[*] IP: optimize as router not host
< > IP: tunneling
< > IP: GRE tunnels over IP
[?] IP: aliasing support
[*] IP: TCP syncookie support (not enabled per default)
--- (it is safe to leave these untouched)
< > IP: Reverse ARP
[*] IP: Allow large windows (not recommended if <16Mb of memory)
< > The IPv6 protocol (EXPERIMENTAL)

Configurando as duas interfaces

- Coloque a sua LAN “dentro” do firewall, atribuindo as máquinas IPs inválidos.
- Assim, a rede de fora não enxergará a sua LAN (rede local).
- Esse “mascaramento” de IPs é o que chamamos de *IP Masquerading*.

Arquitetura possível




```
#route -n
Kernel routing table
Destination     Gateway         Genmask         Flags MSS    Window Use Iface
24.94.1.0       *               255.255.255.0   U     1500   0        15 eth0
192.168.1.0     *               255.255.255.0   U     1500   0         0 eth1
127.0.0.0       *               255.0.0.0       U     3584   0         2 lo
default         24.94.1.123     *               UG    1500   0        72 eth0
```

Após configurar as interfaces

- De dentro da LAN:
  - pingar todas as máquinas pertencentes a ela.
  - não pingar o IP (24.94.1.123). Se conseguir é porque a opção de masquerade ou ip-forward está habilitada (/proc/sys/net/ipv4/ip_forward = 1).
- Da sua máquina firewall
  - pingar qualquer endereço da Internet.

IPCHAINS - Firewall Linux

- Atuam em nível de kernel.
- ipchains -V / man ipchains.
- REGRAS
  - ACCEPT: aceita o pacote.
  - REJECT: descarta, devolvendo um destination unreachable -> “inalcançavel”.
  - DENY: apenas descarta!

Regras do Firewall

- Quando um pacote se encaixa em uma regra com alguma destas determinações, esta é seguida e as demais desprezadas, liberando o pacote da verificação restante do *firewall.*
- *chain:* um conjunto de regras (cadeia).
- Regra: algo como “Se o cabeçalho do pacote se parecer com isso, então eis o que deve ser feito com ele.”

Políticas das cadeias

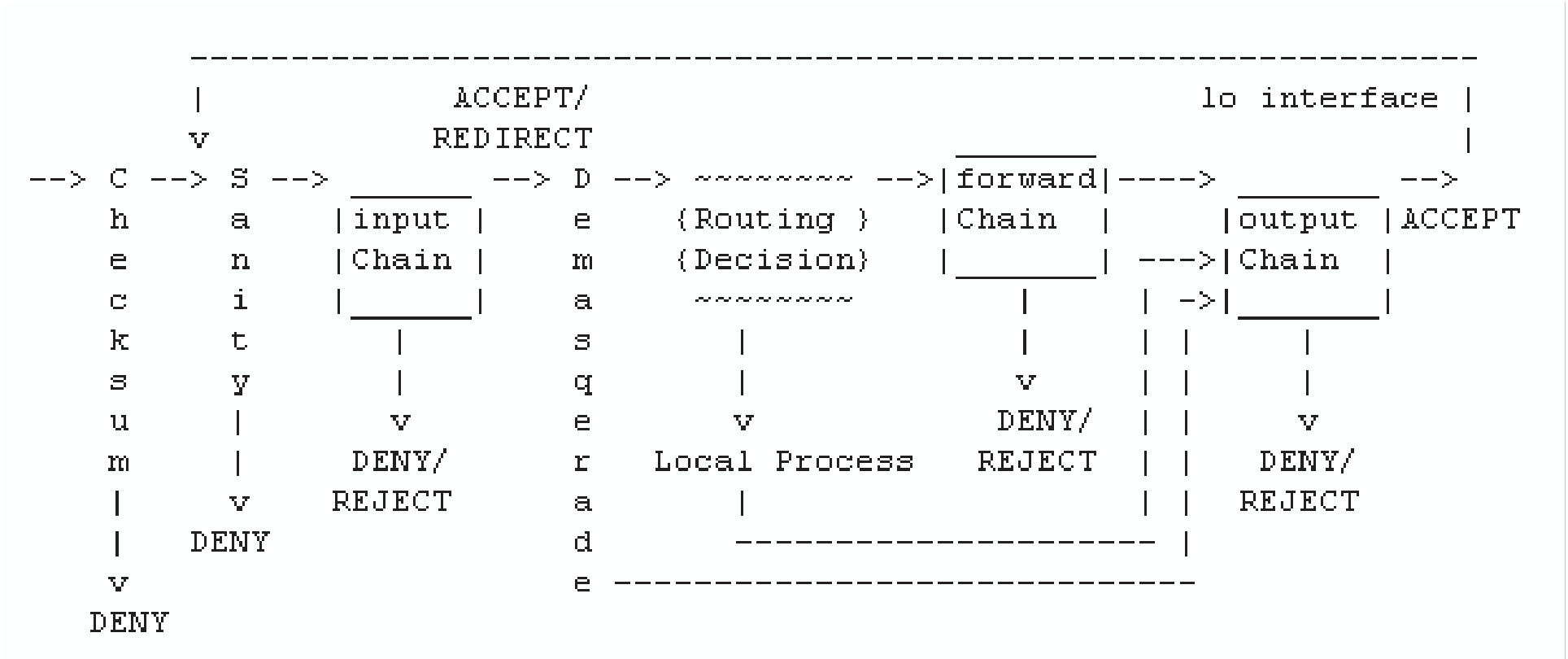
- Qual o destino de um pacote IP se ele não se encaixa em nenhuma das regras das cadeias?
  - Pacote liberado (ACCEPT).

Construção de Regras

```
firewall   add      accept    tcp
           del      deny      udp     from origem porta   to destino porta   via interf
                    reject    icmp
                              all
```

Firewall : comando para executar o firewall (**`ipchains`**).
add/deny: adiciona ou remove uma regra;
accept/deny/reject: destino do pacote;
tcp/udp/icmp/all: tipo de pacote (**all** representa todos);
from origem porta: origem do pacote e porta.
(a porta não tem sentido em pacotes do tipo **icmp** e **all**);
to destino porta: destino do pacote e porta .
(a porta não tem sentido em pacotes do tipo **icmp** e **all**);
via interf: por qual interface de rede.

Entendendo as regras



Pacote chegando em uma máquina ...

Significado

- Checksum: testa se o pacote foi corrompido.
- Sanity: antes de cada *chain*, faz verificação de sanidade. *Checksum* é uma maneira.
- *input chain*: Primeira cadeia a ser testada ao entrar no firewall. Continua se o veredicto não for *Deny* ou *Reject*.

Significado (cont.)

- *Demasquerade*: Se o pacote é um *reply* a um outro previamente mascarado, ele será "desmascarado", indo direto para a *output chain*.
- Se não estiver usando *masquerade*, apague mentalmente essa etapa do diagrama.

Significado (cont.)

- Routing decision: Campo destino examinado pelo código de roteamento p/ entregá-lo a um processo local ou repassá-lo (*forward*) a uma máquina remota.

Significado (cont.)

- *Local process*: Um pacote que roda em uma máquina pode receber pacotes após o roteamento e pode enviar pacotes [que passará pelo do processo de roteamento e irá através da *output chain*].

Significado (cont.)

- Interface loopback (*lo interface*): se um pacote de um processo local destina-se a outro processo local, este irá através da output chain pela interface lo e retornará através da input chain pela mesma lo. Isso garante o teste das regras internamente.

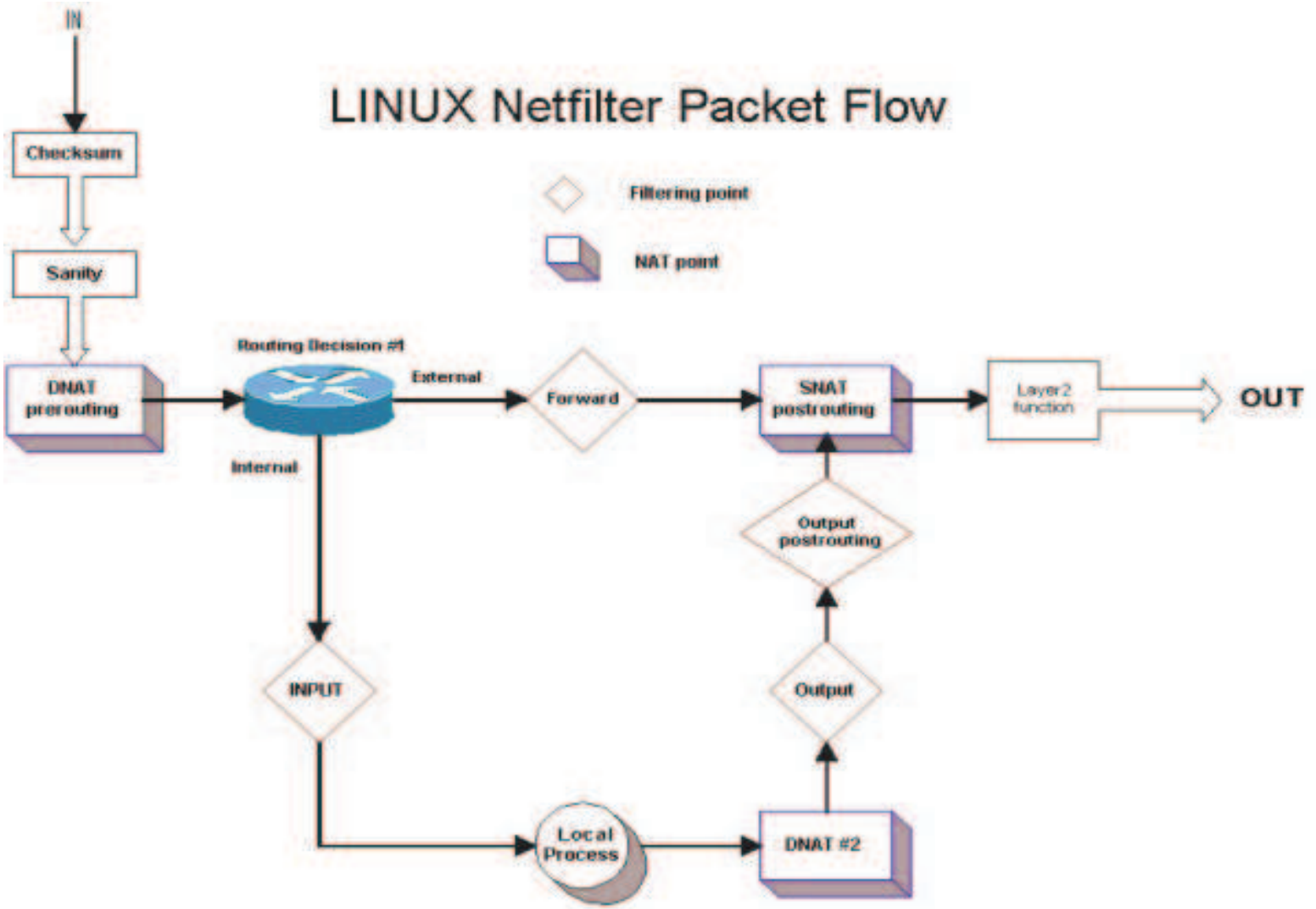
Significado (cont.)

- Local: se o pacote não foi criado para um processo local, verifica-se a *forward chain*. Caso contrário, o pacote irá para a *output chain*.
- *Forward chain*: regra percorrida por todos pacotes que passam por uma máquina e são destinadas a outras.

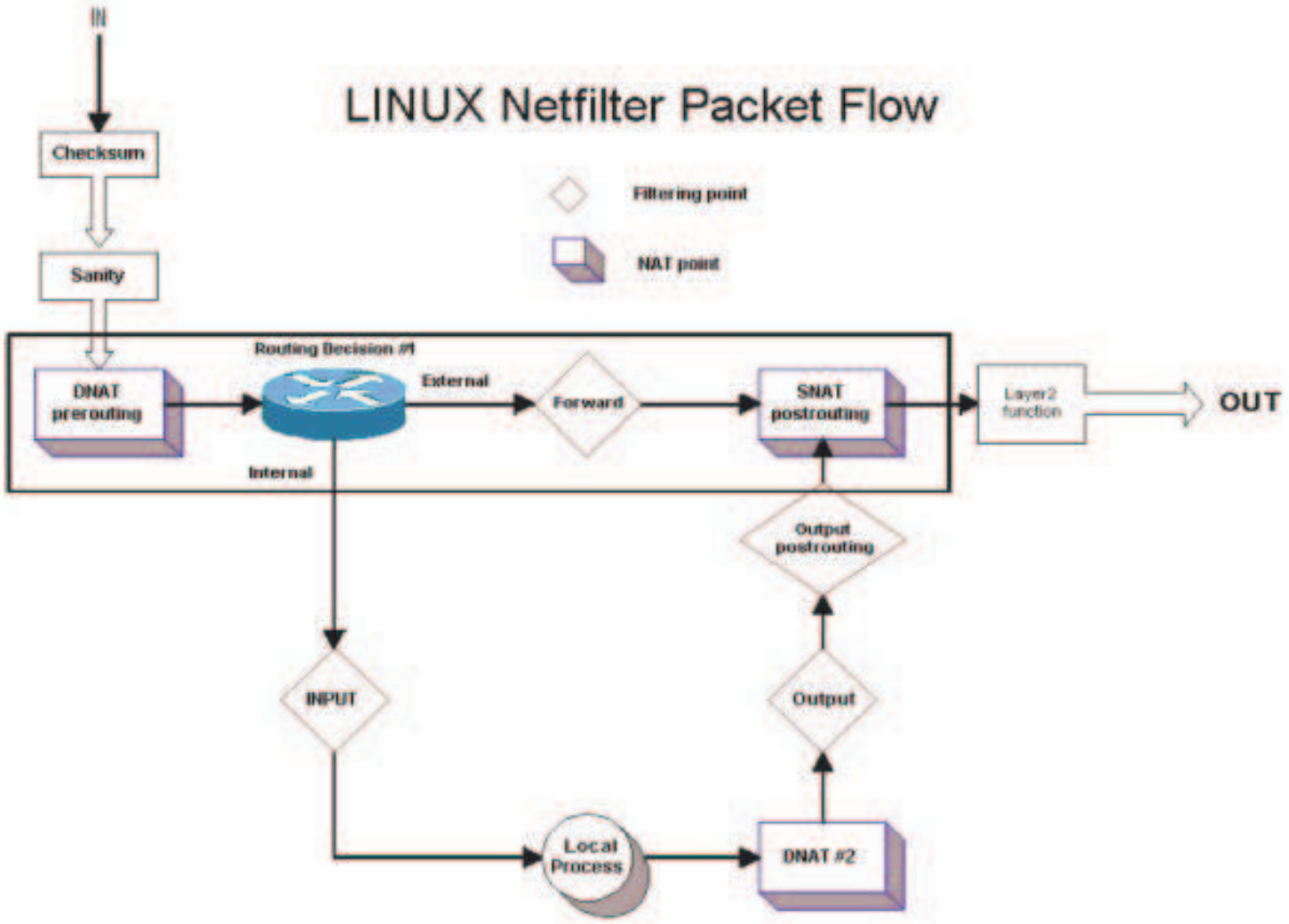
Significado (cont.)

- *Output chain*: regra percorrida por qualquer pacote imediatamente antes de ser mandado para fora.

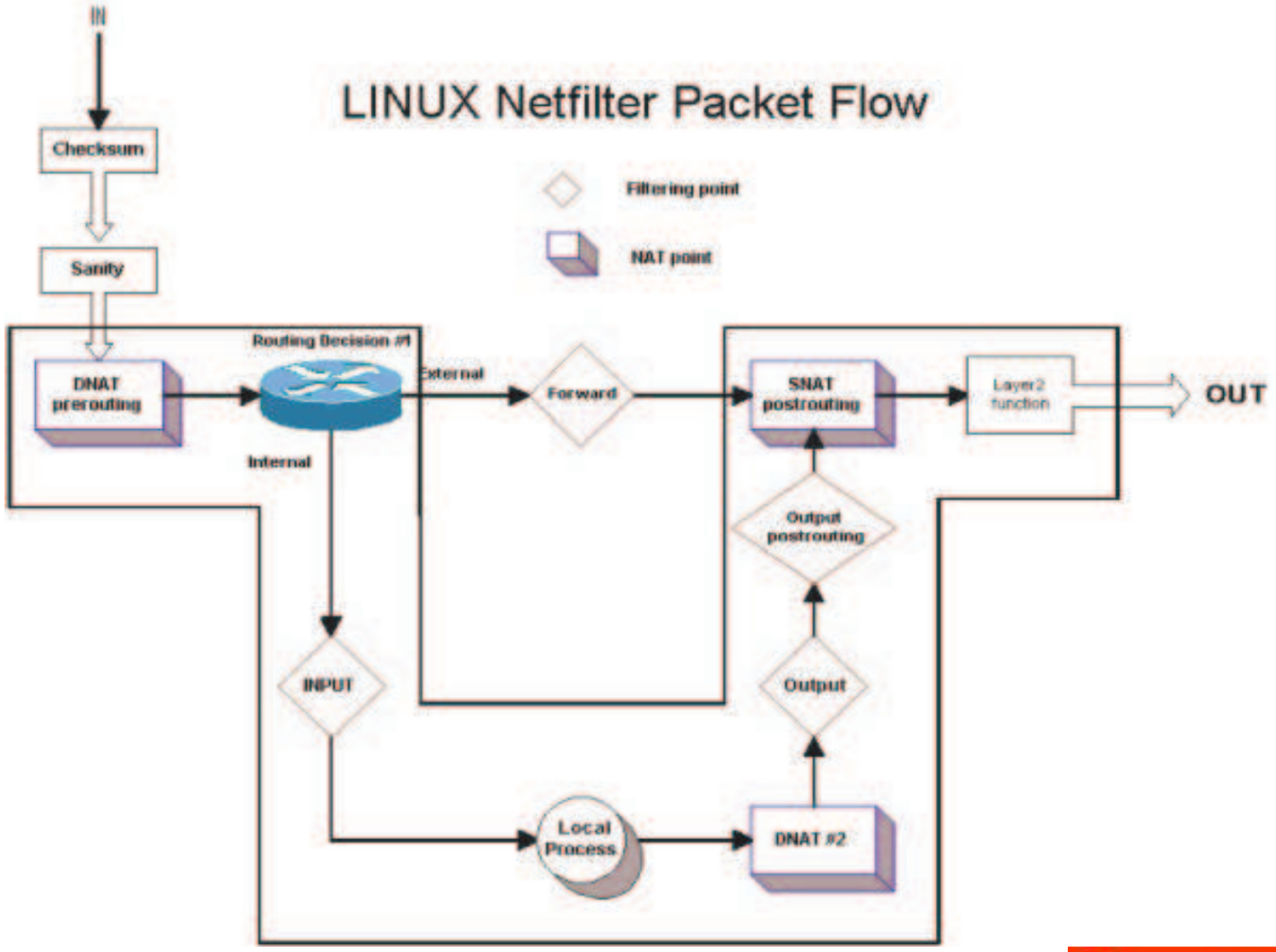
Lógica NetFilter



Forward chain



Input/Output chain



Sintaxe ipchains

Comandos

1. Cria uma nova cadeia (-N).

2. Apaga uma cadeia vazia (-X).

3. Muda a política de uma cadeia existente. (-P).

4. ***Listas as regras em uma cadeia (-L). (-v) -> mostra detalhes.***

5. ***Força as regras para fora de uma cadeia (-F).***

6. Zera o contador de bytes e pacotes de todas as regras da cadeia (-Z).

Sintaxe ipchains

7. ***Adiciona uma nova regra à uma cadeia (-A).***

8. Insere uma nova regra em uma certa posição da cadeia (-I).

9. Substitui uma regra em uma certa posição da cadeia (-R).

10. ***Apaga uma regra em uma certa posição da cadeia (-D).***

11. Apaga a primeira regra que bate com as da cadeia (-D).

12- Lista as conexões do tipo masquerade (-M -S).

13 - Seta os valores de time-out para masquerade (-M -S).

Sintaxe ipchains

Chaves

input - de entrada

output - de saída

forward - encaminhamento de pacotes

Sintaxe ipchains

Opções - Destino

-s -> origem ACCEPT - aceita

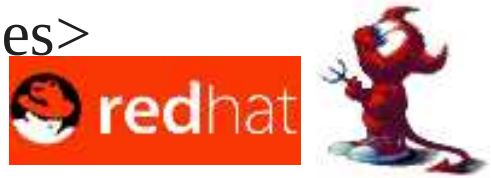
-p -> protocolo DENY - nega

-d -> destino MASQ - máscara

-i -> interface REDIRECT - redireciona

-j -> jump - salta para próxima regra

ipchains <comando> <chave> <destino> <opções>



A opção -j (jump to)

- Especifica o que fazer com o pacote
- E se não for especificado -j ?
  - ipchains -A input -s 192.168.1.1
- Políticas:
  - ACCEPT, DENY, REJECT
  - MASQ (somente com *forward chain*)
  - REDIRECT
  - RETURN

MASQ,REDIRECT,RETURN

- MASQ: kernel deve mascarar o pacote
  - kernel compilado com essa opção.
- REDIRECT: kernel deve enviar o pacote para uma porta local específica ao invés de para onde ele originalmente deveria ir (TCP e UDP).
- RETURN: leva ao final das regras de uma cadeia.

Sintaxe Ipchains - Exemplos

Listando as regras

ipchains -L

Ressetando todas as regras

ipchains -F output

ipchains -F forward

ipchains -F input

Sintaxe Ipchains - Exemplos

Negando tudo

ipchains -p output DENY

ipchains -p input DENY

ipchains -p forward DENY

Sintaxe Ipchains/Iptables - Exemplos

Negando o protocolo

ipchains -A input -p icmp -s 0/0 -d 0/0 -j DENY -l

opção -l -> loga o comando no /var/log/message

iptables –A INPUT -p icmp –s 0/0 –d 0/0 –j DROP

Negando Hosts

ipchains -A input -j DENY -s <endereço URL ou ip> -l

ipchains -A output -s ´www.naopermitido.com.br' -j DENY

iptables –A INPUT –s 'www.naopermitido.com.br' –j DROP

Sintaxe Ipchains/Iptables - Exemplos

Negando porta ftp (21), ssh (22) e telnet (23)

ipchains -A output -p tcp -s 0.0.0.0/0 21:23 -j DENY -i eth1

opção -i-> interface de rede

iptables -A INPUT -p tcp -s 0/0 -i eth1 --dport 21:23 -j DROP

Negando um intervalo de portas irc (6660 até 6667)

ipchains -A output -p tcp -s 0.0.0.0/0 6660:6667 -j DENY

iptables -A FORWARD -p tcp --dport 6660:6667 -j REJECT

Sintaxe Ipchains/Iptables - Exemplos

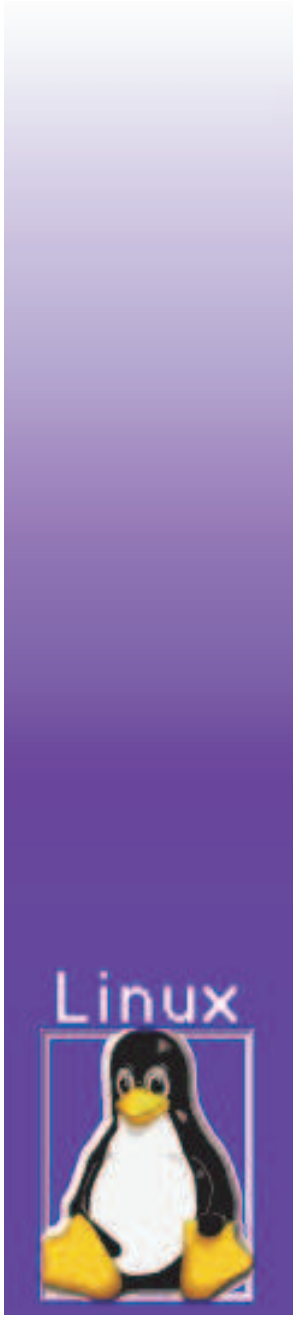
Redirecionando tráfego da porta (80) para (8080) -> proxy -> squid!

ipchains -A input -p tcp -s 10.9.1.0/24 -d 0/0 80 -j REDIRECT 8080 -i eth1

#habilita o encaminhamento de pacotes por mascaramento

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -t nat -A PREROUTING -s 10.9.1.0/24 -p tcp --dport 80 -j REDIRECT --to-port 8080



Squid

Proxy / Cache

Proxy / Cache

- Proxy
  - um agente que tem autorização para agir em nome de outro.
- Cache
  - local “disfarçado” em disco para se preservar e esconder provisões (dados) que são inconvenientes para se transportar pela rede internet.

O que é o *Squid* ?

- Agente que aceita solicitações de clientes (browsers) e as passa aos servidores apropriados.
- Armazena uma cópia num cache de disco local.
- Seu benefício só é sentido se o mesmo dado “endereço” é requisitado várias vezes.

Squid

- É um programa proxy **HTTP** que faz cache (*caching proxy*), pois armazena os dados acessados “visitados”.
- Quais dados?
  - páginas HTML, sons, imagens (objetos).
- Faz filtragem, mas não pode ser considerado como um sistema *firewall*.

Squid - Protocolos

- Desde que as requisições sejam enviadas por clientes via HTTP, Squid suporta (além de HTTP, obviamente):
  - FTP
  - SSL (Secure Socket Layer)

Comunicação inter-cache

- Caches Squid “conversam” via UDP através dos protocolos:
  - HTTP: recuperar cópias de objetos de outros caches.
  - ICP: Internet Cache Protocol. Tenta descobrir se um determinado objeto está em um determinado cache.
  - SNMP: Simple Network Management Protocol.
  - Cache Digest. Recupera tabela de índices de objetos de outros caches.

Comunicação inter-cache

- Por que?
  - Base de usuário. Quanto maior a ‘user base’, maior a ‘hit rate’ (taxa de acerto). Grandes bases são possíveis se os caches cooperam.
  - Redução de carga local.
  - Espaço em disco e memória. Cache usam muito disco e/ou memória RAM.

Instalação

- Requisitos de hardware (em ordem decrescente de importância)
  - Disk Random Seek Time
  - Memória RAM
  - Throughput de disco sustentável
  - Poder de CPU

Quantidade de Disco

- Suponha um cache pessoal, com 1GB disco.
- Navegação diária: 10 MB dados.
  - 100 dias para encher o cache

  ➡ taxa de entrada influencia quantidade de disco a alocar.
- Decida a qtd de disco segundo a qtd de dados que vai passar pelo cache por dia..

RAM

- Para otimizar busca, Squid utiliza memória para guardar tabelas do tipo Look-up.
- Cada objeto no disco consome cerca de 75 bytes de índice em RAM.
- Média de um objeto na internet 13 Kbytes. Supondo 1 GB cache -> 80000 objetos -> requerem 6 MB RAM.
- Um cache razoável (8 GB) requer 48 MB RAM !

Setup do Sistema

- Criar usuário e grupo squid -> “Default”.
- Verificar permissão de diretórios
  - ...../bin/squid; ../etc/squid; .../var/spool/squid; .../var/log/squid; .....
- Executando ‘squid -z’, cria-se o cache “default (/var/spool/squid)”
- Devido às permissões isso pode falhar.

Configuração

- /etc/squid/squid.conf

- porta http: http_port 8080
- local de cache: **cache_dir /var/spool/squid 100 16 256**
  - 100 MB de cache; 16 diretórios, cada qual com 256 sub-diretórios

Configuração

- Usuários e grupos

  cache_effective_user squid
  cache_effective_group squid

- Lista de controle de Acesso
  - importante configurar.
  - Lembre-se: por default, tudo é negado !!!!

Proxy Transparente

- Clientes não precisam configurar o browser.
  - Redirecionamento de porta via regra “ipchains e /ou iptables”.
- Squid se encarrega de interceptar os pacotes e colocá-los no cache.
- Setup é transparente, mas sua utilização não o é.
  - Há mascaramento de IPs. O IP origem do pacote será mudado !

Proxy Transparente

- Tipo de setup em que o squid roda na máquina que também é o gateway primário.
- É preciso:
  - configurar Kernel para transparência: redirecionar conexões porta 80 para a porta que o squid escuta (http_port 8080).
  - configurar squid (veja documentação).

Atuais versões

Versões do Squid em distribuição atualmente

Squid 1.1.x

Utilizada há quase dois anos, é a segunda geração do software (a primeira é a 1.0.x). Provê alguns recursos avançados de configuração de hierarquias, mas sua tendência é ser inteiramente substituída pela versão 2.x devido aos grandes avanços em áreas como gerenciamento de memória, comunicação entre servidores, entre outros.

Squid 2.x

Primeiro *release* lançado em outubro/1998, apresenta avanços como: suporte a HTTP 1.1 (*persistent connections*), I/O de disco assíncrono (usando *pthreads*), melhor utilização de memória, uso de cache digest para comunicação entre servidores, gerenciamento via snmp, mensagens de erro customizáveis, entre outros.

Instalação do Squid

Preparação do Sistema Operacional

Criação de usuário

O squid deve ser executado por um usuário comum (ex: *user*: **nobody**, *group*: **nogroup**). O mais recomendado é a criação de um usuário específico para essa finalidade (ex: *user*: **squid**, *group*: **squid**)

Configuração do Sistema Operacional

Deve-se ficar atento para qualquer limitação do sistema operacional que possa impedir o usuário **squid** de realizar suas tarefas. Entre as limitações mais comumente encontradas estão: número máximo de conexões simultâneas, número de processos simultâneos, quantidade de memória alocada por processo, etc.

Resultado da instalação (após *make all; make install*)

$home/bin/RunAccel : script utilizado para iniciar o squid no modo *accelerator*

$home/bin/RunCache : script utilizado para iniciar o squid no modo *cache server*

$home/bin/cachemgr.cgi : cgi usado para coletar estatísticas geradas pelo squid (deve ser instalado em um web server)

$home/bin/client : cliente HTTP (*URL retriever*)

$home/bin/dnsserver : usado pelo squid resolver nomes (fqdn)

$home/bin/squid : squid propriamente dito

$home/bin/unlinkd : usado pelo para apagar arquivos sob demanda

Resultado da instalação

$home/etc/errors : diretório com os arquivos (html) de mensagens de erros

$home/etc/icons : ícones usados pelo squid para listagens de diretórios

$home/etc/mib.txt : MIB (snmp) usada para monitorar o squid

$home/etc/mime.conf : associa extensões de arquivos a tipos *mime*

$home/etc/squid.conf : arquivo de configuração do squid (controle de acesso, hierarquia, etc)

$home/logs/ : diretório onde serão armazenados os logs de acesso, erro, etc.

Configuração do Squid

squid.conf

Configuração do servidor é realizada através do *squid.conf*

- Quantidade de memória destinada ao armazenamento de objetos
- Tamanho do disco usado para cache
- Timeout de conexões
- Opções de segurança como: permissões de acesso, logs e ident
- Personalização das mensagens de erro
- Configuração do agente snmp (comunidade, permissões etc)
- Configuração de hierarquia (quem são os pais e irmãos)

squid.conf

Configuração da quantidade de memória (RAM) usada para objetos

cache_mem *bytes*

bytes: quantidade de memória reservada para armazenar objetos em trânsito, *hot-cache* e *negative cache*

Memória para hot-cache = quanto mais melhor. Na prática, entretanto, deve-se respeitar os limites do sistema, evitando assim o uso de swap. Geralmente calcula-se:

cache_mem = Total_RAM - (squid_meta_data + outros_processos)

squid.conf

Tamanho máximo dos objetos gravados em disco

maximum_object_size *bytes*

bytes: tamanho máximo do objeto que será armazenado no cache

maximum_object_size grande = maior byte Hit Ratio

maximum_object_size pequeno = menor ganho em largura de banda, mas possível melhora na velocidade (tempo de resposta)

Na prática, recomenda-se valores de entre 4 e 16 Mb, dependendo do papel do proxy na hierarquia e da disponibilidade de disco.

squid.conf

Arquivos de log:

cache_access_log *file*
Registra todas as requisições recebidas (hora, tamanho, tempo, hit/miss etc)

cache_log *file*
Informações sobre o estado do servidor, mensagens de erro, etc.

cache_store_log *file*
Registra a entrada e saída de objetos no cache

cache_swap_log *file*
Usado pelo squid para gravar os meta-dados dos objetos gravados no disco. Esses dados são usados sempre que o squid é iniciado.

squid.conf

Diretório usado para armazenar os objetos do cache:

cache_dir *directory Mbytes level1 level2*
cache_dir *directory Mbytes level1 level2*
(...)

directory: diretório do disco onde ficarão os objetos do cache.
Mbytes: espaço máximo que pode ser ocupado por esse arquivos
level1: número de sub-diretórios que serão criados dentro de *directory*
level2: número de sub-diretórios dentro do *level1*. É no *level2* que os objetos são efetivamente gravados.

Se for especificada mais de uma cláusula **cache_dir**, o squid faz a distribuição do cache entre as várias partições.

squid.conf: segurança

Definindo uma lista de acesso:

acl *aclname acltype value*

aclname: nome qualquer para identificar a lista de acesso. Esse nome será usado quando se for referencia a lista.

acltype value: tipo da lista (cada tipo requer um valor num formato específico):

- **src** *client_ip_addr/netmask ...*
- **dst** *server_ip_addr/netmask ...*
- **srcdomain** *client_ip_name ...*
- **dstdomain** *urlserver_name ...*
- **srcdom_regex** *client_name ...*
- **dstdom_regex** *urlserver_name ...*
- **time** [{**S**|**M**|**T**|**W**|**H**|**F**|**A**}] [*hh*:*mm*-*hh*:*mm*]

acltype value:

- **url_regex** *urlregex ...*
- **urlpath_regex** *pathregex ...*
- **port** *portrange ...*
- **proto** *protocol ...*
- **method** *methodname ...*
- **browser** *regexp ...*
- **user** *username ...*
- **src_as** *number ...*
- **dst_as** *number ...*
- **proxy_auth** [*refresh*]

Exemplos:
acl localhost src 127.0.0.1/255.255.255.255
acl all src 0.0.0.0/0.0.0.0
acl ok_ports port 80-85 443 563

squid.conf: segurança

Permitindo e negando acessos:

http_access {**allow**|**deny**} *aclname*
icp_access {**allow**|**deny**} *aclname*
miss_access {**allow**|**deny**} *aclname*

aclname: nome referente a uma lista de acesso anteriormente criada

Exemplo - squid.conf

```
http_port 8080
icp_port 3128
cache_mem  8 MB
maximum_object_size 4096 KB
cache_dir ufs /var/spool/squid 200 16 256
cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log
pid_filename /var/log/squid/squid.pid
```

```
#servidor proxy com autenticação (programa autenticador
   do squid - ncsa)
    auth_param basic program /usr/bin/ncsa_auth /etc/squid/squid_passwd

#auth_param basic program <uncomment and complete this
   line>
    auth_param basic children 5

#auth_param basic realm Squid proxy-caching web server
    auth_param basic realm Digite o seu login
    auth_param basic credentialsttl 2 hours
    auth_param basic casesensitive off
```

```
#Defaults:
     acl all src 0.0.0.0/0.0.0.0
     acl manager proto cache_object
     acl localhost src 127.0.0.1/255.255.255.255

#Definicao da rede local
     acl our_network1 src 192.168.1.0/255.255.255.0
     acl our_network2 src 192.168.2.0/255.255.255.0
     acl SSL_ports port 443 563
     acl Safe_ports port 80 443 563 70 210 1025-65535

#programa autenticador e definição da acl p/ proxy c/ autenticação
    acl password proxy_auth REQUIRED
    acl squid_password proxy_auth "/etc/squid/squid_passwd"

#Listas de IP's (Adm) com acesso a qualquer URL
    acl ip_dos_admins src "/etc/squid/ip_dos_admins"

#Regras do usuario para acesso a paginas e downloads
    acl negado url_regex "/etc/squid/negado.txt”
    acl download_http urlpath_regex "/etc/squid/file_types”

#Regra do usuario para acesso a um determinado horario
    acl hora time MTWHF 08:00-18:00
acl CONNECT method CONNECT
```

```
#Regras do usuario para acesso a paginas e downloads
http_access allow ip_dos_admins
http_access deny negado
http_access deny download_http

#Pagina redirecionada para as url´s nao permitidas
deny_info err_page_name negado
deny_info err_page_name download_http

http_access allow manager localhost

#Restricoes de acesso aplicadas a rede local
http_access allow localhost
http_access allow squid_password
http_access allow our_network1 hora
http_access allow our_network2 hora
#http_access allow our_network password

http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
```

```
icp_access allow all
miss_access allow all

# Memória livre?
memory_pools off

# Aplicando o Modo ‘Acelerado’
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Obrigado.