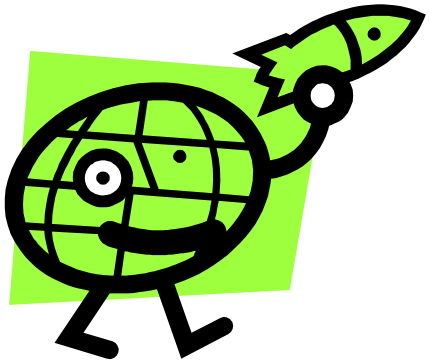
ADMINISTRAÇÃO DE REDES DE COMPUTADORES

TCP/IP FIREWALLS

Engª de Sistemas e Informática
Licenciatura em Informática

UALG/FCT/DEEI 2005/2006

Os atacantes



- **Crackers**
- **Hackers**
- **Script-kiddies**

Principais ataques

- **Trojan Horse**

É um programa ou script que se esconde num programa autorizado. Por exemplo o virús "Love Bug" Maio de 2000.

Algumas distribuições de Linux, por exemplo Red Hat utilizam técnicas de encriptografia Pretty Good Privacy (PGP) e/ checksum.

- **Back Door**

Código adicionado a um programa que permite o acesso ao sistema. Por um exemplo um programa pode ter sido alterado para enviar e-mails com determinado tipo de informação sempre que o programa é executado.

Red Hat Package Manager (RPM) ou Debian Package Management System, podem correr um programa que verificar se um pacote de software foi alterado e indicar mudanças ocorridas em ficheiros individuais.

Principais ataques

- **Trusted Horse**

Permite a um atacante fazer login sem password, porque o seu login e host estão incluídas nos ficheiros apropriados: /etc/hosts.equiv and .rhosts
Os comandos que utilizam esses ficheiros são por ex: rlogin, rsh,rcp. Para evitar esse problema muitos administradores cancelam estes comandos e usam Secure Shell SSH.

- **Buffer Overflow**

Faz com **que um programa leia dados que não lhe são destinados.**

- **Scanning ou Sniffing**

Captura de Pacotes IP de modo a obter os seus endereços IP e portos de modo determinar quais os serviços disponíveis nesse sistema.

Principais ataques

- **Spoofing**

Processo de imitar outro utilizador da rede. Por exemplo pode ser usado para enviar e-mails para um utilizador como se fossem provenientes de uma máquina de confiança.

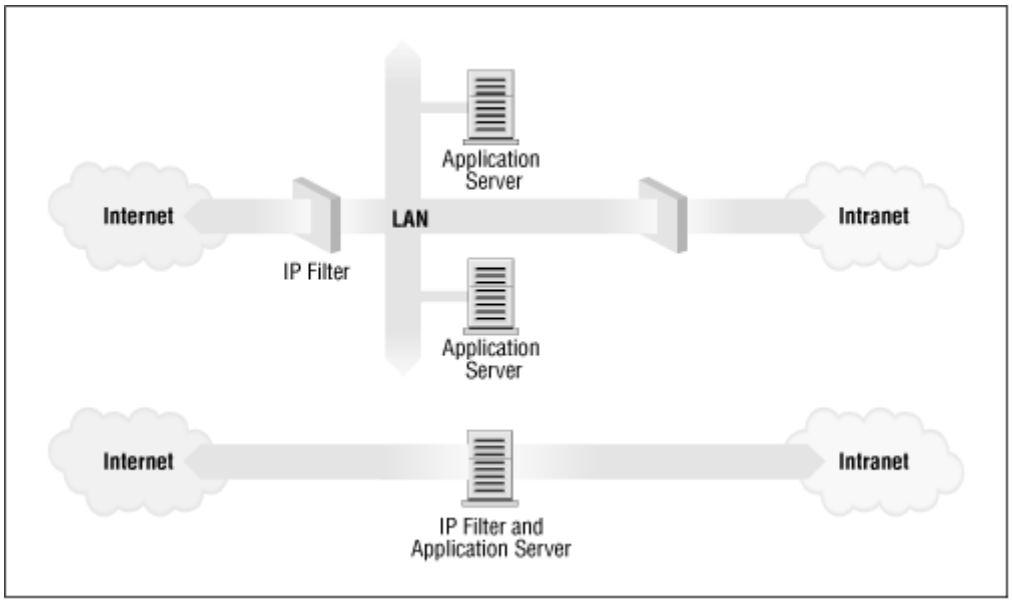
O Kernel do Linux pode ser configurado para efectuar Source Address Verification.

- **Denial of Service**

O atacante faz com que uma aplicação fique muito ocupada de modo a não poder responder aos pedidos legítimos.

- **Password Cracking**

Arquitecturas de rede com Firewalls



Como funciona uma Firewall

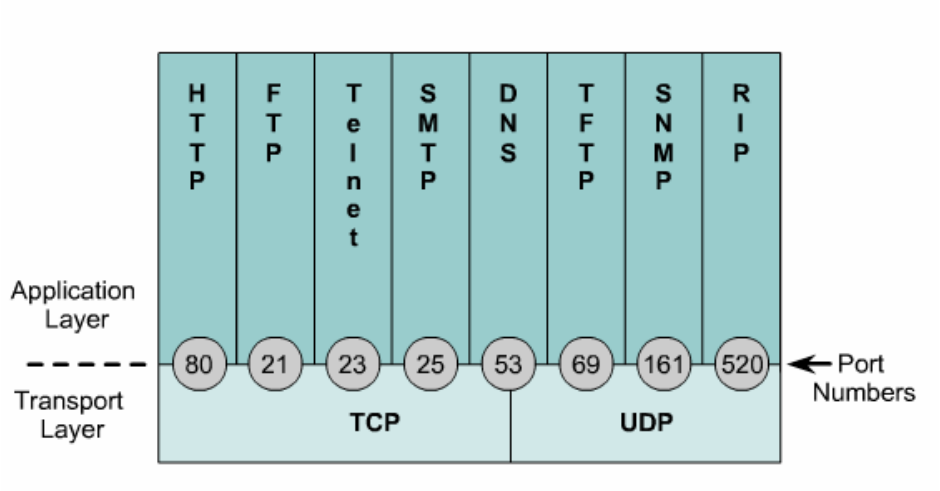
Filtra o tráfego ao nível das camadas de ligação e/ou rede e/ou transporte

-Interfaçe de rede de entrada ou saída
-Endereço IP de origem ou de destino
-Tipo de protocolo: TCP, UDP, ICMP, etc.
-Porta de origem ou destino (para TCP/UDP)
-Tipo de dados: SYN/ACK, data, ICMP Echo request, etc.

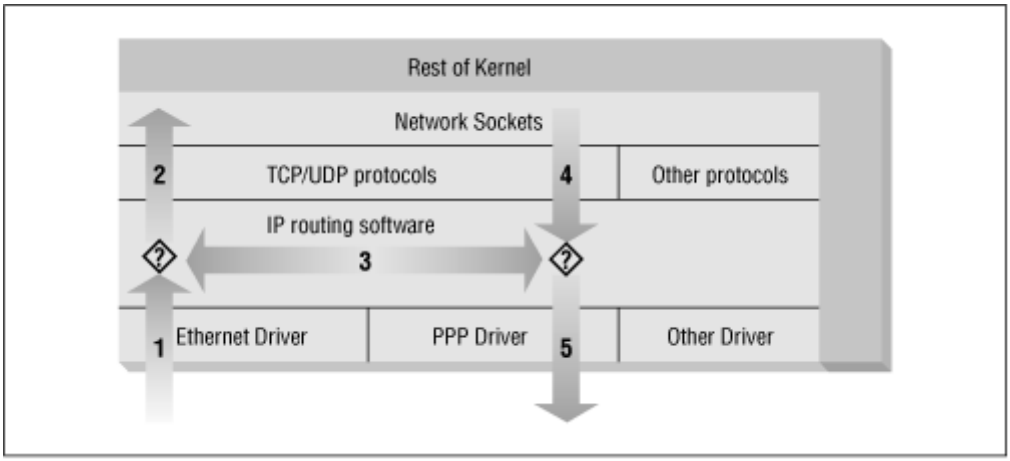
Riscos associados a uma Firewall

- É inútil quando o tráfego gerado não desejado é gerado na rede interna.
- É inutil contra IP spoofing.
- É inútil quando o seriço não permitido é oferecido noutra porta (que não está bloqueada).
- Oferece uma segurança falsa se não estiver bem configurada.

Riscos associados a uma Firewall



Filtragem de pacotes IP



Comandos

-A chain

Append one or more rules to the end of the nominated chain. If a hostname is supplied as either source or destination and it resolves to more than one IP address, a rule will be added for each address.

-I chain rulenum

Insert one or more rules to the start of the nominated chain. Again, if a hostname is supplied in t he rule specification, a rule will be added for each of the addresses it resolves to.

-D chain

Delete one or more rules from the specified chain that matches the rule specification.

-D chain rulenum

Delete the rule residing at position *rulenum* in the specified chain. Rule positions start at one for the first rule in the chain.

-R chain rulenum

Replace the rule residing at position *rulenum* in the specific chain with the supplied rule specification.

-C chain

Check the datagram described by the rule specification against the specific chain. This command will return a message describing how the datagram was processed by the chain. This is very useful for testing your firewall configuration.

Comandos

-L [chain]

List the rules of the specified chain, or for all chains if no chain is specified.

-F [chain]

Flush the rules of the specified chain, or for all chains if no chain is specified.

-Z [chain]

Zero the datagram and byte counters for all rules of the specified chain, or for all chains if no chain is specified.

-N chain

Create a new chain with the specified name. A chain of the same name must not already exist. This is how user-defined chains are created.

-X [chain]

Delete the specified user-defined chain, or all user-defined chains if no chain is specified. For this command to be successful, there must be no references to the specified chain from any other rules chain.

-P chain policy

Set the default policy of the specified chain to the specified policy. Valid firewalling policies are `ACCEPT`, `DENY`, `REJECT`, `REDIR`, or `RETURN`. `ACCEPT`, `DENY`, and `REJECT` have the same meanings as those for the tradition IP firewall implementation. `REDIR` specifies that the datagram should be transparently redirected to a port on the firewall host. The `RETURN` target causes the IP firewall code to return to the Firewall Chain that called the one containing this rule and continues starting at the rule after the calling rule.

-p [!]protocol
Specifies the protocol of the datagram that will match this rule. Valid protocol names are `tcp`, `udp`, `icmp`, or `all`. You may also specify a protocol number here to match other protocols. For example, you might use `4` to match the `ipip` encapsulation protocol. If the `!` is supplied, the rule is negated and the datagram will match any protocol other than the protocol specified. If this parameter isn't supplied, it will default to `all`.
-s [!]address[/mask] [!] [port]
Specifies the source address and port of the datagram that will match this rule. The address may be supplied as a hostname, a network name, or an IP address. The optional `mask` is the netmask to use and may be supplied either in the traditional form (e.g., /255.255.255.0) or the modern form (e.g., /24). The optional `port` specifies the TCP or UDP port, or the ICMP datagram type that will match. You may supply a port specification only if you've supplied the *-p* parameter with one of the `tcp`, `udp`, or `icmp` protocols. Ports may be specified as a range by specifying the upper and lower limits of the range with a colon as a delimiter. For example, `20:25` described all of the ports numbered from 20 up to and including 25. Again, the `!` character may be used to negate the values.
-d [!]address[/mask] [!] [port]
Specifies the destination address and port of the datagram that will match this rule. The coding of this parameter is the same as that of the *-s* parameter.
-j target
Specifies the action to take when this rule matches. You can think of this parameter as meaning "jump to." Valid targets are `ACCEPT`, `DENY`, `REJECT`, `REDIR`, and `RETURN`. We described the meanings of each of these targets earlier. However, you may also specify the name of a user-defined chain where

processing will continue. If this parameter is omitted, no action is taken on matching rule datagrams at all other than to update the datagram and byte counters.
-i [!]interface-name
Specifies the interface on which the datagram was received or is to be transmitted. Again, the `!` inverts the result of the match. If the interface name ends with +, then any interface that begins with the supplied string will match. For example, `-i ppp+` would match any PPP network device and `-i ! eth+` would match all interfaces except Ethernet devices.
[!] -f
Specifies that this rule applies to everything but the first fragment of a fragmented datagram

Opções

-b
Causes the command to generate two rules. One rule matches the parameters supplied, and the other rule added matches the corresponding parameters in the reverse direction**.**

-v
Causes `ipchains` to be verbose in its output. It will supply more information.

-n

Causes `ipchains` to display IP address and ports as numbers without attempting to resolve them to their corresponding names.

-l

Enables kernel logging of matching datagrams. Any datagram that matches the rule will be logged by the kernel using its *printk()* function, which is usually handled by the `sysklogd` program and written to a log file. This is useful for making unusual datagrams visible.

-o[maxsize]

Causes the IP chains software to copy any datagrams matching the rule to the userspace "netlink" device. The maxsize argument limits the number of bytes from each datagram that are passed to the netlink device. This option is of most use to software developers, but may be exploited by software packages in the future.

Opções

-m markvalue

Causes matching datagrams to be *marked* with a value. Mark values are unsigned 32-bit numbers. In existing implementations this does nothing, but at some point in the future, it may determine how the datagram is handled by other software such as the routing code. If a markvalue begins with a + or −, the value is added or subtracted from the existing markvalue.

-t andmask xormask

Enables you to manipulate the "type of service" bits in the IP header of any datagram that matches this rule. The type of service bits are used by intelligent routers to prioritize datagrams before forwarding them. The Linux routing software is capable of this sort prioritization. The *andmask* and *xormask* represent bit masks that will be logically ANDed and ORed with the type of service bits of the datagram respectively. This is an advanced feature that is discussed in more detail in the IPCHAINS-HOWTO.

-x

Causes any numbers in the `ipchains` output to be expanded to their exact values with no rounding.

Opções

-y

Causes the rule to match any TCP datagram with the SYN bit set and the ACK and FIN bits clear. This is used to filter TCP connection requests.

Opções