Chair for Network Architectures and Services
Department of Informatics
TU München – Prof. Carle

Network Security

Chapter 9
Security Policies and Firewalls

Overview

- Introduction: What does *secure* mean?
- Firewalls
- Virtual Private Networks
- Case study: Linux Netfilter

Introduction: What does *secure* mean?

- Definition: Security Policy

  "A security policy, a specific statement of what is and is not allowed, defines the system's security." [Bishop03]

- Definition: Security Mechanisms

  "Security Mechanisms enforce the policies; their goal is to ensure that the system never enters a disallowed state." [Bishop03]

  - Examples of Security Mechanisms:
    - IPsec gateways, firewalls, SSL, …

  - A system is *secure* if, started in an allowed state, always stays in states that are allowed.

  - The policy *defines* security, the security mechanisms *enforce* it.

Security Components

Security has three components [Bishop03]

- Requirements
  - Define security goals
  - E.g. Confidentiality, Integrity, Accountability, Availability, Controlled Access
  - "*What do we want?*"
- Policy
  - Rules to implement the requirements
  - "*How to get there?*"
- Mechanisms
  - Enforce the policy
  - E.g. firewall rules, IPsec SPD entries, …

Example

- A network admin reports:
  "*Our management wants to ensure that, because of a recent incident, the originators of all internal eMails must now be clearly identifiable. I generated X.509 certificates for all employees and set up their mail clients to always sign their outgoing mails. Unsigned eMails are now dropped by default*"

- Security Requirements:
  Sender accountability of all internal eMails

- Security Policy:
  All eMails must be cryptographically signed

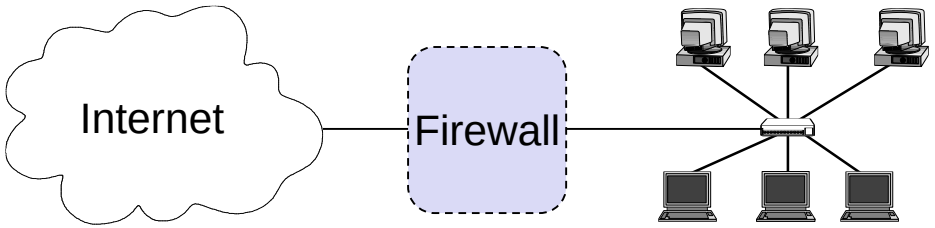
- Security Mechanisms:
  X.509 certificates + signatures, dropping of unsigned eMails by mailserver

A closer look at policy-heavy security mechanisms

Firewalls

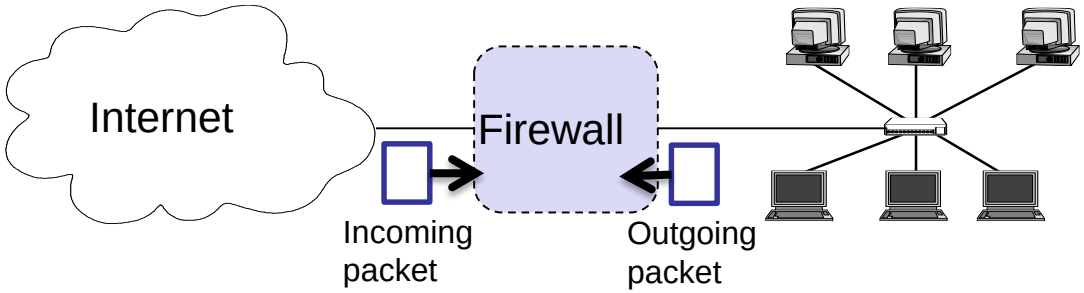
Introduction to Network Firewalls (1)

- In building construction, a firewall is designed to keep a fire from spreading from one part of the building to another
- A network firewall, however, can be better compared to a moat of a medieval castle:
  - It restricts people to entering at one carefully controlled point
  - It prevents attackers from getting close to other defenses
  - It restricts people to leaving at one carefully controlled point
- Usually, a network firewall is installed at a point where the protected subnetwork is connected to a less trusted network:
  - Example: Connection of a local area network to the Internet



  - Conceptually, firewalls realize access control on the network level

Introduction to Network Firewalls (2)



- Packets can arrive from both sides
  - *Incoming packets* from the Internet to the local network
  - *Outgoing packets* from the local network to the Internet
  - Please note:
    - The external network does not have to be the Internet, firewalls can also operate between different local networks.
    - We also say external and internal network.
- How does the firewall know what to do with the packets?
  - The firewall is configured by an administrator.
  - The administrator configures:
    - A default rule
    - Rule set

Basic strategies for default rules

- Default deny strategy: (~ Whitelisting)
  - *"Everything that is not explicitly permitted is denied"*
  - Examine the services the users of the protected network need
  - Consider the security implications of these services and how the services can be safely provided
  - Allow only those services that can be safely provided and for which there is a legitimate need
  - Deny any other service
- Default permit strategy: (~ Blacklisting)
  - *"Everything that is not explicitly forbidden is permitted"*
  - Permit every service that is not considered dangerous
  - Example:
    - Network file system (NFS) is not permitted across the firewall
    - Incoming SSH connections are only allowed to one specific host

Example: Strict Whitelisting

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	State	Action
A	Outbound	Internal	External	TCP	>1023	80	New, Estab.	Permit
B	Inbound	External	Internal	TCP	80	>1023	Established	Permit
C	Either	Any	Any	Any	Any	Any	Any	Deny

- Only allow outgoing HTTP (TCP port 80), deny the rest
- Outgoing policy
  - Only surfing the Internet (unencrypted)
- Incoming policy
  - No one can come in

What can a firewall do with a packet?

- The firewall forwards the packet.
  - This is the common operation in case the packet belongs to a flow or application that you want to allow.
  - Typical terms for this: Allow / Permit / Accept / Pass
- The firewall deletes the packet and does not forward it.
  - This is the common operation in case the packet belongs to a flow or application that you want to stop.
  - Typical terms for this: Drop / Deny / Reject
- Other options include
  - Log that a certain type of packet appeared,
  - Send error message to sender (e.g. via ICMP)
  - Rate limit, tee, mirror, inform the admin, etc.

Information that firewalls have access to

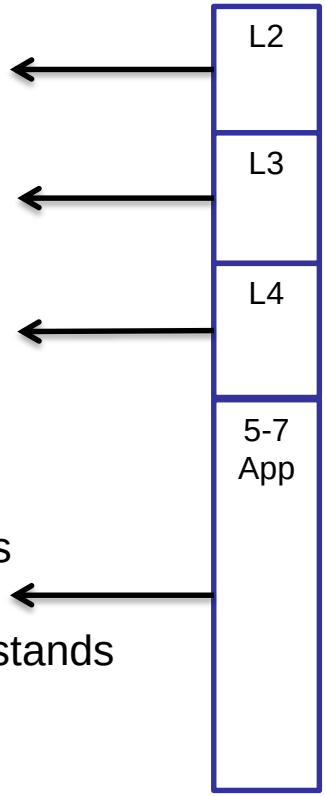
- How can a firewall gain information for its decision?
  - Without additional state-keeping, it can only use the packet and the data in its header fields.
  - It determines the entities, protocols, protocol states, and/or application to make the decision.

Link Layer: direction of packet, next physical hop

Network Layer: communication end points (entities, e.g. IPv4 addresses), transport protocol

Transport Layer: ports (applications), protocol state

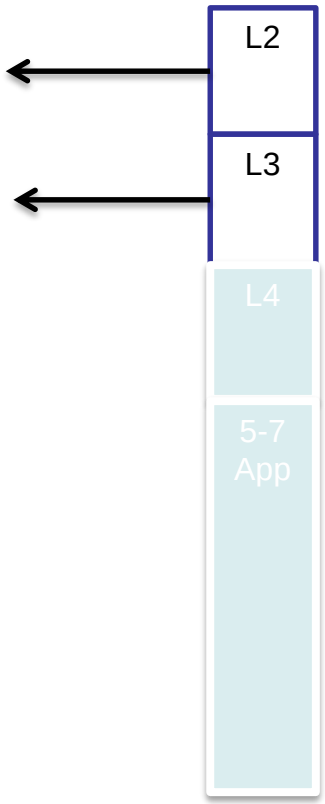
Application Layer: Application Layer Filtering requires an application level packet filter. This is not standard in firewall operation. Requires that the firewall understands the application protocol.



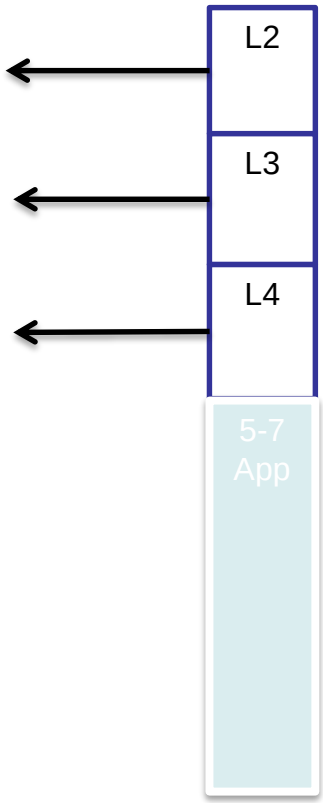
Impact of encryption (1)

- The use of encryption may prevent firewalls to access certain information in the packets.

IPSec - ESP



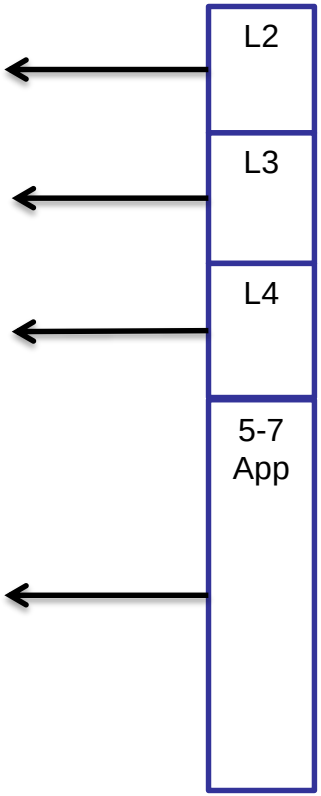
SSL / TLS



Impact of encryption (2)

- Does Link Layer (L2) encryption completely stop the firewall?

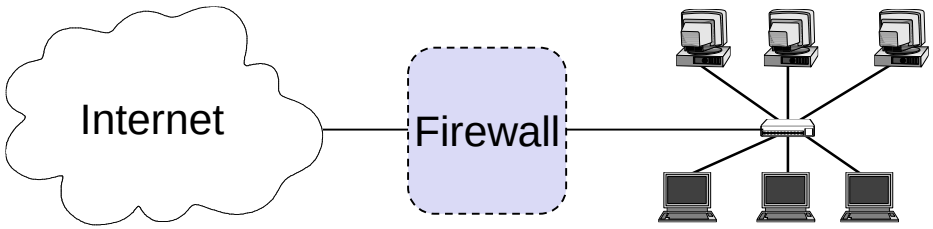
  - Not if the firewall is the L2 end point!
  - As firewalls are usually used for network layer communication to the Internet or other external TCP/IP networks, link layer communication does not pass the firewall.



More details on Protocol Header Fields for Firewalls

- Access Protocol (Link Layer):
  - Network Layer Protocol: IP, other access protocols in case of VPNs
  - Access Protocol Addresses: e.g. Ethernet MAC Address.
    - Beware: Ethernet addresses are easily spoofable
      ```
      # ifconfig eth0 hw ether de:ad:be:ef:de:ad
      ```
- IP:
  - Source address
  - Destination address
  - Flags, especially the indication of an IP fragment
  - Transport protocol type: TCP, UDP, ICMP, ...
  - Options:
    - E.g. source routing:
      - the sender explicitly specifies the route an IP packet will take
      - as this is often used for attacks most firewalls discard these packets
    - In general, IP options are rarely used

Example Policy Snippet for the IP Layer



- Prevent illegal IP addresses from the Internet
  - 192.168.0.0/16 is not routed over the public internet
    A packet incoming from the Internet with this source or destination IP address should be blocked.
    Same for other private IP ranges and unallocated IP ranges
- Prevent spoofing from the internal network
  - Assume the internal network is in the IP range 131.159.254.0/24
    Any packet on the internal interface that claims to origin from a source not in this range should be dropped
- If your network does not support IPv6, drop it
  - Or it might bypass your firewall
- Prevent unwanted IP header options
  - E.g. source routing

More details on Protocol Header Fields for Firewalls

- TCP:
  - Source Port, Destination Port:
    - Source and destination ports determine (with a limited degree of confidence) the sending / receiving application.
    - **Well-Known Ports** (0-1023): ports for well-known services like HTTP (80), DNS (53), HTTPS (443).
    - **Registered Ports** (1024-49151): for user services, some registered officially, some simply taken. IPSec NAT Traversal (4500), BitTorrent tracker (6969), …
    - **Ephemeral Ports** (49151-65535, originally 1024-65535): ports meant to be used temporarily, e.g. by clients.
  - Control Flags:
    - ACK: this bit is set in every segment but the very first one transmitted in a TCP connection, it therefore helps to identify connection requests
    - SYN: this bit is only set in the first two segments of a connection, so it can be used to identify connection confirmations
    - RST: if set this bit indicates an ungraceful close of a connection, it can be used to shut peers up without returning helpful error messages
- Application Protocol (~ Deep Packet Inspection):
  - In some cases a firewall might even need to peek into application protocol.
    - E.g. to detect applications that might hide themselves

Notation

- Specifying packet filtering rules:
  - As a packet filter protects one part of a network from another one, there is an implicit notion of the direction of traffic flow:
    - *Inbound:* The traffic is coming from an interface which is outside the protected network and its destination can be reached on an interface which is connected to the protected network
    - *Outbound:* the opposite of inbound
    - For every packet filtering rule this direction is specified as either *"inbound"*, *"outbound"*, or *"either"*
  - Source and destination address specifications can make use of wildcards, e.g. 125.26.*.* denotes all addresses starting with 125.26.
    - In our examples, we often simply denote addresses as *"internal"* or *"external"* when we want to leave exact network topology out of account
  - For source and destination ports we sometimes write ranges, e.g. ">1023"
  - We assume filtering rules to be applied in the order of specification, that means the first rule that matches a packet is applied

What can be done with packet filtering / firewalls?

In practice the following observations serve as a guide:

- Operations that require quite detailed knowledge of higher layer protocols are easier to realize in proxy systems
  - Example protocol checking: *“Let in packets bound for the DNS port, but only if they are formatted like DNS packets”*
- Operations that are simple but need to be done fast and on individual packets are easier to do in packet filtering systems

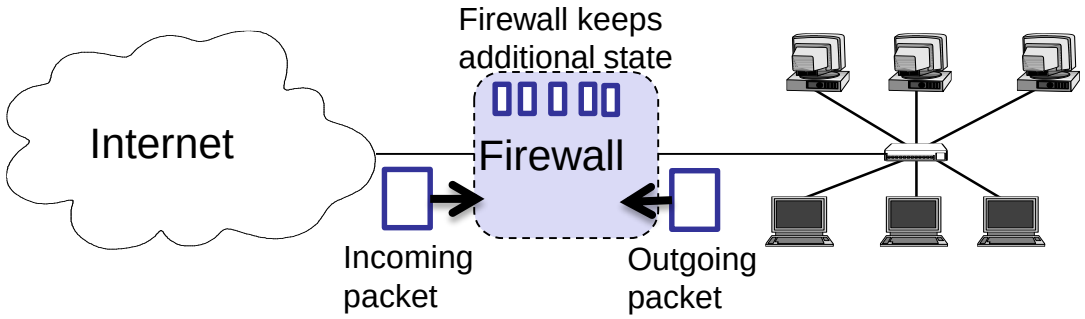
Stateless vs Stateful Packet Filtering

- In the following, we discuss and compare:

- Stateful Packet Filtering
  - Packets that arrive may generate a state in the firewall.
  - The firewall remembers having seen the packet / a packet with certain properties (e.g. the IP-5-tuple).
  - This may change the processing of subsequent packets with similar properties.
    - E.g.: *“Let incoming UDP packets through only if they are responses to outgoing UDP packets that have been observed”*

- Stateless Packet Filtering
  - The firewall only operates on the rules and each individual packet.
  - No state information is generated when processing a packet.

Stateful Filtering – Introduction (1)



- Packets that arrive may generate a state in the firewall.
  - The basic idea is to track all connections
  - E.g. track that *A* initiated a TCP connection to *B*
  - E.g. memorize that *A* sent a UDP packet to *B* in order to detect replies.
    - Warning: UDP is stateless. An attacker could exploit this, for example, sending spoofed DNS replies in the hope that you might accept one accidently as an answer to one of your original DNS queries.
  - On connection termination or after some timeout, state will be deleted.
  - Keeping state is expensive and less efficient than operating stateless
  - Danger to run out of memory

Stateful Filtering – Introduction (2)

Why do we want stateful filtering?

- Can check protocol state in the firewall.
  - E.g. "*I can establish a TCP connection to google and get replies but not the other way round.*"
- Can detect replies even in case of UDP/ICMP usage. Therefore, we can also define for UDP/ICMP who is allowed to initiate a communication.
  - E.g. "*I can ping you, you can't ping me.*"
- Possibility to react to potentially abusive behavior.
  - E.g. "*If you send me 10 messages to 10 closed ports in 1 minute, I will drop you completely for 10 minutes.*"
- If you track connections and connection requests of a host, you can also add rate limiting, etc.
  - E.g. "*Most of our clients are in Europe or America. We only allow 20 connections to port 22 per minute per IP for other countries.*"

Stateful Filtering – States

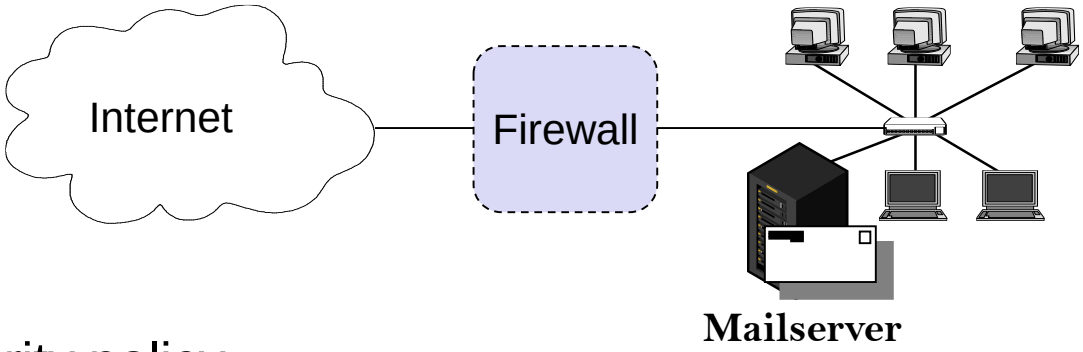
- An important feature of stateful firewalling is connection tracking.
  - The firewall sees the first packet of a connection.
  - Subsequent packets are considered to be part of this connection.

States that a connection can be in:

- New
  - This is the first packet for a yet unknown connection.
- Established
  - This is a packet for an existing connection.
- Related
  - This is a packet related to an existing connection. Application-specific configuration is needed before firewalls can recognize related packets.
  - E.g. FTP uses different ports for control connection and related data connections. Here, problem to allow a first data connection packet.
- Invalid
  - Packet contains invalid header field values.

Example: Stateful Filtering (eMail)

- In this section, we develop an example rule set for a stateful firewall
- Later, we will do the same for a stateless packet filter.



- The security policy
  - Incoming and outgoing email should be the only allowed traffic into and out of a protected network
  - Email is SMTP, TCP port 25
  - Anyone in the internal network can send out emails to arbitrary mailservers in the Internet
  - Incoming emails must only arrive at the **Mailserver**

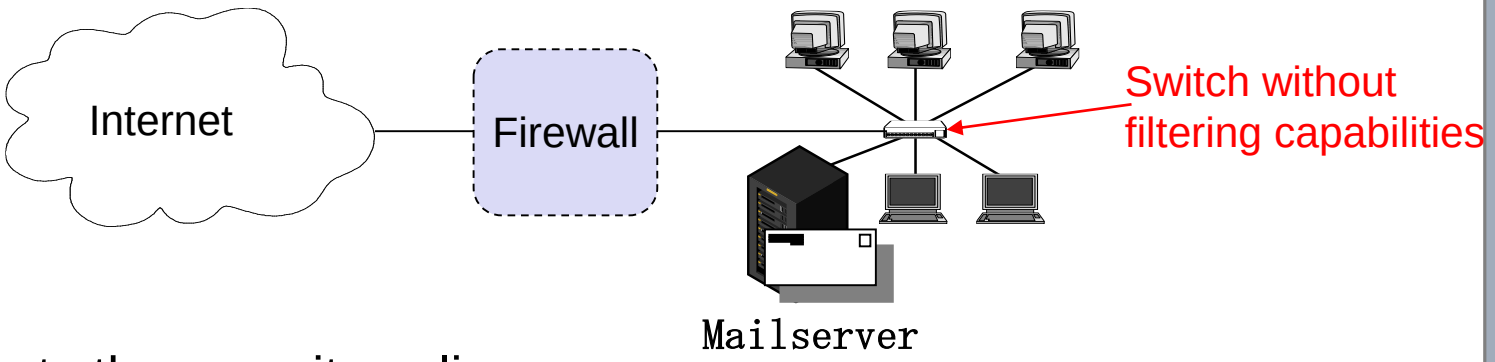
Example: Stateful Filtering (email)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	State	Action
A	Inbound	External	**Mailserver**	TCP	Any	25	New	Permit
B	Outbound	Internal	External	TCP	Any	25	New	Permit
C	Either	Any	Any	TCP	Any	Any	Established	Permit
D	Either	Any	Any	Any	Any	Any	Any	Deny

- Rule A allows new incoming SMTP (TCP port 25) connections to establish a connection with the internal **Mailserver**
- Rule B allows establishing SMTP connection from the internal network to the Internet
- Rule C allows all established connections. Only with rule A and B, a connection can be in the ESTABLISHED state.
- Rule D denies the rest (whitelisting)

Stateful Filtering (email) - Discussion

- Can we do better?
  - Internal hosts can establish connections to the **Mailserver**
- Can we prevent his?
  - No! The firewall cannot intercept these connections, attributable to the network topology.



- This subverts the security policy
- Simple fix 1: Check the security requirements, update the policy
- Simple fix 2: Replace the internal switch by a second firewall

Stateful Filtering (email) - Discussion

Possible weakness

- In the range of the well-known ports, is **Mailserver** on TCP port 25 the only externally reachable entity?
- No!
- Assume an internal host sends out a TCP packet with source and destination port 25 to `shadymail.example`
- Rule B establishes a new state in the firewall.
- Now, for `shadymail.example`, using source port 25, the internal host is reachable on the well-known port 25!
- Fix: make sure that only source ports >1023 are allowed to establish a connection

Example: Stateful Filtering (eMail)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	State	Action
A	Inbound	External	Mailserver	TCP	>1023	25	New	Permit
B	Outbound	Internal	External	TCP	>1023	25	New	Permit
C	Either	Any	Any	TCP	Any	Any	Established	Permit
D	Either	Any	Any	Any	Any	Any	Any	Deny

Stateful Filtering (eMail) - Discussion

Tuning the rule set

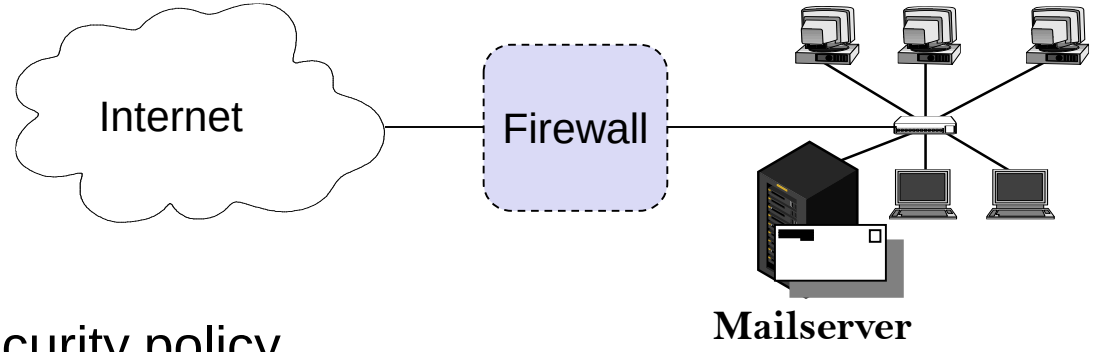
- Firewall rules are matched sequentially
- Few packets will establish a new connection
- Many packets will use an established connection
- Move rule C to the front
- A connection can only be in ESTABLISHED state by rule A and B, the transformation preserves the semantics

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	State	Action
C	Either	Any	Any	TCP	Any	Any	Established	Permit
A	Inbound	External	Mailserver	TCP	>1023	25	New	Permit
B	Outbound	Internal	External	TCP	>1023	25	New	Permit
D	Either	Any	Any	Any	Any	Any	Any	Deny

Stateless Filtering

- The same scenario for a stateless firewall

Recap:



- The security policy
  - Incoming and outgoing email should be the only allowed traffic into and out of a protected network
  - Email is SMTP, TCP port 25
  - Anyone in the internal network can send out emails to arbitrary mailservers in the Internet
  - Incoming emails must only arrive at the **Mailserver**

Example: Stateless Filtering (email)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Mailserver	TCP		25		Permit
B	Outbound	Mailserver	External	TCP		>1023		Permit
C	Outbound	Internal	External	TCP		25		Permit
D	Inbound	External	Internal	TCP		>1023		Permit
E	Either	Any	Any	Any		Any		Deny

- Rule A allows incoming email to enter the network and rule B allows the acknowledgements to exit the network
- Rules C and D are analogous for outgoing email
- Rule E denies all other traffic

Example: Stateless Filtering (email)

Discussion

- Consider, for example, a packet which “wants” to enter the protected subnet and has a *spoofed* IP source address from the internal network:
  - As all allowed inbound packets must have external source and internal destination addresses (A, D) this packet is successfully blocked
  - The same holds for outbound packets with external source addresses (B, C)
- Consider now telnet traffic:
  - As a telnet server resides usually at port 23, and all allowed inbound traffic must be either to port 25 or to a port number > 1023, incoming packets to initiate an incoming telnet connection are successfully blocked
  - The same holds for outgoing telnet connections

Example: Stateless Filtering (eMail)

A possible attack

- The rule set is flawed. For example, it does not block the X11-protocol for remote operation of X-Window applications on the Mailserver
  - An X11-server usually listens at port 6000, clients use port numbers > 1023
  - Thus, an incoming X11-request is not blocked (D),
    neither is any answer (B)
  - This is highly undesirable, as the X11-protocol offers many vulnerabilities to an attacker, like reading and manipulating the display and keystrokes

Example: Stateless Filtering (eMail)

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Mailserver	TCP	>1023	25		Permit
B	Outbound	Mailserver	External	TCP	25	>1023		Permit
C	Outbound	Internal	External	TCP	>1023	25		Permit
D	Inbound	External	Internal	TCP	25	>1023		Permit
E	Either	Any	Any	Any	Any	Any		Deny

- The flaw can be fixed by including the source ports
  - As now outbound traffic to ports >1023 is allowed only if the source port is 25 (B), traffic from internal X-clients or -servers (port >1023) will be blocked
  - The same holds for inbound traffic to ports >1023 (D)
- However, it can not be assumed for sure that an attacker will not use port 25 for his attacking X-client:
  - In this case the above filter will let the traffic pass

Example: Stateless Filtering (eMail)

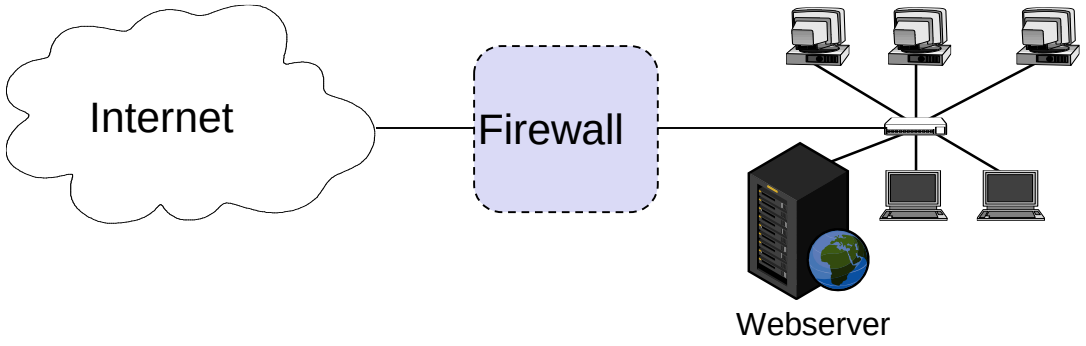
Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Mailserver	TCP	>1023	25	Any	Permit
B	Outbound	Mailserver	External	TCP	25	>1023	Yes	Permit
C	Outbound	Internal	External	TCP	>1023	25	Any	Permit
D	Inbound	External	Internal	TCP	25	>1023	Yes	Permit
E	Either	Any	Any	Any	Any	Any	Any	Deny

- This problem can be addressed by also specifying TCP's ACK-flag in rules B and D:
  - As the ACK-flag is required to be set in rule B, it is not possible to open a new TCP connection in the outbound direction to ports >1023, as TCP's connect-request has the ACK-flag not set
  - The same holds for the inbound direction, as rule D requires the ACK-flag to be set
- As a basic rule, any filtering rule that permits incoming TCP packets for outgoing connections should require the ACK-flag to be set

Stateless Filtering – Limitations and Advantages

- Stateless Filtering assumes that information in the packets can be trusted.
  - An attacker could send SYN/ACK to the network as initial packet.
    - It will pass the firewall.
    - Yet, the host will ignore it.
- Protocols like UDP do not have useful state information
  - Not possible to utilize such information to differentiate between initiator and responder.
- Advantages of stateless operation
  - Simpler to implement, less complexity
  - Highest performance

Firewall Scenario 2



- Allow HTTP traffic initiated by external hosts to webserver (TCP port 80)
- Allow internal hosts to initiate
  - HTTP traffic to Internet (TCP, port 80)
  - DNS traffic to Internet (UDP, port 53)
- Do not allow other communication, in particular no communication initiated by external hosts to the local hosts other than the webserver.

Firewall Scenario 2 – Example *Stateful* Filtering

Rule Set for Stateful Filtering

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	State	Action
A	Inbound	External	Webserver	TCP	>1023	80	New	Permit
B	Outbound	Internal	External	TCP	>1023	80	New	Permit
C	Outbound	Internal	External	UDP	>1023	53	New	Permit
D	Either	Any	Any	Any	Any	Any	Established	Permit
E	Either	Any	Any	Any	Any	Any	Any	Deny

Firewall Scenario 2 – Example *Stateless* Filtering

Rule Set for Stateless Filtering

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest.Port	ACK	Action
A1	Inbound	External	Webserver	TCP	>1023	80	Any	Permit
A2	Outbound	Websrv.	External	TCP	80	>1023	Yes	Permit
B1	Outbound	Internal	External	TCP	>1023	80	Any	Permit
B2	Inbound	External	Internal	TCP	80	> 1023	Yes	Permit
C1	Outbound	Internal	External	UDP	>1023	53	-	Permit
C2	Inbound	External	Internal	UDP	53	>1023	-	Permit
D	Either	Any	Any	Any	Any	Any	Any	Deny

- New connections are identified using the TCP SYN message (only message with ACK = false).
- In the direction of the accepted initiator both states are accepted, in the other direction only messages with ACK = true.
- This cannot be used for UDP, so rule C2 messages are not necessarily replies.

Common Firewall Policy Errors

- How is your firewall management interface reachable?
  - From the Internet? From the complete internal network?
  - Via telenet? Via UPnP
- What is allowed over the Internet?
  - NetBIOS? NFS? RPC? Telnet?
  - Other ICMP than Unreachable, Fragmentation Needed, TTL Exceeded, Ping
  - IP header options?
- IPv4 and IPv6?
  - Are the rule sets compliant?
- Outbound rule *ANY*?
  - Even private IP ranges?
  - Even from IP ranges that don't belong to you?
- Does your understanding of Inbound and Outbound match the Firewall's understanding?
  - If eth0 is your internal interface and the firewall says inbound on eth0, our understand says outbound.

Common Firewall Policy Errors

- Shadowing
  "*refers to the case where all the packets one rule intends to deny (accept) have been accepted (denied) by preceding rules"* [fireman06]

Rule	Direction	Src. Addr.	Dest. Addr.	Action
A		Any	192.168.0.0/16	Permit
B		Any	192.168.42.0/24	Deny

- Rule B will never match!

Firewall Terminology & Building Blocks for Firewall Architectures

- *Firewall:*
  - A component or a set of components that restricts access between a protected network and the Internet or between other sets of networks
- *Packet Filtering:*
  - The action a device takes to selectively control the flow of data to and from a network
  - Packet filtering is an important technique to implement **access control** on the subnetwork-level for packet oriented networks, e.g. the Internet
- *De-militarized zone (DMZ) :*
  - A subnetwork added between an external and an internal network, in order to provide an additional layer of security; also called **perimeter network**
- *Bastion Host:*
  - A computer that must be highly secured because it is more vulnerable to attacks than other hosts on a subnetwork
  - A bastion host in a firewall is usually the main point of contact for user processes of hosts of internal networks with processes of external hosts

Bastion Hosts (1)

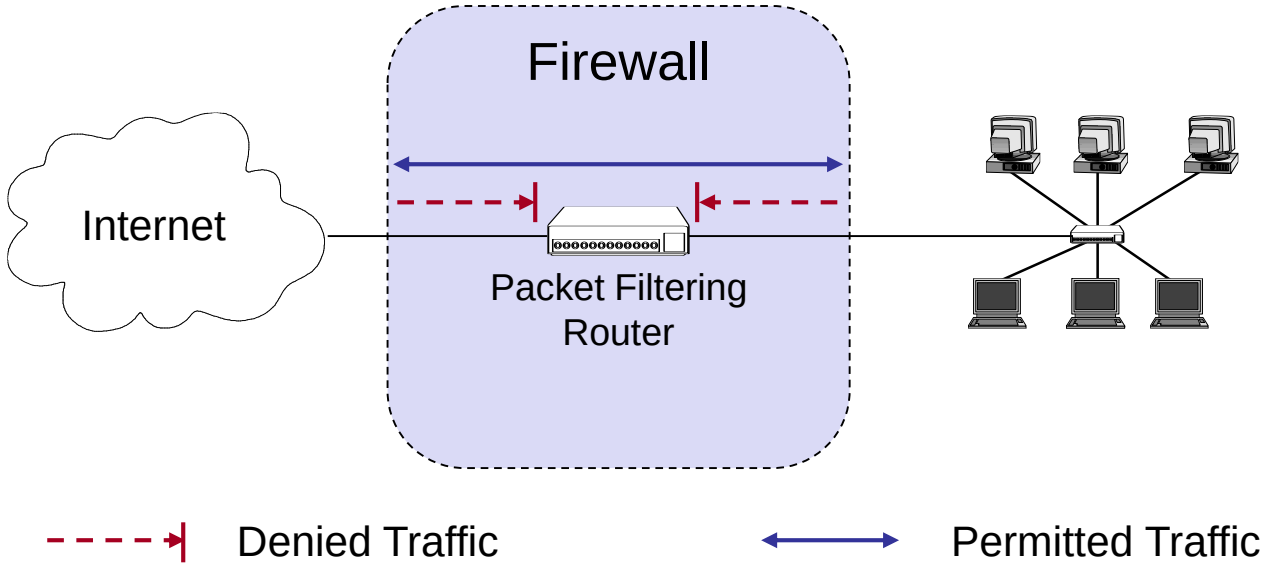
- A bastion host is defined as a host that is more exposed to the hosts of an external network than the other hosts of the network it protects
- A bastion host may serve for different purposes:
  - Packet filtering
  - Providing proxy services
  - A combination of both
- The principles for building a bastion hosts are extensions of those for securing any mission critical host:
  - Keep it simple
  - Prepare for the bastion host to be compromised:
    - Internal hosts should not trust it more than necessary
    - If possible, it should be connected in such a way to the network that it cannot sniff internal traffic
    - Provide extensive logging for incident detection / analysis, if possible such that it can not be easily tampered with even when the host is compromised

Bastion Hosts (2)

- Further guidelines:
  - Make the bastion host unattractive:
    - Slower machines are less appealing targets and are less useful if compromised
    - However, if the bastion host offers some resource consuming service, e.g. WWW-service, it may be wiser not to make it too slow
    - The fewer tools are available on the bastion host, the less useful the machine is to an attacker
  - Get a reliable hardware configuration
  - The bastion host should be placed at a physically secure location
  - Disable any user accounts on the bastion host (e.g. only administrators can login to the Bastion host)
  - Secure the system logs (e.g. by writing them directly to a printer, or using the printer port to a dedicated PC which is not networked)
  - Do regular backups of the system logs and the configuration (using a dedicated backup device)
  - Monitor the machine closely (reboots, usage / load patterns, etc.)
  - If possible, restore the machine regularly from a prepared installation

Firewall Architectures (1)

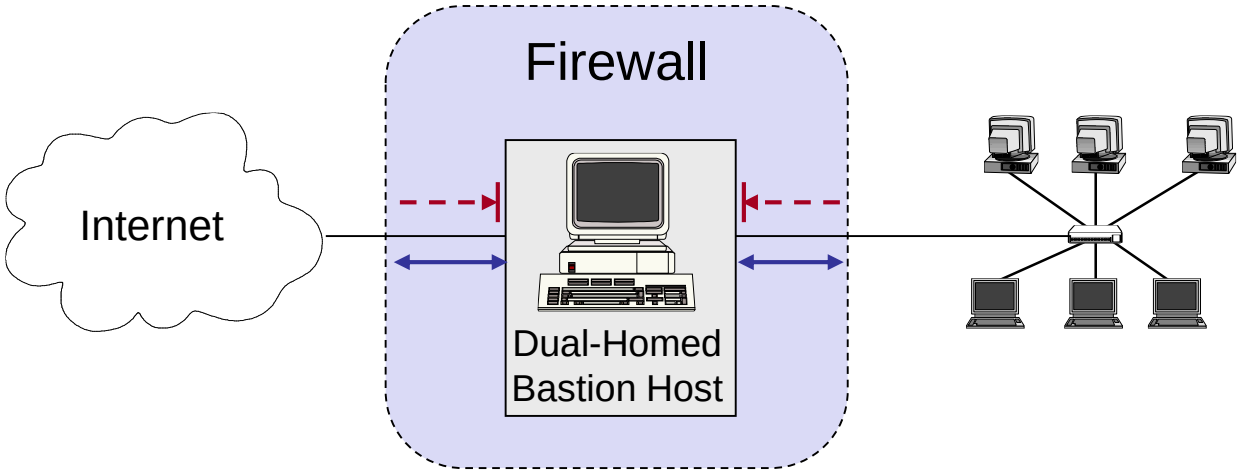
The Simple Packet Filter Architecture



- The most simple architecture just consists of a packet filtering router
- It can be either realized with:
  - A standard workstation (e.g. Linux PC) with at least two network interfaces plus routing and filtering software
  - A dedicated router device, which usually also offers filtering capabilities

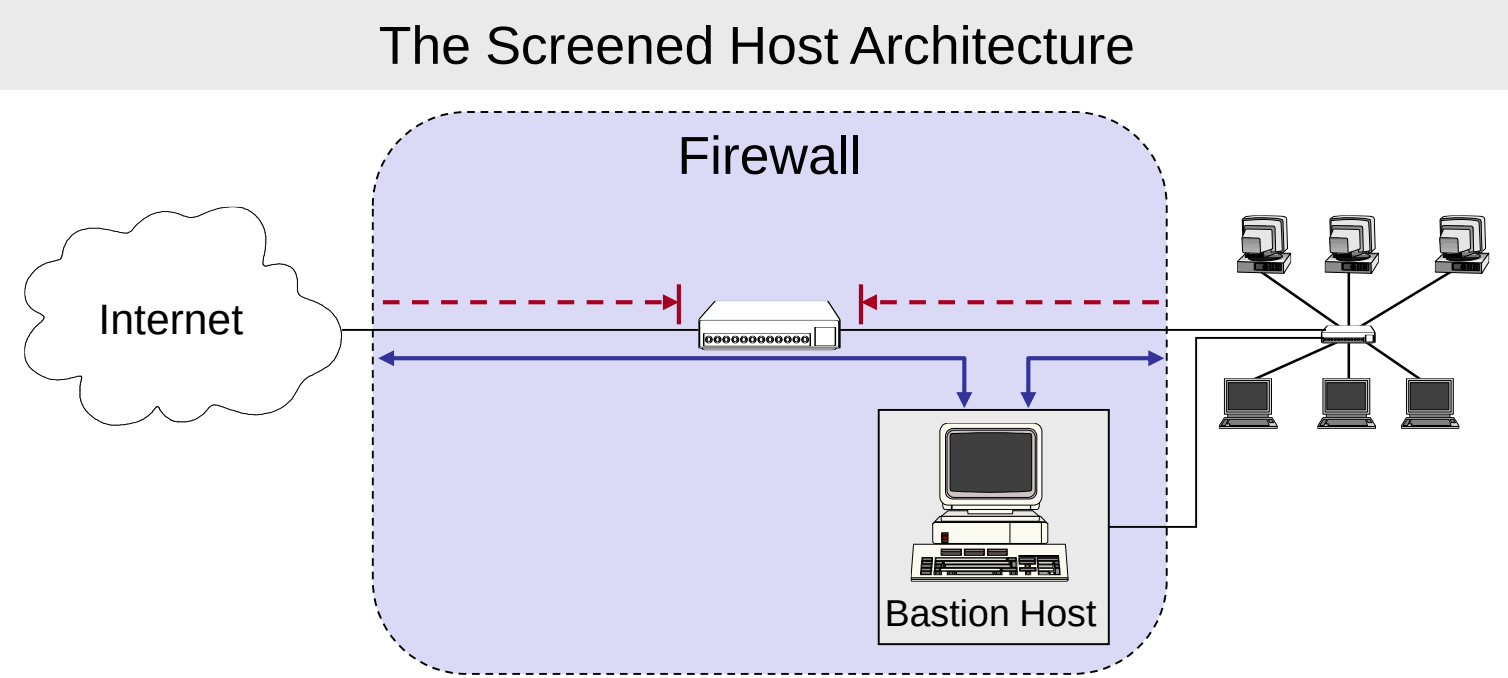
Firewall Architectures (2)

The Dual-Homed Host Architecture



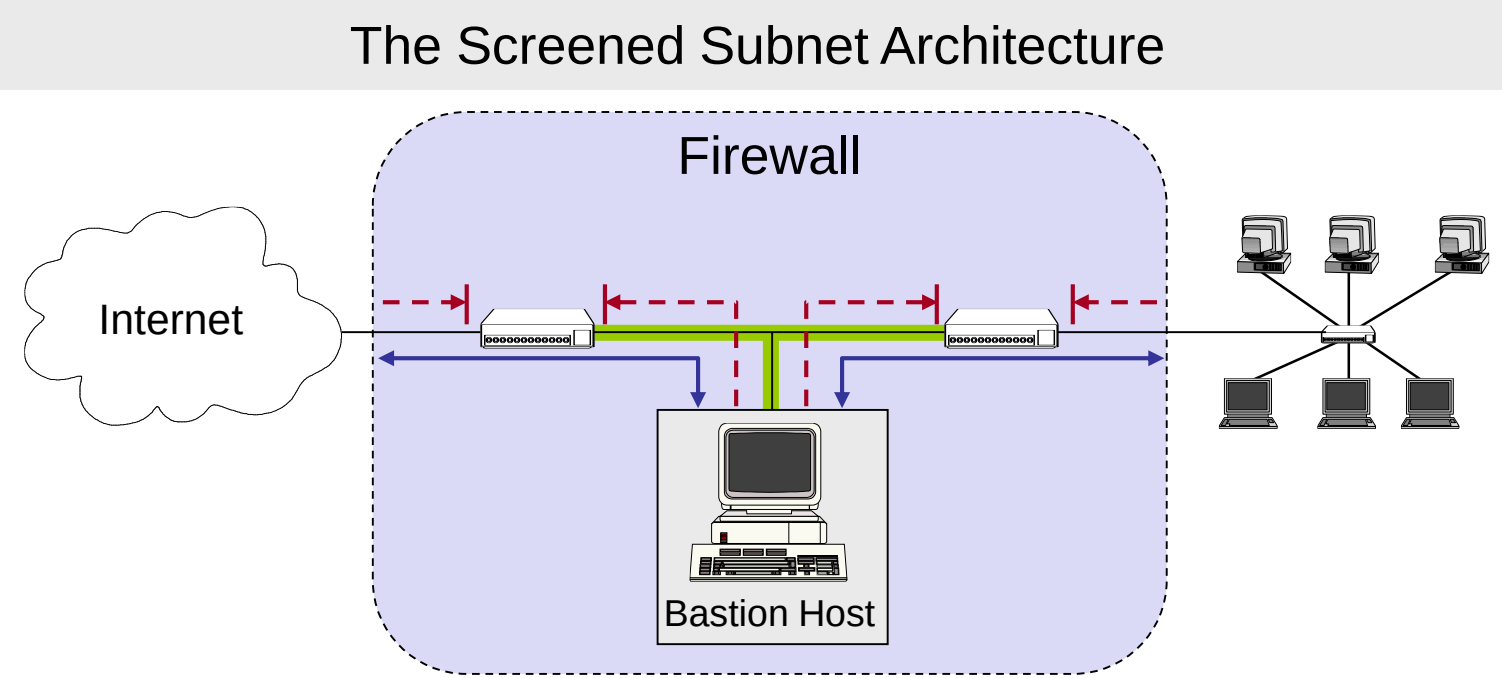
- The dual-homed host provides:
  - Proxy services to internal and / or external clients
  - Potentially additional packet filtering capabilities
- Properties of the dual-homed host:
  - It has at least two network interfaces
- Drawback: As all permitted traffic passes through the bastion host, this may introduce a performance bottleneck

Firewall Architectures (3)



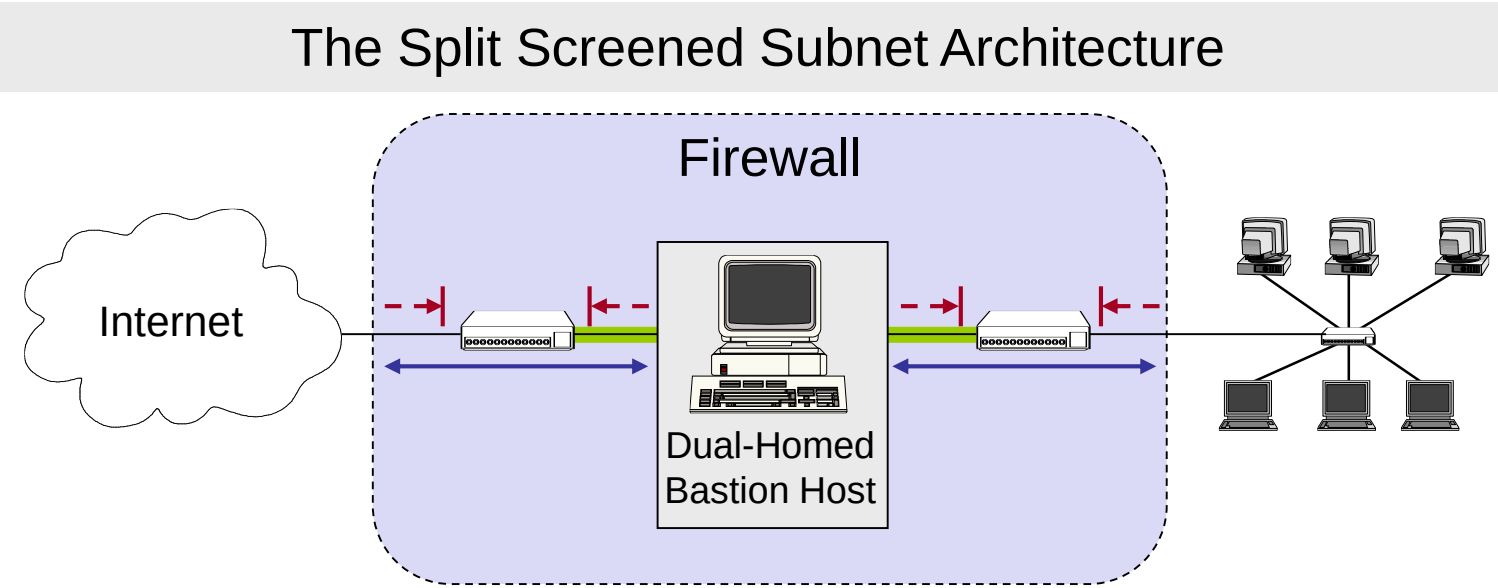
- The packet filter:
  - Allows permitted IP traffic between screened host and the Internet
  - Blocks direct traffic between other internal hosts and the Internet
- The screened host provides proxy services:
  - The screened host acts as a bastion host, being partially protected by the packet filter

Firewall Architectures (4)



- A DMZ is created between two packet filters
- The inner packet filter serves for additional protection in case the bastion host is compromised:
  - This avoids a compromised bastion host to sniff internal traffic
- The DMZ (i.e., perimeter network) is also a good place to host a publicly accessible servers, e.g. webservers

Firewall Architectures (5)



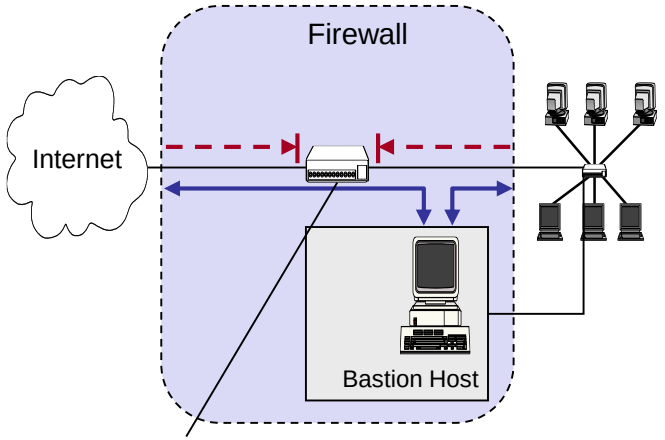
- A dual-homed bastion host splits the perimeter network in two distinct networks
- This provides defense in depth, as:
  - The dual-homed bastion host provides finer control on the connections as his proxy services are able to interpret application protocols
  - The bastion host is protected from external hosts by an outer packet filter
  - The internal hosts are protected from the bastion host by an inner packet filter

Bastian Host and (Stateless) Firewall – Example Ruleset from the Stateless Filtering Section

Rule	Direction	Src. Addr.	Dest. Addr.	Protocol	Src. Port	Dest. Port	ACK	Action
A	Inbound	External	Bastion	TCP	>1023	25	Any	Permit
B	Outbound	Bastion	External	TCP	25	>1023	Yes	Permit
C	Outbound	Bastion	External	TCP	>1023	25	Any	Permit
D	Inbound	External	Bastion	TCP	25	>1023	Yes	Permit
E	Either	Any	Any	Any	Any	Any	Any	Deny

- If the firewall comprises a bastion host, the packet filtering rules should further restrict traffic flow (→ screened host architecture):
  - As in the modified rules above only traffic between the Internet and the bastion host is allowed, external attackers can not attack SMTP on arbitrary internal hosts any longer

The Screened Host Architecture



Only traffic of the Bastion Host has to traverse the firewall.

Conclusion to Network Firewalls (1)

- What firewalls can do:
  - A firewall is a focus for security decisions
  - A firewall can enforce a security policy, i.e. concerning access control
  - A firewall can log Internet activity efficiently
  - A firewall can block unwanted traffic if the traffic can be characterized,
    - e.g. with an IP 5-tuple: IP source address, IP destination address, source port number, destination port number, transport protocol
  - A firewall can limit exposure to security problems in one part of a network

Conclusion to Network Firewalls (2)

- What firewalls can not do:
  - A firewall can't protect against malicious insiders
  - A firewall can't protect against connections that don't go through it
  - A firewall can't protect against completely new threats
  - A firewall can't fully protect against viruses,
    - e.g. if viruses are spread through emails, and the email service is allowed through the firewall, which is typically the case
  - A firewall does not perform cryptographic operations, e.g. message authentication

    (however, firewalls are often co-located with VPN end-points)
  - A firewall can't set itself up correctly ($\Rightarrow$ cost of operation)

Overview

- Virtual Private Networks

Virtual Private Networks - Defintions

- Various definitions of the term *virtual private network (VPN):*
  - A virtual private network (VPN) is a network that **uses a public telecommunication infrastructure**, such as the Internet, to provide remote offices or individual users with secure access to their organization's network. An alternative to such a virtual private network is an expensive system of owned or leased lines that can be used only by a single organization. The goal of a VPN is to provide the organization with the same capabilities, but at a much lower cost. [SeSec08]
  - A communications environment in which **access is controlled** to permit peer connections only within a defined community of interest, and is constructed through **some form of partitioning** of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis
  - A restricted-use, logical computer network that is constructed from the system resources of a relatively public, physical network (such as the Internet), often by using encryption, and often by tunneling links of the virtual network across the real network [RFC2828]

Techniques for building Virtual Private Networks

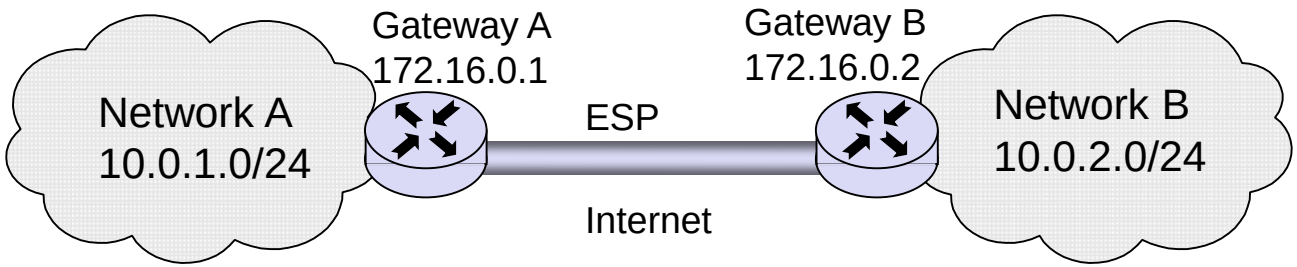
- Make use of dedicated links
  - ATM or Frame Relay virtual connections
  - Multi-Protocol Over ATM (MPOA)
  - Multi-Protocol Label Switching (MPLS)

- *Controlled route leaking / route filtering:*
  - Basic idea: control route propagation to the point that only certain networks receive routes for other networks
  - This intends to realize *"security by obscurity"*

- *Tunneling:*
  - Generic routing encapsulation (GRE)
  - PPP / PPTP / L2TP
    - Note: PPTP was developed by Microsoft and used MS-CHAP(Microsoft Challenge-handshake Authentication Protocol) and MPPE (Microsoft Point-to-Point Encryption Protocol)
    - It is currently considered as inherently insecure, since messages can be easily spoofed
  - IPSec (See examples provided in Chapter 8)
  - SSL (e.g. OpenVPN is based on SSL/TLS)
  - SSH: OpenSSH offers also (since Version 4.3) a possibility for building VPNs

Example: VPN with IPSec

- ***ESP Tunnel*** for VPN



- Configuration at Gateway A:
  - spdadd 10.0.1.0/24 10.0.2.0/24 any -P out ipsec
    esp/tunnel/**172.16.0.1-172.16.0.2**/require ;
  - spdadd 10.0.2.0/24 10.0.1.0/24 any -P in ipsec
    esp/tunnel/**172.16.0.2-172.16.0.1**/require ;

  Note
  - protocol and port (any)
  - the policy to use (–P) specifying direction (in/out), action (ipsec/discard/none), the protocol (ah/esp), the mode (tunnel/transport), and the level (uses/require)
- Configuration at Gateway B:
  - spdadd 10.0.2.0/24 10.0.1.0/24 any -P out ipsec
    esp/tunnel/**172.16.0.2-172.16.0.1**/require ;
  - spdadd 10.0.1.0/24 10.0.2.0/24 any -P in ipsec
    esp/tunnel/**172.16.0.1-172.16.0.2**/require ;

Overview

- Case study: Linux Netfilter

Linux Firewalls

- Packet management in Linux is implemented using the *Netfilter* architecture (www.netfilter.org)
  - Allows to define firewalls, NAT, etc.

- The Linux command „iptables“ is used to configure netfilter rules, e.g.

```
    # Allow port 22 (ssh) new TCP connection from
    # source IP address range 192.168.1.100/32
iptables -A INPUT -p tcp -s 192.168.1.100/32 --dport 22 -m state --state ESTABLISHED
    -j ACCEPT
iptables -A INPUT -p tcp -s 192.168.1.100/32 --dport 22 -m state --state NEW -j ACCEPT
```

-A INPUT: append new rule to rule chain INPUT

-m (match)
-m state --state: NEW/ESTABLISHED/RELATED

-j (jump): ACCEPT/DROP

(Note: First iptables statement above increases performance.)

Netfilter Concept 1: Chains (1)

- The packets are processed in so-called „***chains***“
- A “chain” is a collection of rules that *may* be applied – i.e. chains give a “treatment” a name and a value
- Each incoming or outgoing packet is checked against these rules
- Each rule says “if the packet header looks like this, then here's what to do with the packet”, e.g. drop or accept
- Firewall functionality is implemented using 3 chains
  - The „***input***“ chain: for processing packets addressed to this host
  - The „***output***“ chain: for processing packets coming from local processes and leaving this host
  - The „***forward***“ chain: for processing packets that are traversing this host.

    If a Linux machine is used as a router, this chain will be frequently used.

    If a Linux machine is used as an end-host and not as a Linux router, this chain will not be used

Netfilter Concept 1: Chains (2)

- Chains can also be used for, e.g. NAT functionality
  - „***Pre-Routing***“: here packets are processed before a routing decision is taken

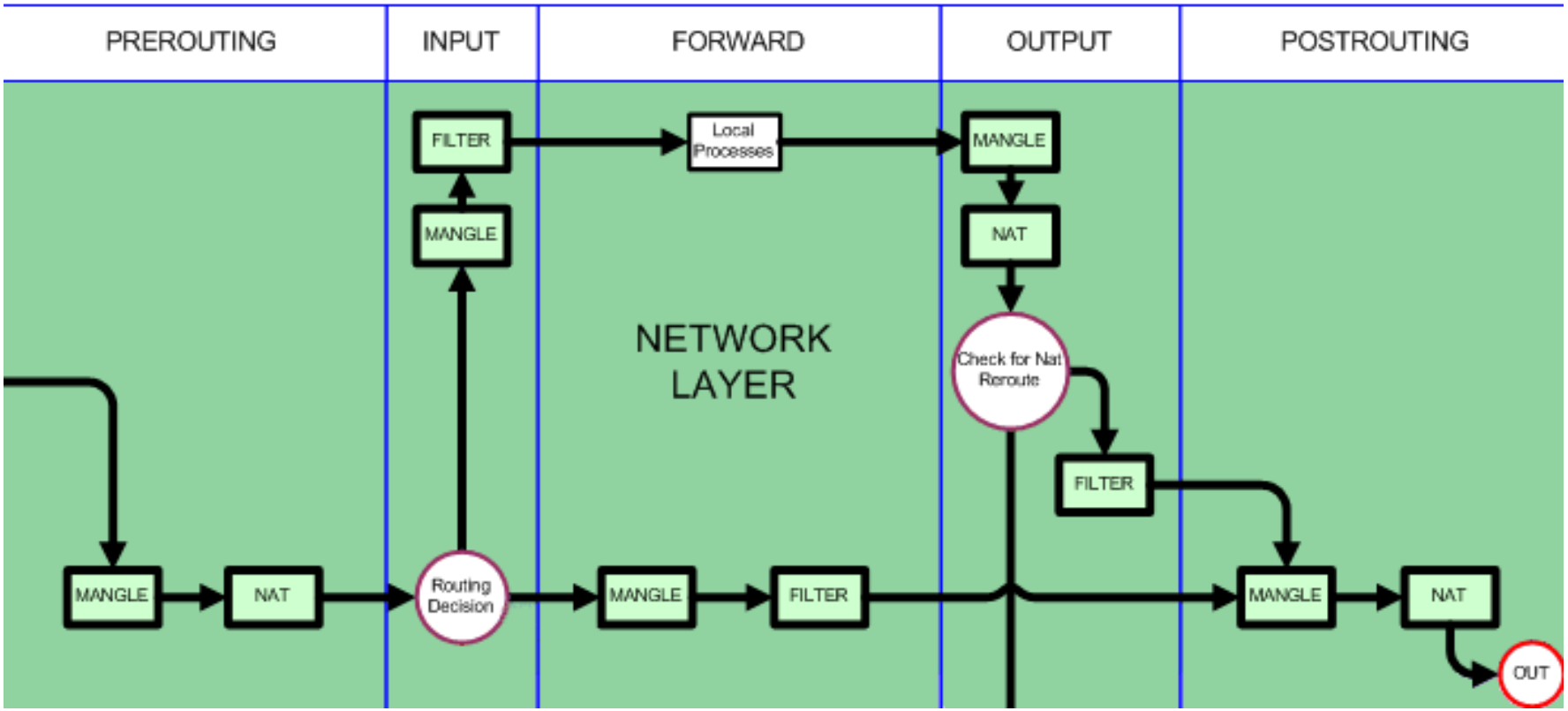
    ➔ If the destination IP address needs to be modified, an appropriate rule is required here
  - “***Post-routing***”: here packets are processed after a routing decision has been taken

    ➔ The destination IP address cannot be modified here.

Netfilter Concept 2: Tables

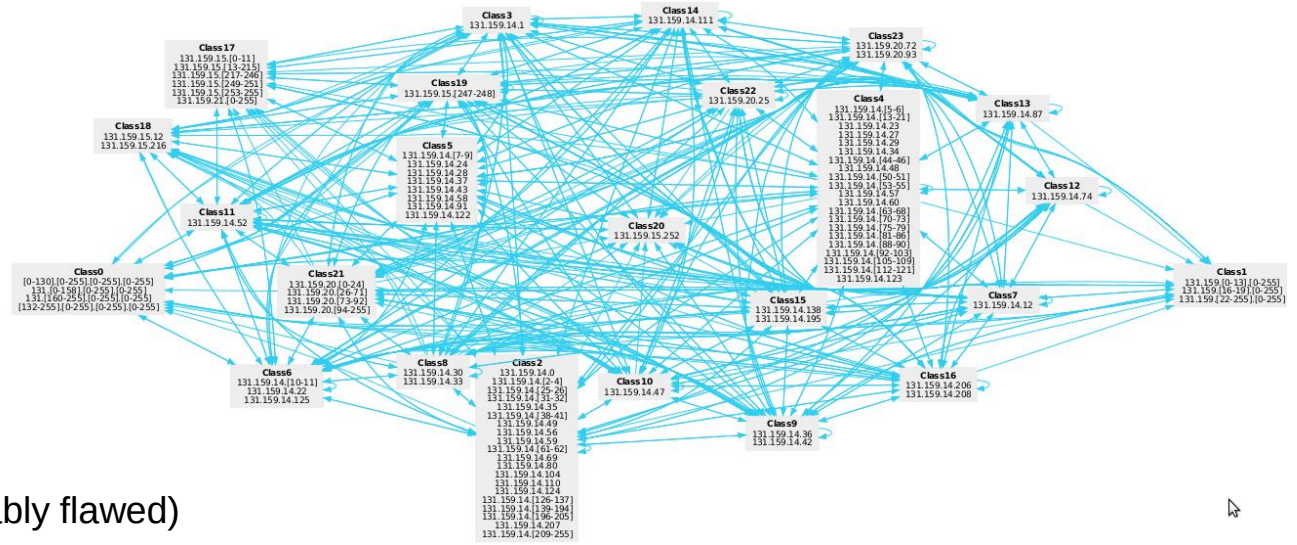
- „Tables“ are used to group chains (and the rules therein)
- Tables indicate the „purpose“ of processing:
  - FILTER = default table
  - NAT = Network Address Translation
  - MANGLE = Change packet according to some rules
  - Etc.
- Tables and chains are evaluated in a fixed order
  - The order of chains is fixed and always adhered to
  - For each chain type, a table may then contain specific entries

Netfilter: Chains and Tables



Case Study: Our Firewall

- Netfilter / iptables
- 2884 lines
  - A 2000-2001 found the typical average rule set size is 144 [Wool04]
- Prevent MAC spoofing if IPv4 address and MAC address are known
  - Iptables -A custom_chain -s my_ipv4_addr -m mac --mac-source my_mac_address -j RETURN
  - Iptables -A custom_chain -s my_ipv4_addr -j DROP
- Our SSH landscape



Calculated by ITVal (probably flawed)

Additional References

[netfi08] "Netfilter Online Documentation",
http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.html

[Sem96a] C. Semeria. *Internet Firewalls and Security.* 3Com Technical Paper, 1996.

[Wack95a] J. P. Wack, L.J. Carnahan. *Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls.* NIST Special Publication 800-10, 1995.

[Zwi00a] E. Zwicky, S. Cooper, B. Chapman. *Building Internet Firewalls.* Second Edition, O'Reilly, 2000.

[SeSec08] "SearchSecurity.com Definitions; virtual private network" http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213324,00.html

[Bishop03] M. Bishop. "What is computer security?" Security and Privacy, 2003

[fireman06] L. Yuan, H. Chen, J. Mai, C. Chuah, Z. Su, P. Mohapatra. *FIREMAN: a toolkit for firewall modeling and analysis.* Security and Privacy, 2006

[Wool04] A. Wool (2004): *A quantitative study of firewall configuration errors*. Computer, IEEE 37(6), pp. 62 – 67