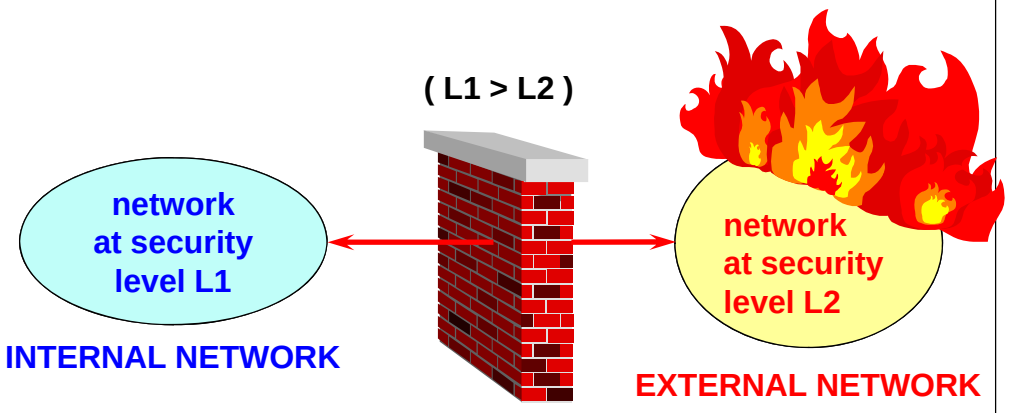
Firewall and IDS/IPS

Antonio Lioy
< lioy @ polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

What is a firewall?

- firewall = wall to protect against fire propagation
- controlled connection between networks at different security levels = boundary protection



Ingress vs. Egress firewall

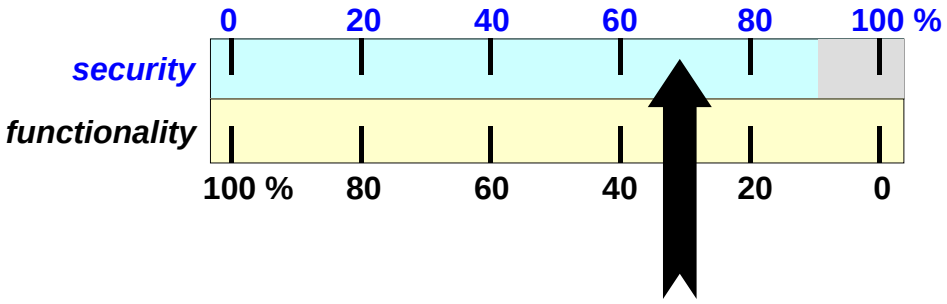
- **ingress firewall**
  - incoming connections
  - typically to protect my public services
  - sometimes as part of an application exchange initiated by my users
- **egress firewall**
  - outgoing connections
  - typically to check the activity of my personnel (!)
- **easy classification for channel-based services (e.g. TCP applications), but difficult for message-based services (e.g. ICMP, UDP applications)**

Firewall design

You don't "buy" a firewall, you design it!
(you can buy its components)

- **we need to achieve an optimal trade-off ...**
- **... between security and functionality**
- **... with minimum cost**

The security index



THE THREE COMMANDMENTS OF FIREWALL

I. the FW must be the only contact point of the internal network with the external one

II. only the “authorized” traffic can traverse the FW

III. the FW must be a highly secure system itself

D.Cheswick
S.Bellovin

Authorization policies

"All that is not explictly permitted, is forbidden"

- **higher security**
- **more difficult to manage**

"All that is not explictly forbidden, is permitted"

- **lower security (open gates)**
- **more easy to manage**

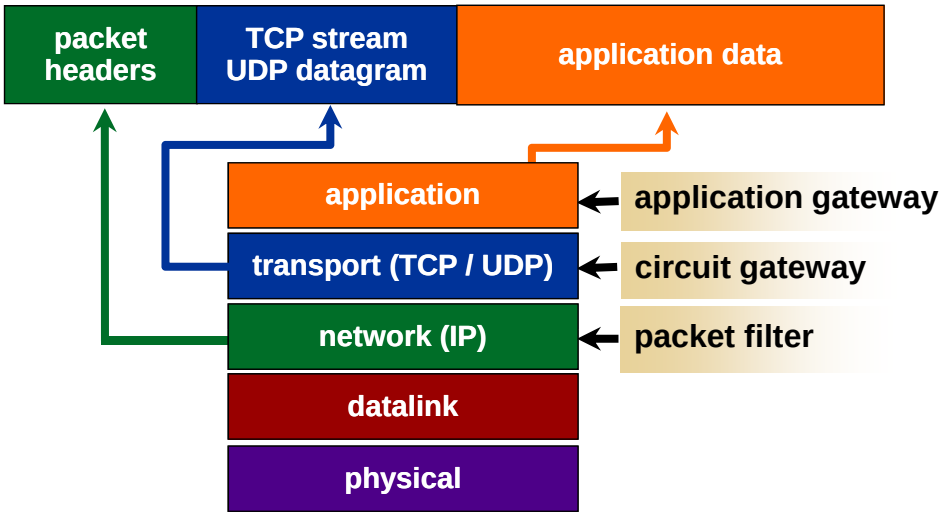
General concepts

- **the bigger an object, the more difficult to verify it**
- **if a process has not been activated, its bugs are not relevant**
- **"big is NOT beatiful" = minimal configuration**
- **a FW is not a general-purpose machine:**
  - minimal sw
  - no users
  - …
- **everybody is guilty unless he proves his innocence**

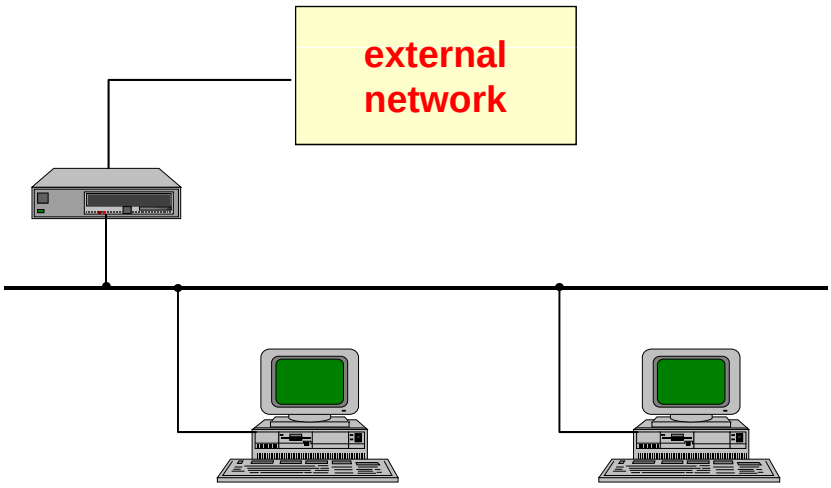
FW: basic components

- **screening router (choke)**
  router that filters traffic at IP level
- **bastion host**
  secure system, with auditing
- **application gateway (proxy)**
  service that works on behalf of an application, with access control
- **dual-homed gateway**
  system with two network cards and routing disabled

A which level the controls are made?



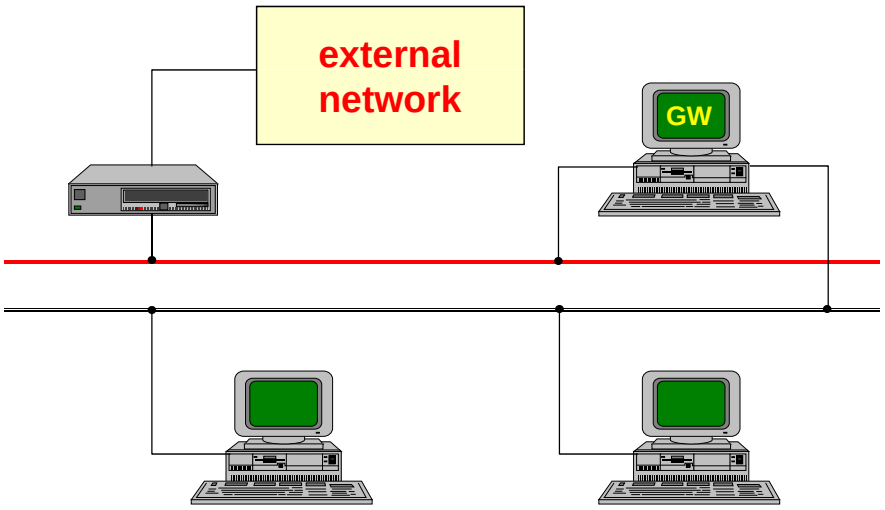
"Screening router" architecture



"Screening router" architecture

- exploits the router to filter the traffic both at IP and upper levels
- no need for dedicated hardware
- no need for a proxy and hence no need to modify the applications
- simple, easy, cheap and ... insecure!

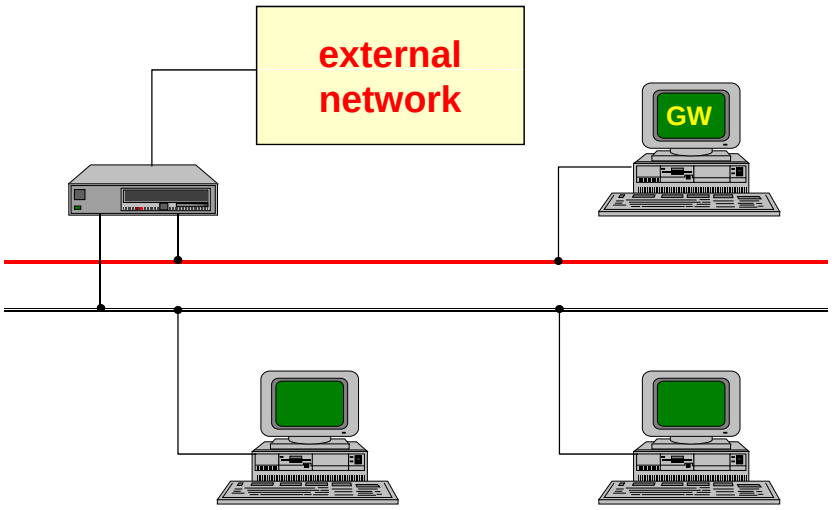
"Dual-homed gateway" architecture



"Dual-homed gateway" architecture

- easy to implement
- small additional hardware requirements
- the internal network can be masqueraded
- unflexible
- large work overhead

"Screened host" architecture



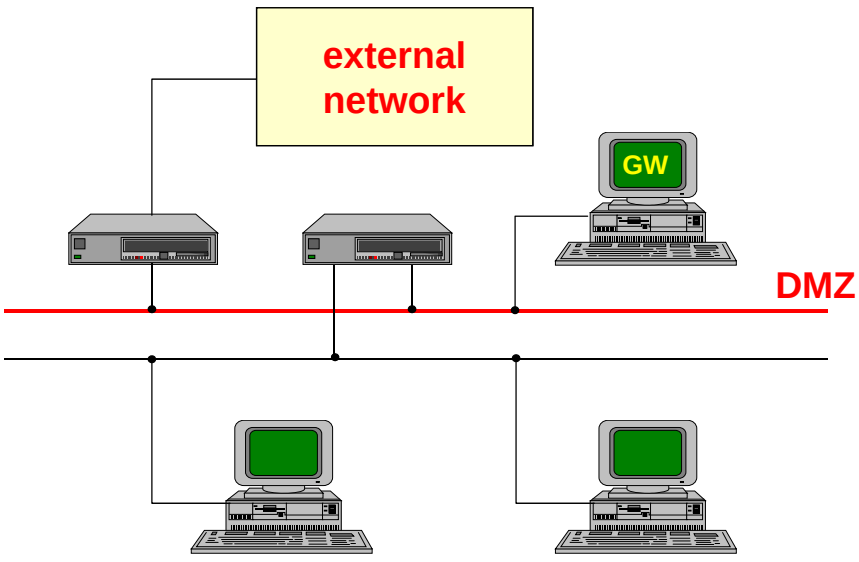
"Screened-host" architecture

- **router:**
  - blocks the packets from INT to EXT unless they come from the bastion host
  - blocks the packets from EXT to INT unless they go to the bastion host
  - exception: directly enabled protocols
- **bastion host:**
  - circuit/application level gateway to selectively enable some services servizi

"Screened-host" architecture

- more expensive
- more flexible
- complex to manage: two systems rather one
- possible selectively relax the controls over some services / hosts
- only the hosts/protocols passing through the bastion can be masqueraded (unless the router offers the NAT functionality)

"Screened subnet" architecture

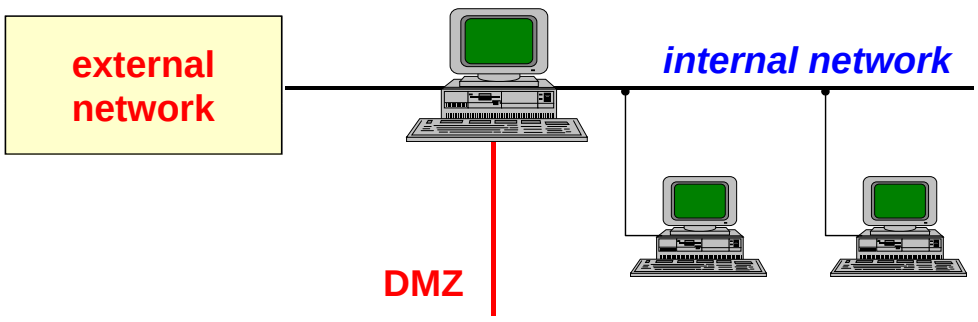


"Screened subnet" architecture

- **DMZ (De-Militarized Zone)**
- **the DMZ is home not only to the gateway but also to other hosts (tipically the public servers):**
  - Web
  - remote access
  - . . .
- **the routing may be configured so that the internal network is unknown**
- **expensive**

"Screened subnet" architecture (version 2)

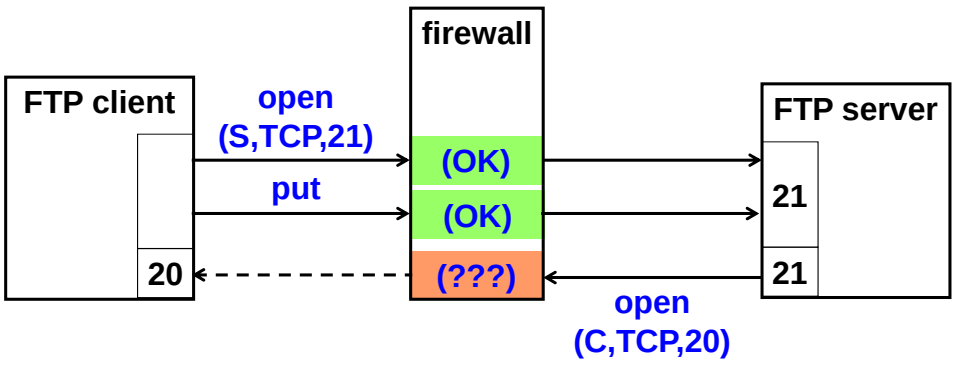
- **to reduce costs and simplify management often the routers are omitted (and their function incorporated into the gateway)**
- **AKA “three-legged firewall”**



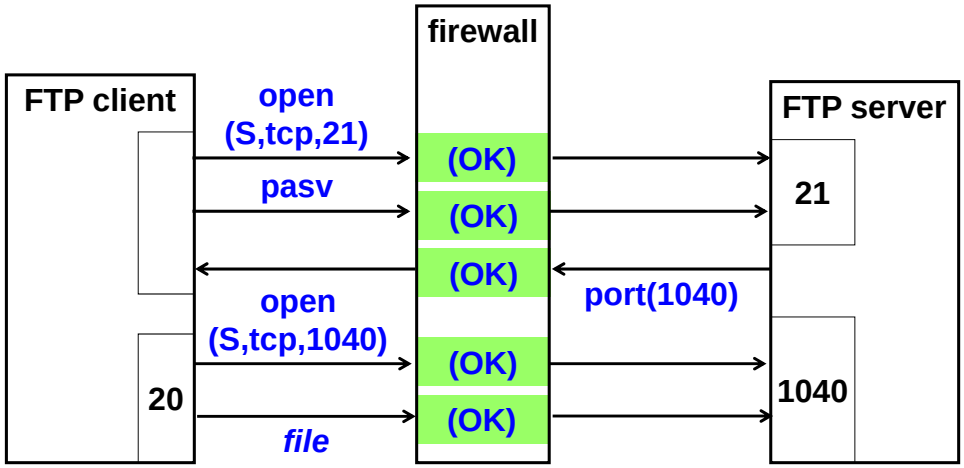
Filters at network level (I)

- **address checking (ingress / egress filtering)**
- **disable incoming services (with exceptions):**
  - e.g. TELNET only towards INET
  - e.g. incoming HTTP only to the DMZ web server
  - problem with FTP (the data transfer channel is always created by the server)
- **ICMP**
  - is dangerous (used for denial-of-service) but useful (ping, traceroute) so don't disable it, just rate-limit it
  - closely monitor REDIRECT packets

Problem with FTP across a firewall



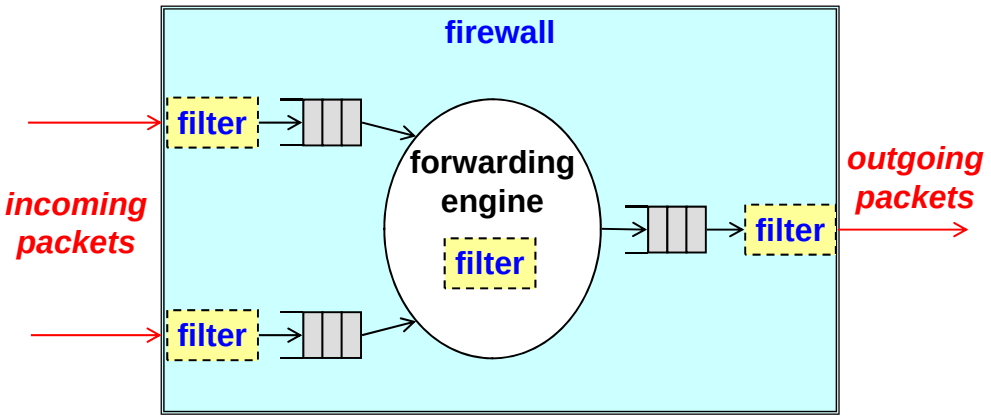
Passive FTP



Filters at network level (II)

- **UDP**
  - is a datagram service, not a virtual circuit (so its checking poses a higher load)
  - RPC uses random ports
  - it is suggested to completely disable it (but DNS)
- **distinguish internal and external interfaces**
- **pay attention to the number of filtering rules and their order: this may drastically change performance**

Filtering points



Filters on a router: an example

- **policy: mail for network 130.193 managed only by 130.193.2.1**
- **sintax of CISCO router:**
  - access-list 100 permit tcp
    0.0.0.0 255.255.255.255
    130.193.2.1 0.0.0.0
    eq 25
  - access-list 101 deny tcp
    0.0.0.0 255.255.255.255
    130.193.0.0 0.0.255.255
    eq 25

Bastion host - configuration

- **should run only the relevant processes**
- **must log (securely!) all its activities**
- **network log onto an internal secure system**
- **must have *source routing* disabled**
- **must have *IP forwarding* disabled**
- **should have "mouse traps" (e.g. fake `ls`)**

Firewall technologies

- **different controls at various network levels:**
  - (static) packet filter
  - stateful (dynamic) packet filter
  - cutoff proxy
  - circuit-level gateway / proxy
  - application-level gateway / proxy
  - stateful inspection
- **differences in terms of:**
  - performance
  - protection of the firewall O.S.
  - keeping or breaking the client-server model

Packet filter

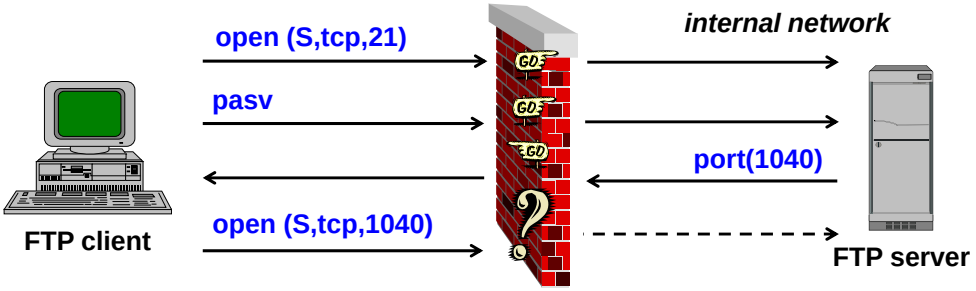
- **historically available on routers**
- **packet inspection at network level**
  - IP header
  - transport header

Packet filter: pros and cons

- **independent of applications**
  - good scalability
  - approximate controls: easy to “fool”
    (e.g. IP spoofing, fragmented packets)
- **good performance**
- **low cost (available on routers and in many OS)**
- **difficult to support services with dynamically allocated ports (e.g. FTP)**
- **complex to configure**

Packet filter & FTP

- **two choices available:**
  - leave open all dynamic ports (>1024)
  - close all dynamic ports
- **difficult trade-off between security and support to FTP!!**



Stateful (dynamic) packet filter

- **similar to packet filter but "state-aware"**
  - state informations from the transport or application level (e.g. FTP PORT command)
  - can distinguish new connections from those already open
    - state tables for open connections
    - packets matching one row in the table are passed without any further control
- **better performance than packet filter**
  - SMP support
- **still has many of the static packet filter limitations**

Application-level gateway

- composed by a set of proxies inspecting the packet payload at application level
- often requires modifications to the client application
- may optionally mask / renumber the internal IP addresses
- when used as part of a firewal, usually performs also peer authentication
- top security!! (e.g. against buffer overflow of the target application)

Application-level gateway (1)

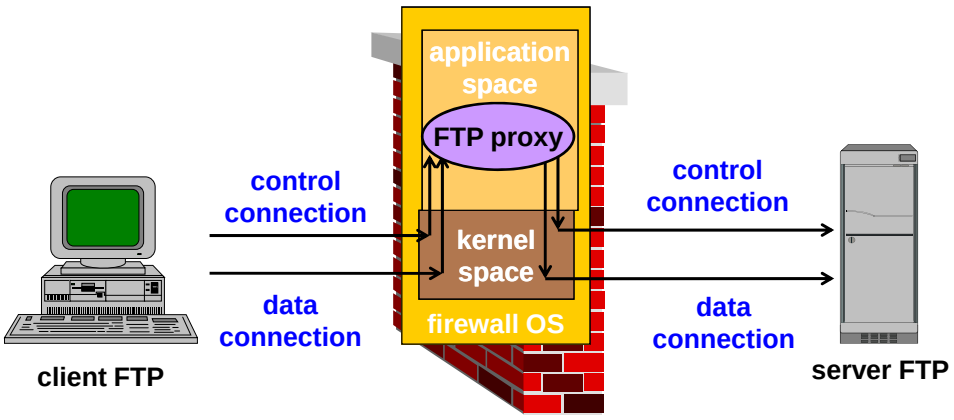
- rules are more fine-grained and simple than those of a packet filter
- every application needs a specific proxy
  - delay in supporting new applications
  - heavy on resources (many processes)
  - low performance (user-mode processes)
- SMP may improve performance
- completely breaks the client/server model
  - more protection for the server
  - may authenticate the client
  - not transparent to the client

Application-level gateway (2)

- **the fw OS may be esposed to attacks**
- **problems with application-level security techniques (e.g. SSL)**
- **variants:**
  - transparent proxy
    - less intrusive for the client
  - strong application proxy
    - only some commands/data are forwarded
    - this is the only correct configuration for an application proxy

Application-level gateway & FTP

- **total control of the application session**



Circuit-level gateway (1)

- **a generic proxy (i.e. not "application-aware")**
  - creates a transport-level circuit between client and server …
  - … but it doesn't understand or manipulate in any way the payload data

Circuit-level gateway (2)

- **breaks the TCP/UDP-level client/server model during the connection**
  - more protection for the server
    - isolated from all attacks related to the TCP handshake
    - isolated from all attacks related to the IP fragmentation
  - may authenticate the client
    - but this requires modification to the application
- **still exhibits many limitations of the packet filter**

SOCKS

- **a transport-level proxy (L4) that implements a circuit-level gateway**
- **invented at MIPS, v4 by NEC, v5 by IETF**
- **aka AFT (Authenticated Firewall Traversal)**
- **requires modified clients:**
  - standard: telnet, ftp, finger, whois
  - library to develop own clients
- **commercial support:**
  - all the major browsers (e.g. FX and IE)
  - some firewalls (e.g. IBM)

SOCKS RFCs

- **RFC-1928 “SOCKS protocol V5”**
- **RFC-1929 “Username/password authentication for SOCKS V5”**
- **RFC-1961 “GSS-API authentication method for SOCKS V5”**
- **RFC-3089 “A SOCKS-based IPv6/IPv4 gateway mechanism”**

SOCKS: how it works

- **the library replaces the standard socket system calls `connect()`, `bind()`, `accept()`, ...**
- **... with calls that:**
  - open a channel to the SOCKS server
  - send version, IP:port, user
- **the server:**
  - checks its ACL
  - opens the required channel (with its own IP address) and relays L4 payloads

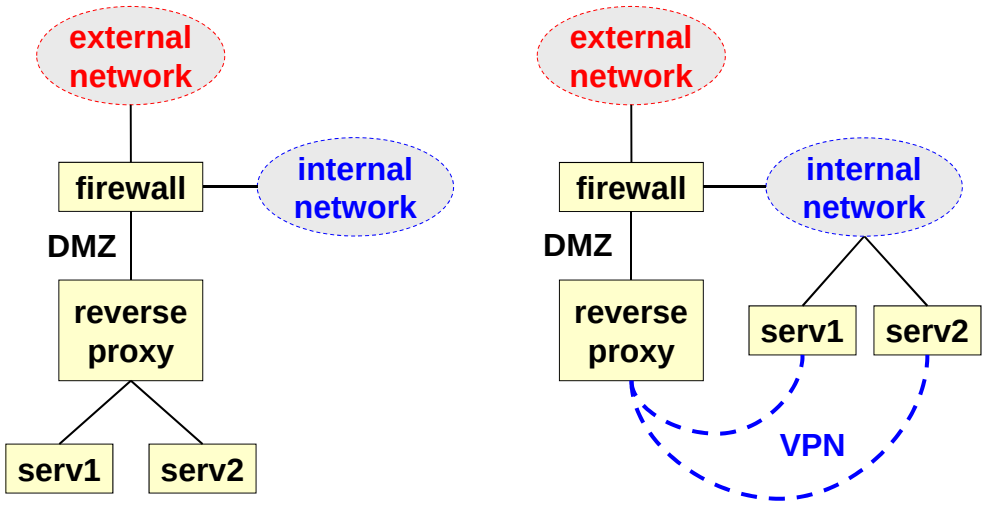
SOCKS: initial problems

- **SOCKS v4:**
  - doesn't distinguish the internal and external nets
  - weak user authentication (based on `identd` or id stored locally on client)
  - supports only TCP
- **solution = SOCKS v5:**
  - supports UDP too
  - better authentication (user+pwd or GSS-API)
  - encryption (between the SOCKS client and server)

Reverse proxy

- **HTTP server acting just as a front-end for the real server(s) which the requests are passed to**
- **benefits:**
  - obfuscation (no info about the real server)
  - load balancer
  - SSL accelerator (with unprotected back-end connections …)
  - web accelerator (=cache for static content)
  - compression
  - spoon feeding (gets from the server a whole dynamic page and feeds it to the client according to its speed, so unloading the application server)

Reverse proxy: possible configurations



FW architectures: which is the right one?

- **theoretically, the higher the level and:**
  - the higher the CPU cycles needed
  - the higher the protection level
- **the (sad) reality:**

> Firewall customers once had a vote, and voted in favor of transparency, performance and convenience instead of security; nobody should be surprised by the results.
>
> (Marcus J. Ranum, the “grandfather of firewalls”, firewall wizard mailing list, oct 2000)

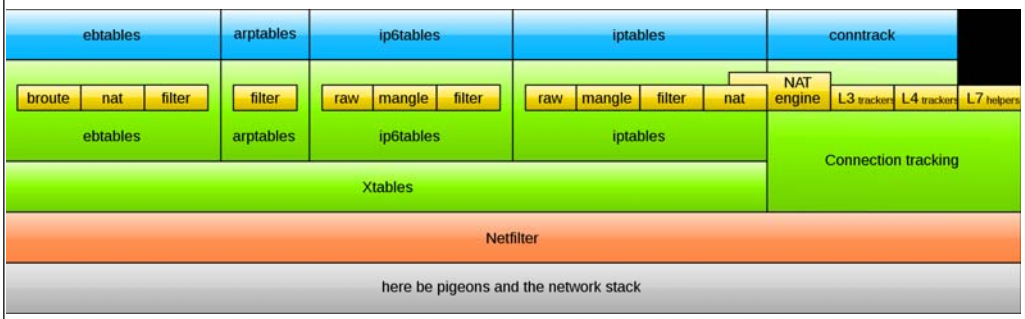
FW architectures: which is the right one?

- **the best choice:**
  - not a single product, but a robust fw architecture that supplements the holes and vulnerabilities of the single components!!!
  - for each component choose something available on several architectures: it’s better to be able to choose rather than leaving the choice to the vendor!!
  - beware of solutions promising to solve each and every security problem: may be it’s just advertisement …

Firewall: commercial products

- **there is plenty of firewall manufacturers/vendors**
- **tipically on UNIX, sometimes on Windows (the latter requires changing the TCP/IP stack!)**
- **the free Firewall Toolkit (FWTK)**
  - TIS (www.tis.com)
  - base application-gateway components
- **the free IPchains / IPfilter / IPtables on Linux**
  - packet-filter

Linux: netfilter components

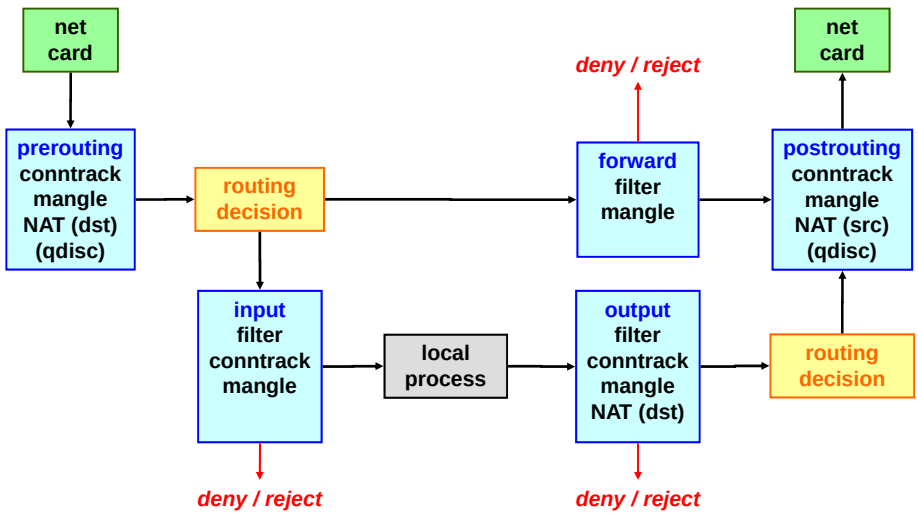


(image from wikipedia)

Default netfilter chains

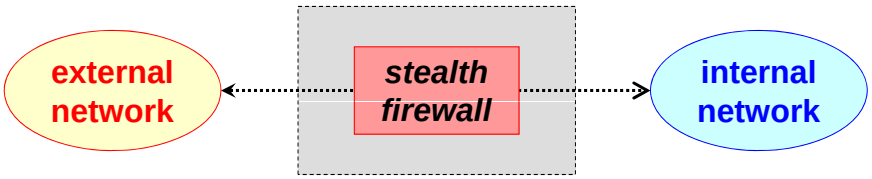
- **PREROUTING**
  - all incoming packets, before any routing decision
- **INPUT**
  - packets with destination the node itself (the "local-delivery" routing table: "ip route show table local")
- **FORWARD**
  - packets to be forwarded, after the routing decision
- **OUTPUT**
  - packets generated by the node itself
- **POSTROUTING**
  - all outgoing packets, just before sending them

Netfilter / iptables: packet flow



Stealth firewall

- firewall without an IP address, so that it cannot be directly attacked
- physical packet interception (by setting the interfaces in promiscuous mode)

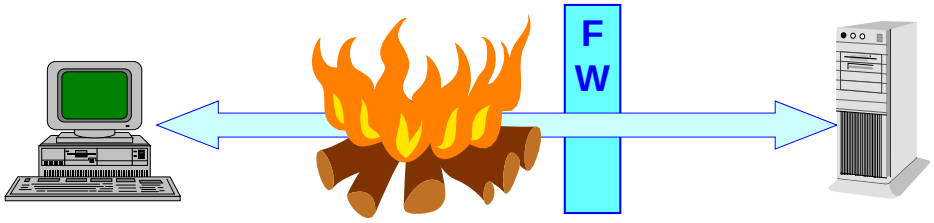


Local / personal firewall

- firewall directly installed at the node to be protected
- typically a packet filter
- w.r.t. a normal network firewall, it may limit the programs than are allowed:
  - to open network channels towards other nodes (i.e. act as a client)
  - to answer network requests (i.e. act as a server)
- important to limit the diffusion of malware and trojans, or plain configuration mistakes
- firewall management must be separated from the system management

Protection offered by a firewall

- **a firewall is 100% effective only for attacks over/against blocked channels**
- **the other channels require other protection techniques:**
  - VPN
  - “semantic” firewall / IDS
  - application-level security



Intrusion Detection System (IDS)

- **definition:**
  - system to identify individuals using a computer or a network without authorization
  - extendable to identify authorized users violating their privileges
- **hypothesis:**
  - the behavioural “pattern” of non-authorized users differs from that of authorized users

IDS: functional features

- **passive IDS:**
  - cryptographic checksum (e.g. tripwire)
  - pattern matching (“attack signature”)
- **active IDS:**
  - “learning” = statistical analysis of the system behaviour
  - “monitoring” = active statistical info collection of traffic, data, sequences, actions
  - “reaction” = comparison against statistical parameters (reaction when a threshold is exceeded)

IDS: topological features

- **HIDS (host-based IDS)**
  - log analysis (OS, service or application)
  - internal OS monitoring tools
- **NIDS (network-based IDS)**
  - network traffic monitoring tools

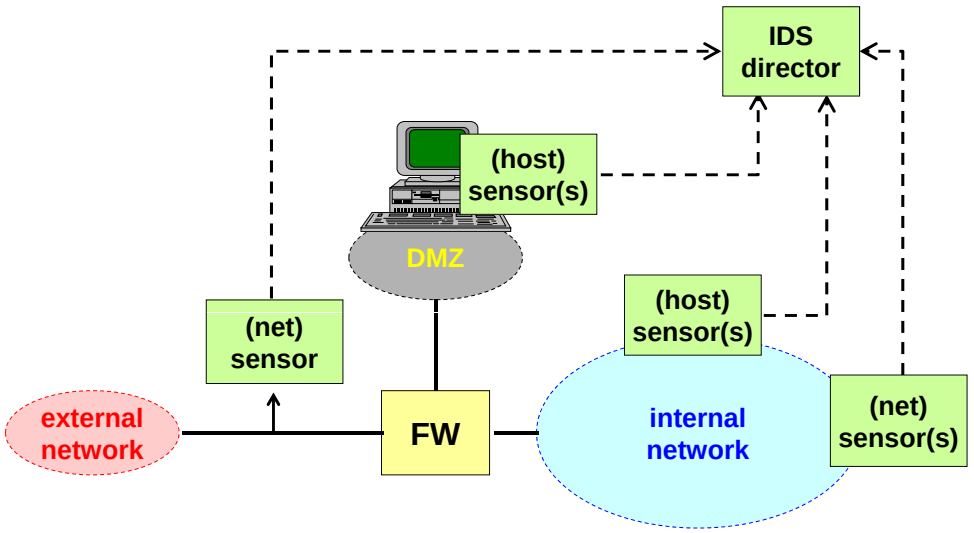
SIV and LFM

- **System Integrity Verifier**
  - checks files / filesystems looking for changes
  - e.g. changes to Windows registry, cron configuration, user privileges
  - e.g. tripwire
- **Log File Monitor**
  - checks the log files (OS and applications)
  - looks for known patterns of successful attacks or attempts
  - e.g. swatch

NIDS components

- **sensor**
  - checks traffic and logs looking for suspect patterns
  - generated the relevant security events
  - interacts with the system (ACLs, TCP reset, ...)
- **director**
  - coordinates the sensors
  - manages the security database
- **IDS message system**
  - secure and reliable communication among the IDS components

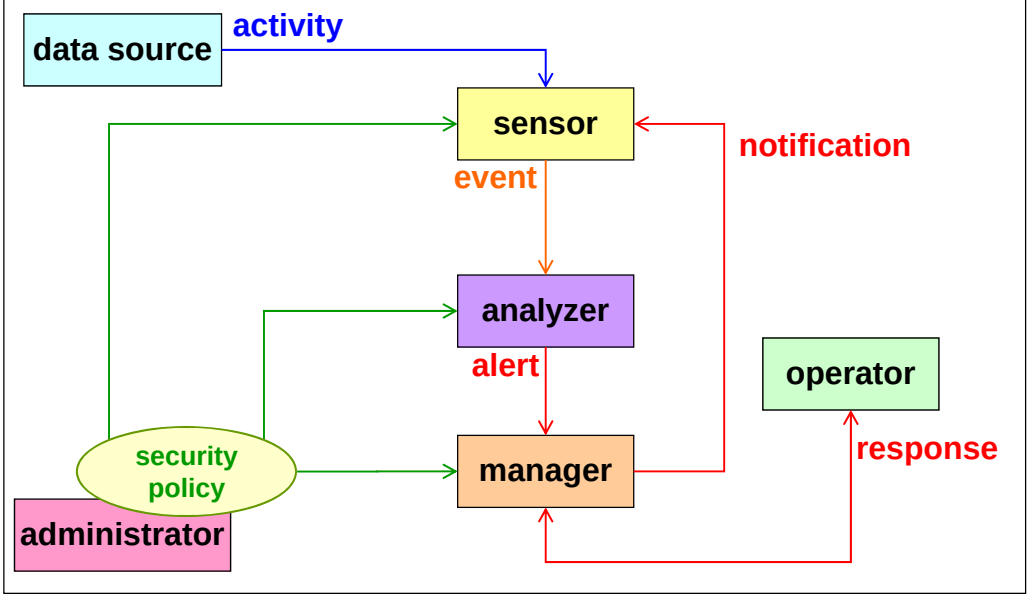
NIDS architecture



IDS/NIDS interoperability

- **much needed because the attacks involve different organizations and/or are detected by different tools**
- **attack signature format:**
  - no standard, but Snort format is in large use
- **alarm format and protocol for alarm transmission:**
  - IDMEF + IDXP + IODEF (IETF)
  - SDEE (Cisco, ISS, SourceFire)

NIDS data flow



IDMEF + IDXP

- **developed by the IETF**
- **Intrusion Detection Message Exchange Format**
  - indipendent from the protocol (IPv4, IPv6)
  - supports internationalization and localization
  - supports data aggregation and filtering (on the manager)
- **Intrusion Detection eXchange Protocol**
  - based on BEEP (RFC-3080)
  - profile exchange (for end-to-end security, for ID)
  - base security profile is TLS

SDEE

- **Secure Device Event Exchange**
- **based on the webservice paradigm:**
  - messages in XML
  - messages transported over HTTP or HTTPS
- **closed standard (?), managed by the ICSAlabs**

IODEF

- **Incident Object Description and Exchange Format**
- **is a superset of IDMEF**
- **used to exchange information between different organizations, keep statistics, evaluate risks, ...**

IPS

- Intrusion Prevention System
- to speed-up and automate the reaction to intrusions = IDS + distributed dynamic firewall
- a technology, not a prudoct, with large impact on many elements of the protection system
- dangerous! may take the wrong decision and block innocent traffic

Honey pot