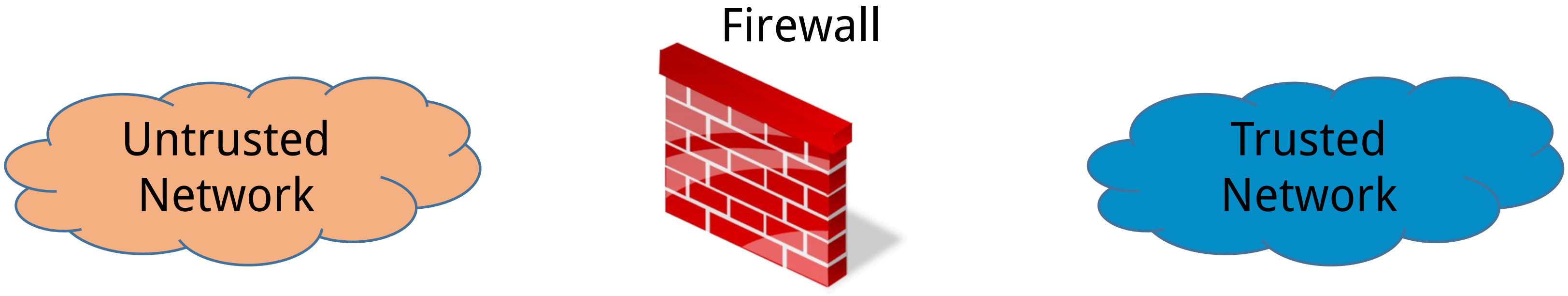
# CSE 410/565: Computer Security

Instructor: Dr. Ziming Zhao

# Firewall

A component or set of components that restricts access between a protected network and the Internet, or between other sets of networks



Untrusted Network

Firewall

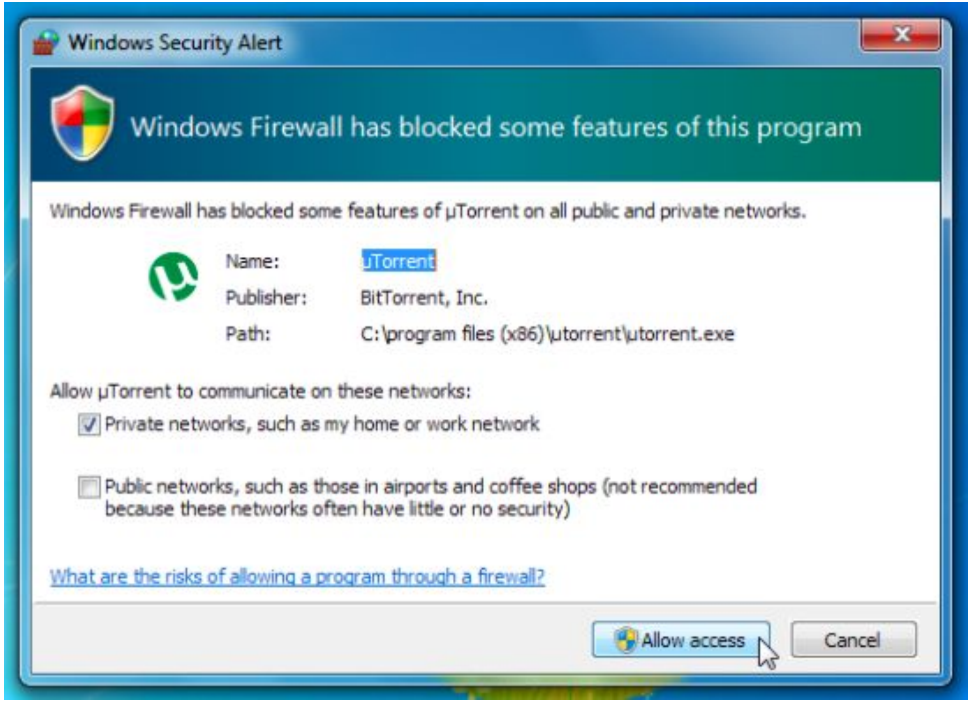
Trusted Network

# Firewall

- A **choke point** of control and monitoring
- Interconnects networks with differing *trust*
- Imposes restrictions on network services
  - only authorized traffic is allowed
- Auditing and controlling access
  - can implement alarms for abnormal behavior
- Itself immune to penetration

- Provides perimeter defence

# Host-based Firewall



Windows Firewall has blocked some features of this program

Windows Firewall has blocked some features of μTorrent on all public and private networks.

Name: μTorrent

Publisher: BitTorrent, Inc.

Path: C:\program files (x86)\utorrent\utorrent.exe

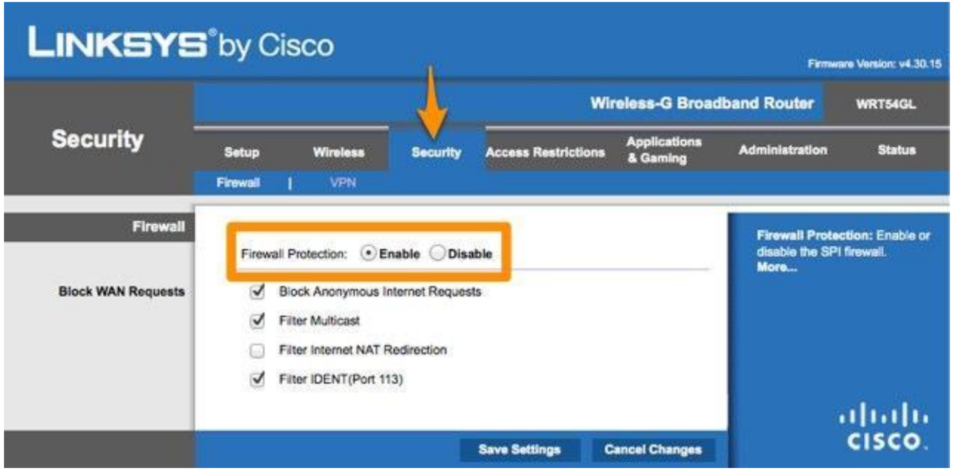
Allow μTorrent to communicate on these networks:

☑ Private networks, such as my home or work network

□ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

What are the risks of allowing a program through a firewall?
Allow access
Cancel

# Network Firewall



**Firewall**

Firewall Protection: ● Enable ○ Disable

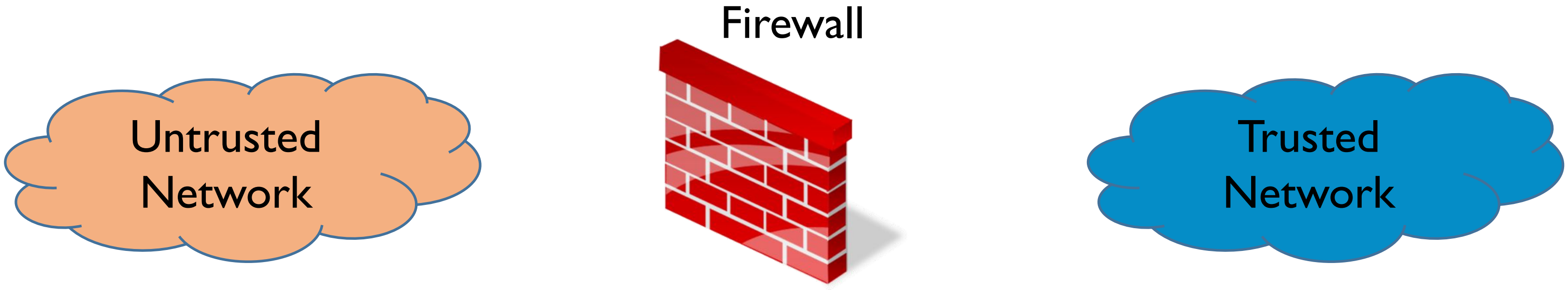
**Block WAN Requests**

- ☑ Block Anonymous Internet Requests
- ☑ Filter Multicast
- □ Filter Internet NAT Redirection
- ☑ Filter IDENT(Port 113)

**Firewall Protection:** Enable or disable the SPI firewall.
**More...**

# Firewall

- Analyze inbound and outbound packets against rules and decide to *block* or *allow* the packet based upon those rules



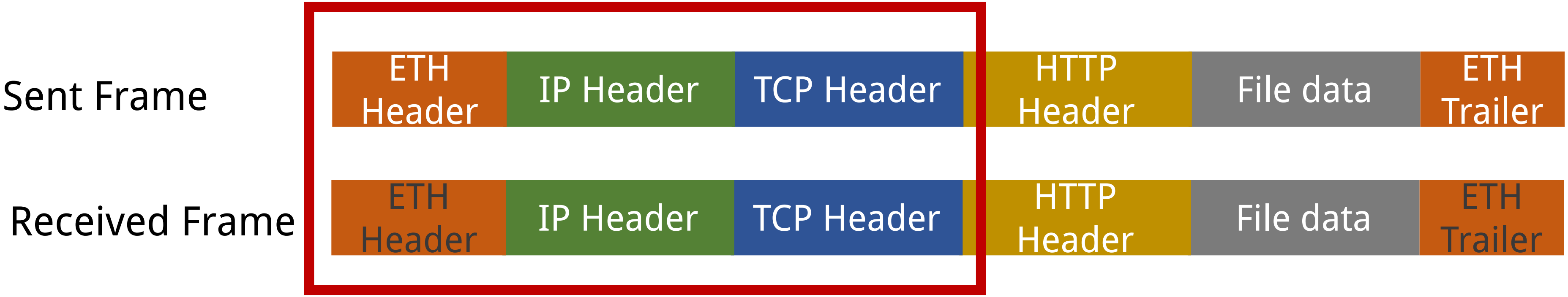
# Configure a Firewall

- Security policy – Firewall rules
- Rule
    - Specify allowable packets in terms of logical expressions on packet fields

# Firewall - *Packet Filters*

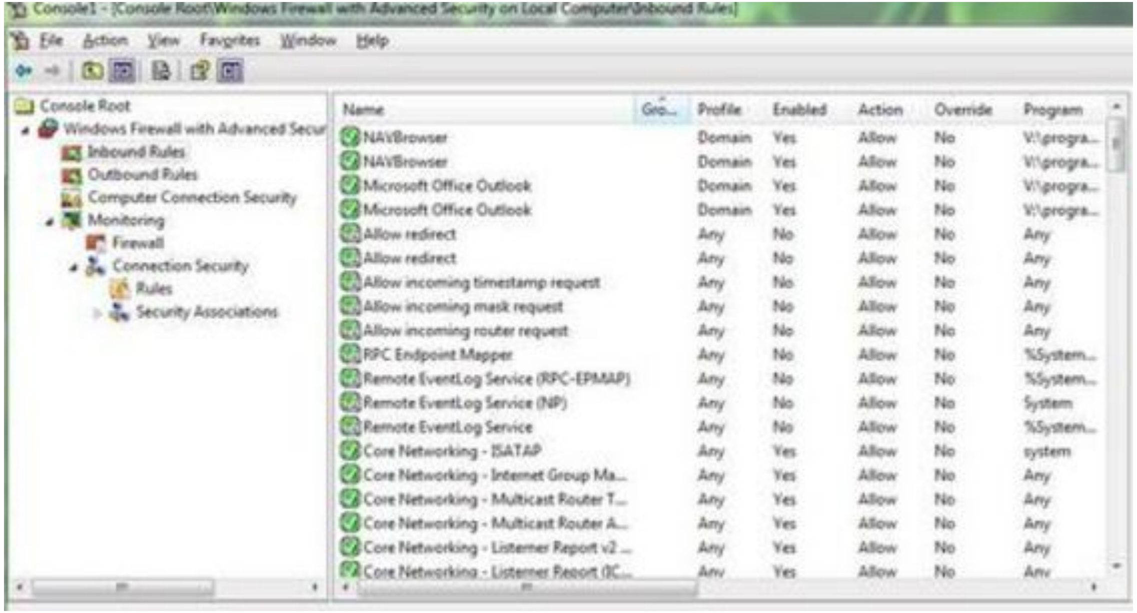
- Simplest
- Uses transport-layer and below information only
  - IP Source Address, Destination Address
  - Protocol/Next Header (TCP, UDP, ICMP, etc)
  - TCP or UDP source & destination ports
  - TCP Flags (SYN, ACK, FIN, RST, PSH, etc)
  - ICMP message type

# Packet Filters



# Packet Filters - Rules

- First-match
- Default ‘deny’



Console1 - [Console Root\Windows Firewall with Advanced Security on Local Computer\Inbound Rules]
File
Action
View
Favorites
Window
Help
Console Root
Windows Firewall with Advanced Secur
Inbound Rules
Outbound Rules
Computer Connection Security
Monitoring
Firewall
Connection Security
Rules
Security Associations

| Name | Gro... | Profile | Enabled | Action | Override | Program |
|---|---|---|---|---|---|---|
| NAVBrowser | | Domain | Yes | Allow | No | V:\progra... |
| NAVBrowser | | Domain | Yes | Allow | No | V:\progra... |
| Microsoft Office Outlook | | Domain | Yes | Allow | No | V:\progra... |
| Microsoft Office Outlook | | Domain | Yes | Allow | No | V:\progra... |
| Allow redirect | | Any | No | Allow | No | Any |
| Allow redirect | | Any | No | Allow | No | Any |
| Allow incoming timestamp request | | Any | No | Allow | No | Any |
| Allow incoming mask request | | Any | No | Allow | No | Any |
| Allow incoming router request | | Any | No | Allow | No | Any |
| RPC Endpoint Mapper | | Any | No | Allow | No | %System... |
| Remote EventLog Service (RPC-EPMAP) | | Any | No | Allow | No | %System... |
| Remote EventLog Service (NP) | | Any | No | Allow | No | System |
| Remote EventLog Service | | Any | No | Allow | No | %System... |
| Core Networking - ISATAP | | Any | Yes | Allow | No | system |
| Core Networking - Internet Group Ma... | | Any | Yes | Allow | No | Any |
| Core Networking - Multicast Router T... | | Any | Yes | Allow | No | Any |
| Core Networking - Multicast Router A... | | Any | Yes | Allow | No | Any |
| Core Networking - Listener Report v2 ... | | Any | Yes | Allow | No | Any |
| Core Networking - Listener Report (IC... | | Any | Yes | Allow | No | Any |

# Packet Filters

- DNS uses port 53
  - No incoming port 53 packets except known trusted servers

| Src IP | Src Port | Dst IP | Dst Port | Action? |
|---|---|---|---|---|
| 8.8.8.8 | 53 | * | * | Allow |

# Packet Filters

Suppose we want to allow inbound mail (SMTP, port 25) but only to our gateway machine.

Also suppose that mail from some particular site MALSITE is to be blocked.

| Their IP | Their Port | Our IP | Our Port | Action? |
|---|---|---|---|---|
| MALSITE | * | * | * | Deny |
| * | * | GW | 25 | Allow |

# Packet Filters

Suppose that we want to implement the policy “any inside host can send mail (but nothing else) to the outside”.

| Their IP | Their Port | Our IP | Our Port | Action? |
|---|---|---|---|---|
| * | 25 | * | * | Allow |

Our defined restriction is based solely on the outside host’s port number, which we have no way of controlling.
Now an attacker can access any internal machines and port by originating his call from port 25 on the outside machine.

# Packet Filters

Suppose that we want to implement the policy “any inside host can send mail to the outside”.

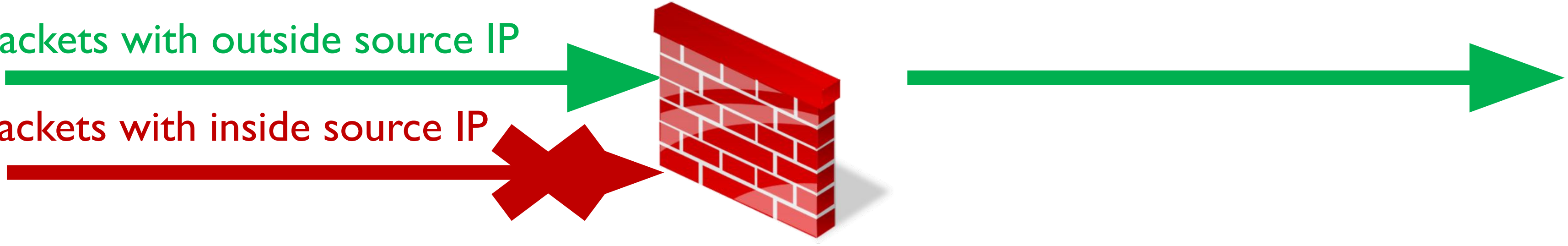
| Src IP | Src Port | Dst IP | Dst Port | TCP Flags | Action? |
|---|---|---|---|---|---|
| {Our hosts} | * | * | 25 | * | Allow |
| * | 25 | * | * | ACK | Allow |

The ACK signifies that the packet is part of an ongoing conversation.

Packets without the ACK are connection establishment messages, which we are only permitting from internal hosts.

# Defend Against Some IP Spoofing

- The FW blocks packets from outside the network with a source address inside the network. This prevents an outside attacker spoofing the address of an internal machine.



Outside world

Firewall – Packet Filter

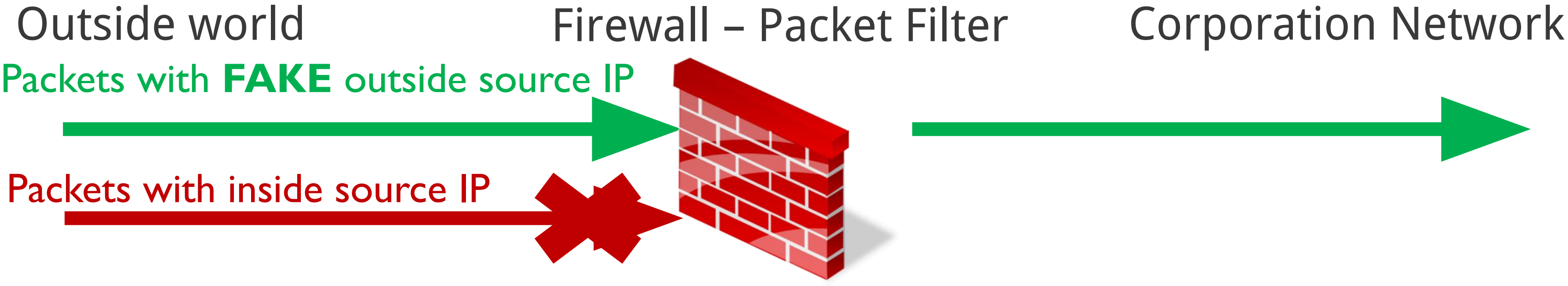
Corporation Network

Packets with outside source IP

Packets with inside source IP

# Defend Against Some IP Spoofing

- The FW blocks packets from outside the network with a source address inside the network. This prevents an outside attacker spoofing the address of an internal machine.



Outside world

Firewall – Packet Filter

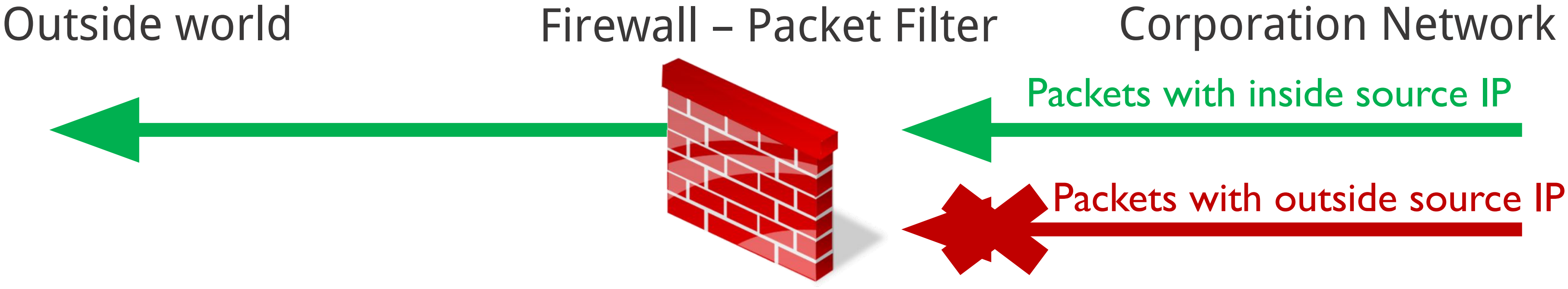
Corporation Network

Packets with **FAKE** outside source IP

Packets with inside source IP

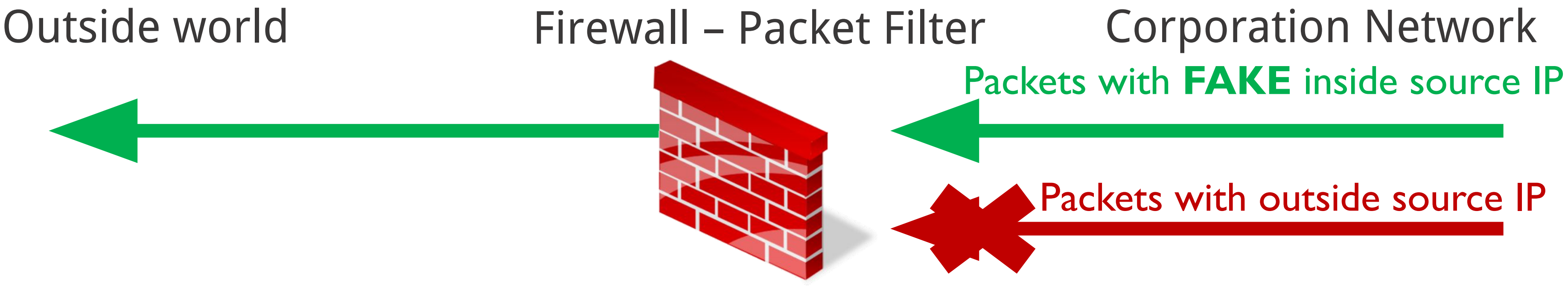
# Defend Against Some IP Spoofing

- The FW would also blocks packets from inside the network with a source address that is not inside. This prevents an attacker within the network performing filtering from launching IP spoofing attacks against external machines.



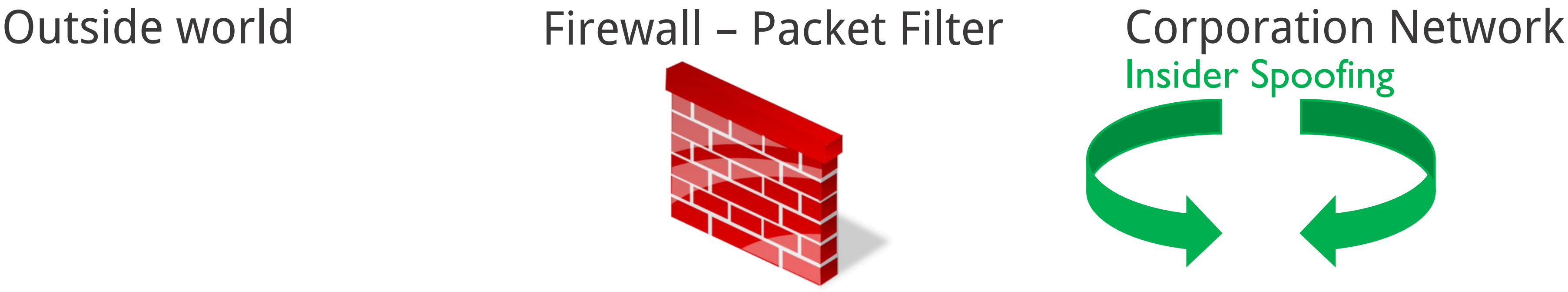
# Defend Against Some IP Spoofing

- The FW would also blocks packets from inside the network with a source address that is not inside. This prevents an attacker within the network performing filtering from launching IP spoofing attacks against external machines.


Outside world
Firewall – Packet Filter
Corporation Network
Packets with **FAKE** inside source IP
Packets with outside source IP

# Defend Against Some IP Spoofing

- It does not work if an inside host is compromised and it tries to spoof another inside host.



Outside world

Firewall – Packet Filter

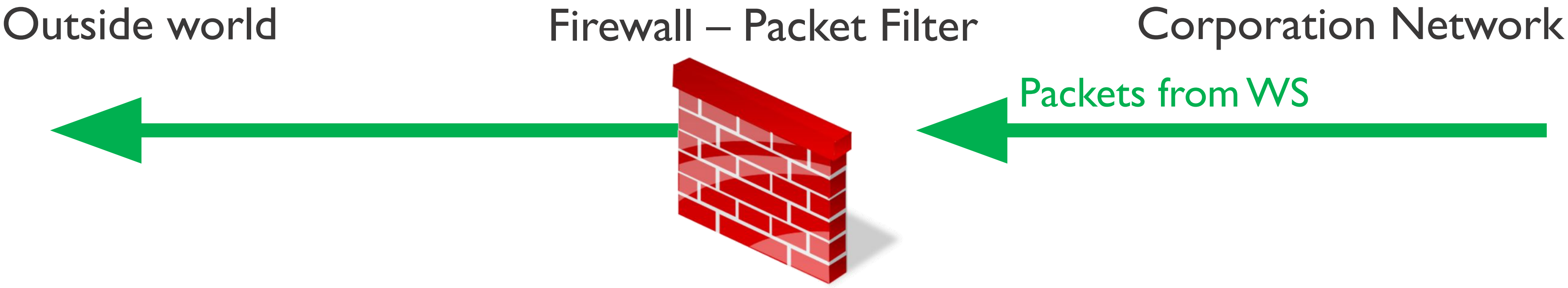
Corporation Network

Insider Spoofing

# Packet Filters – Another Example

Suppose we have a web server (WS). We want to apply ‘packets from/to the web server are allowed only if an outside host initiated the connection’

| Their IP | Their Port | Our IP | Our Port | Action? |
|---|---|---|---|---|
| * | * | WS | 80 | Allow |



Outside world

Firewall – Packet Filter

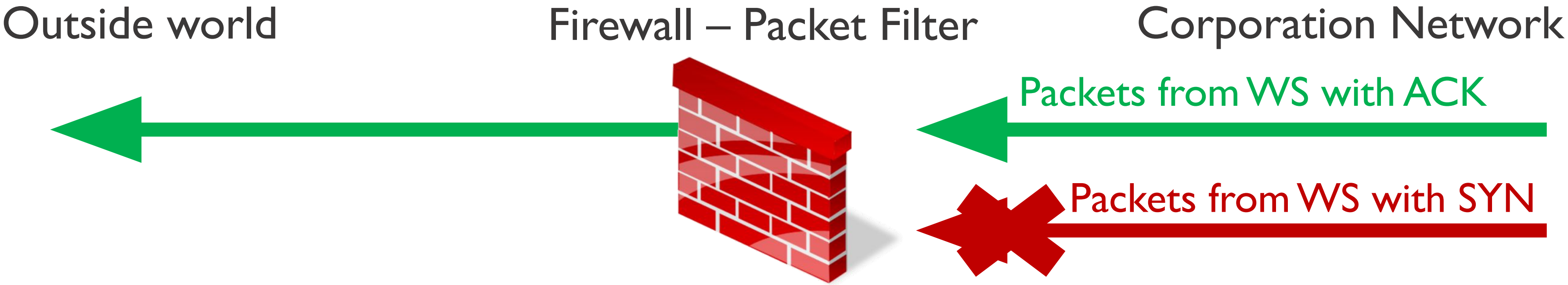
Corporation Network

Packets from WS

# Packet Filters – Another Example

Suppose we have a web server (WS). We want to apply ‘packets from/to the web server are allowed only if an outside host initiated the connection’

| Src IP | Src Port | Dst IP | Dst Port | TCP Flags | Action? |
|---|---|---|---|---|---|
| * | * | WS | 80 | SYN / ACK | Allow |
| WS | 80 | * | * | ACK | Allow |



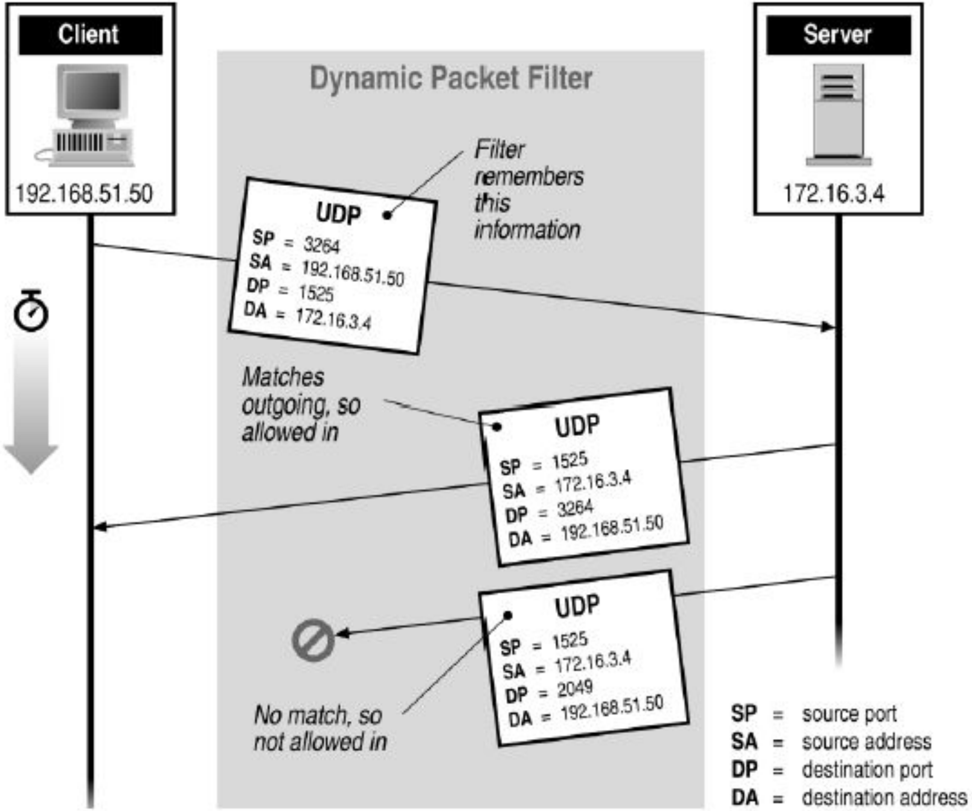
# Packet Filters – Stateless

- It does not remember the TCP connection states or any other information from any packet.

- It only looks at each packet.

- It cannot correlate packets.

# Stateful Firewall

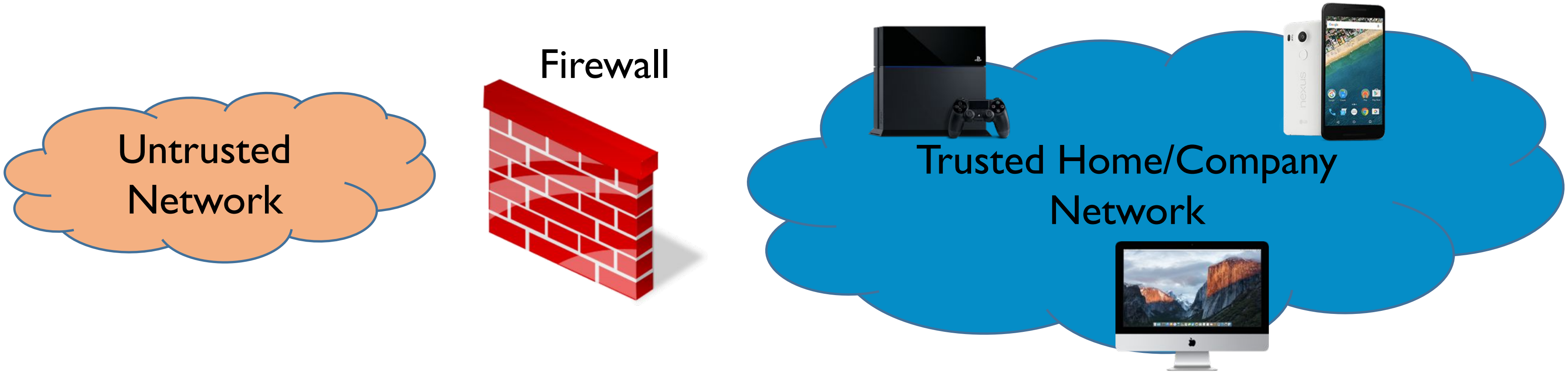


# Different Firewall Needs

Hosts in a trusted company/home network have distinct network needs

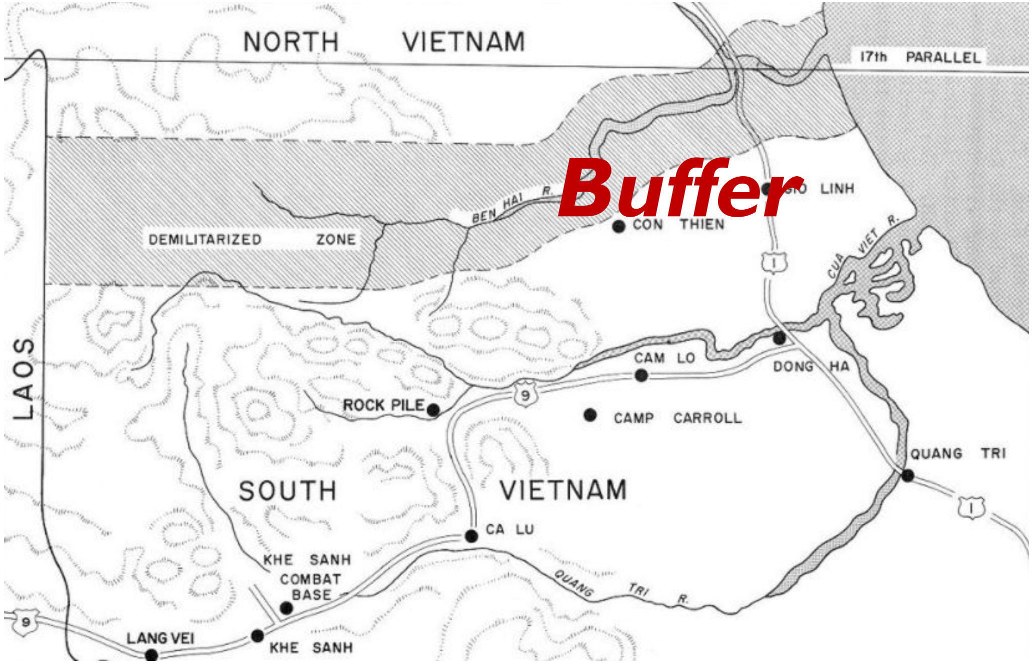
- Access by anyone to public data concerning the company
- Access only by employees to internal data

Solution: inner and outer (DMZ) networks



# Demilitarized Zone (DMZ)

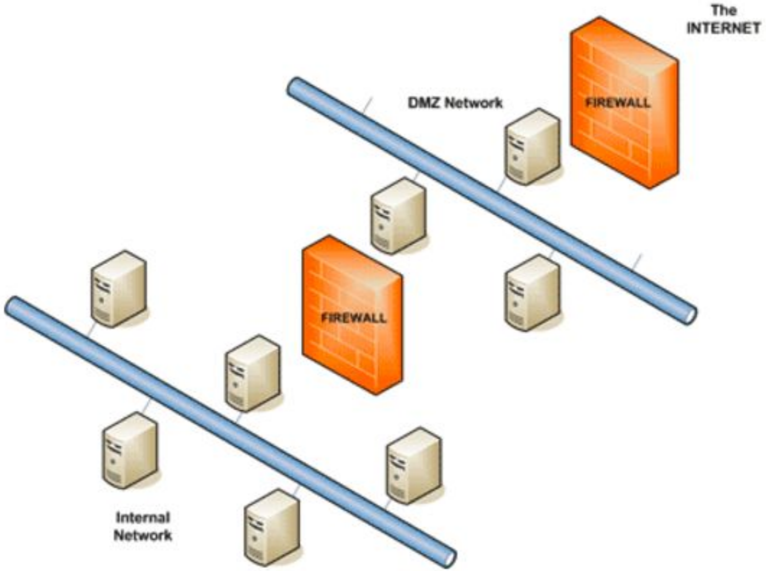
In international affairs, a demilitarized zone is an area in which treaties or agreements between nations, military powers or contending groups *forbid military installations*, activities or personnel.



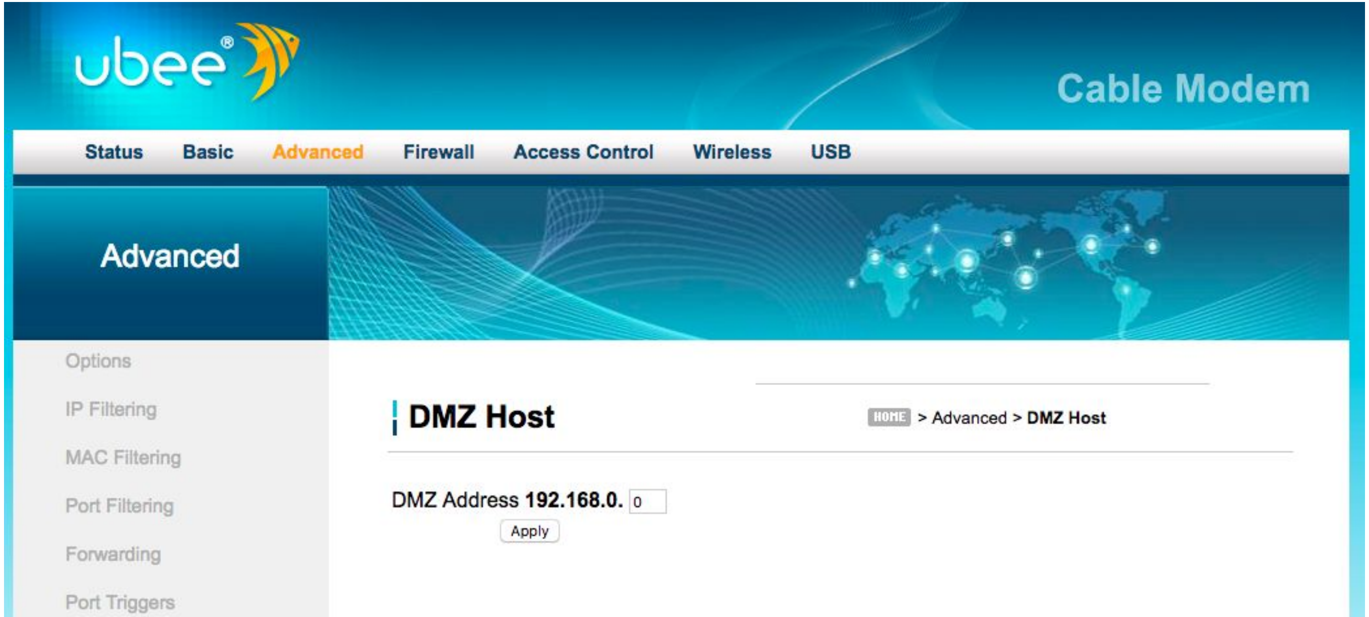
# DMZ

In computer networking, the DMZ likewise provides a *buffer zone* that separates an internal network from the often hostile territory of the Internet.

- Screened subnet
- Perimeter network



# DMZ



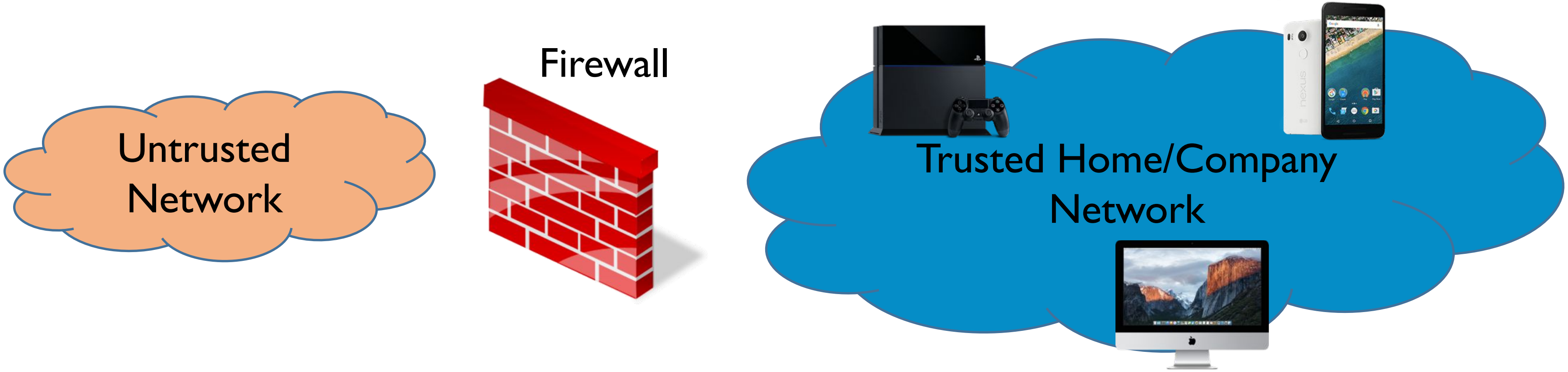
DMZ Host

DMZ Address 192.168.0. 0

# DMZ

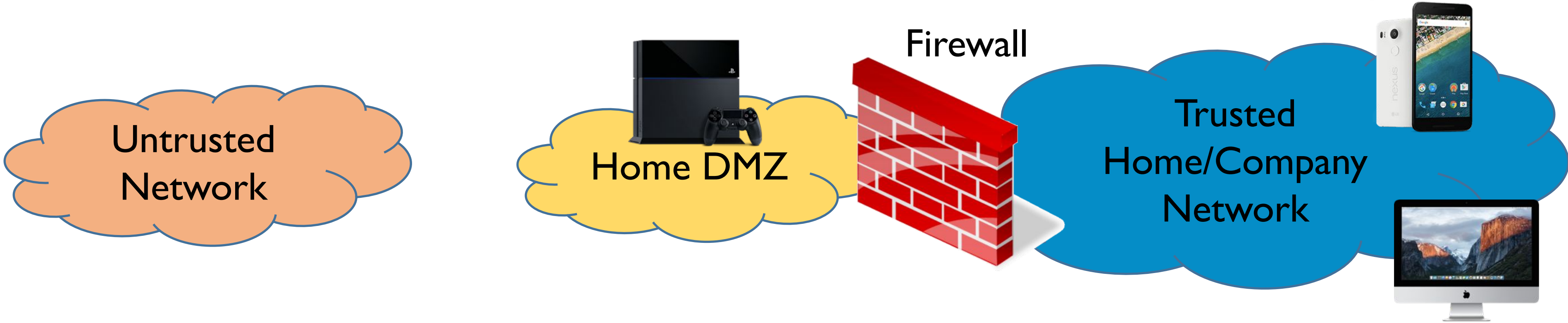
PlayStation

- Not being able to join rooms
- Not being able to speak over headset
- Not being able to hear over headset
- Not being able to join games which are hosted by certain people



# DMZ

1. Assign your PS3 a static IP address
2. Set that IP address as a DMZ host



# Firewall: First Line of Defense

It’s like living in a gated community.

Don’t let the firewall give you a false sense of security.

# Firewall: First Line of Defense

- Does not look at all fields

- IP address spoofing
  - Fake source address to be trusted

- Tiny fragment attacks
  - Split TCP header info over several tiny packets
  - Solution: either discard or reassemble before check

# To log or not to log

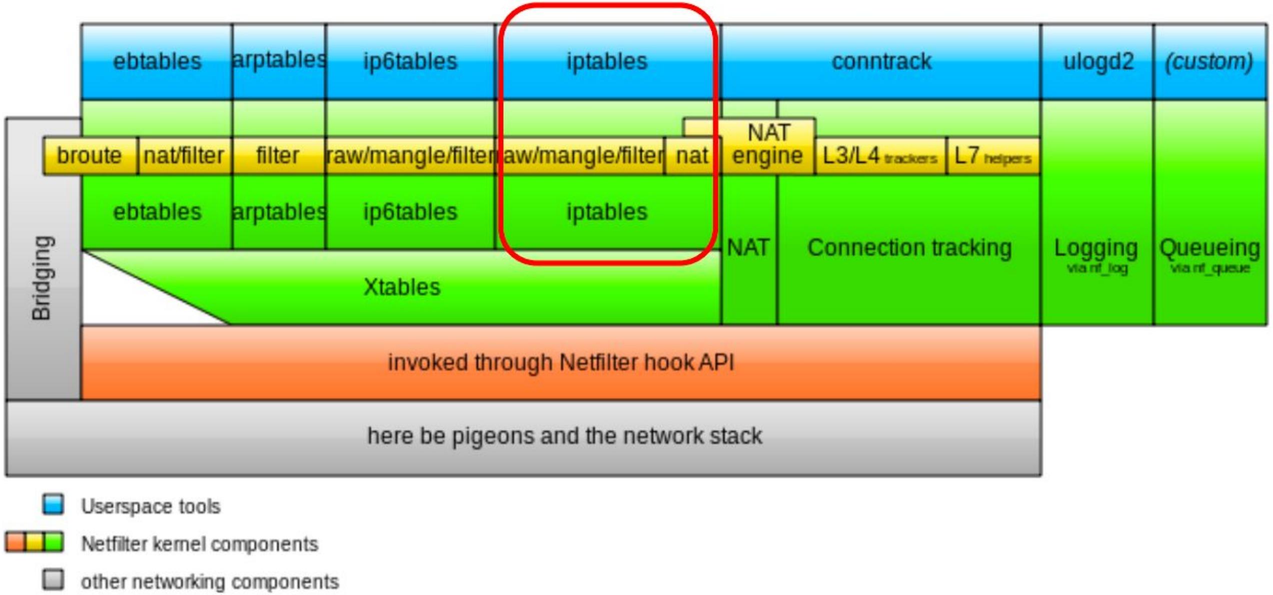
Logging is both good and bad.

If you set your rules to log too much, your logs will not be examined.  If you log too little, you won’t see things you need.  If you don’t log, you have no information on how your firewall is operating.

# iptables

It is a Linux stateful packet filtering firewall.

- It interfaces to the Linux *netfilter kernel module* to perform filtering of network packets based on a ruleset.



ebtables | arptables | ip6tables | iptables | conntrack | ulogd2 | *(custom)*

broute | nat/filter | filter | raw/mangle/filter | aw/mangle/filter | nat | NAT engine | L3/L4 trackers | L7 helpers

ebtables | arptables | ip6tables | iptables | NAT | Connection tracking | Logging via nf_log | Queueing via nf_queue

Bridging

Xtables

invoked through Netfilter hook API

here be pigeons and the network stack

🟦 Userspace tools

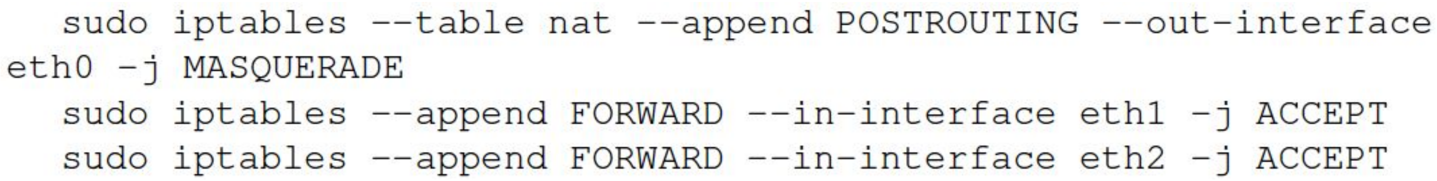
🟧🟨🟩 Netfilter kernel components

⬜ other networking components

# iptables

It is a stateful packet filtering firewall.

- It interfaces to the Linux *netfilter kernel module* to perform filtering of network packets based on a ruleset.



```
    sudo iptables --table nat --append POSTROUTING --out-interface
eth0 -j MASQUERADE
    sudo iptables --append FORWARD --in-interface eth1 -j ACCEPT
    sudo iptables --append FORWARD --in-interface eth2 -j ACCEPT
```

iptables – userspace utility

# iptables History

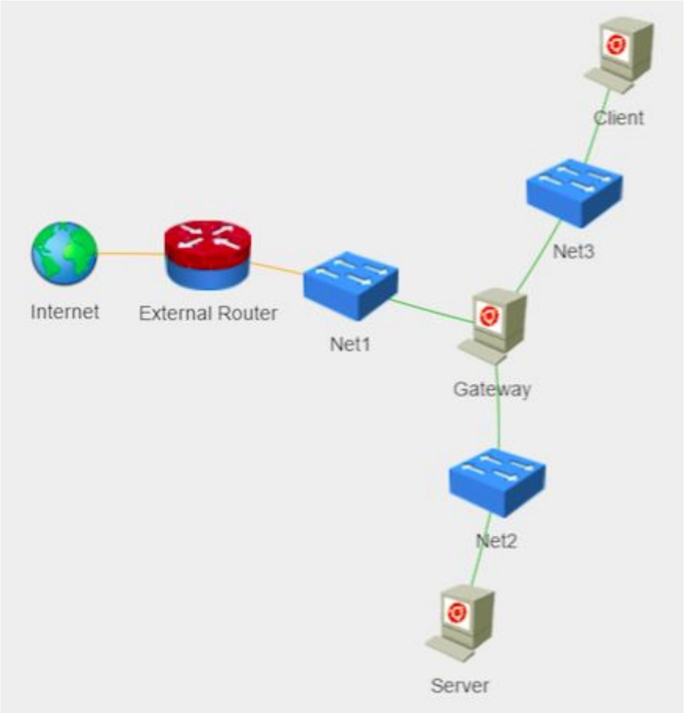
- Ipfwadm: Linux kernel 2.0.34
- Ipchains: Linux kernel 2.2.*
- Iptables: Linux kernel 2.4.*

- The core team was actually started in November 1999

- *nftables* is supposed to replace *netfilter*. (Released with Linux 3.13, Jan 2014)
- *nftables* is configured via the userspace utility *nft,* while netfilter is configured via the utilities *iptables*, *ip6tables*, *arptables* and *ebtables* frameworks.

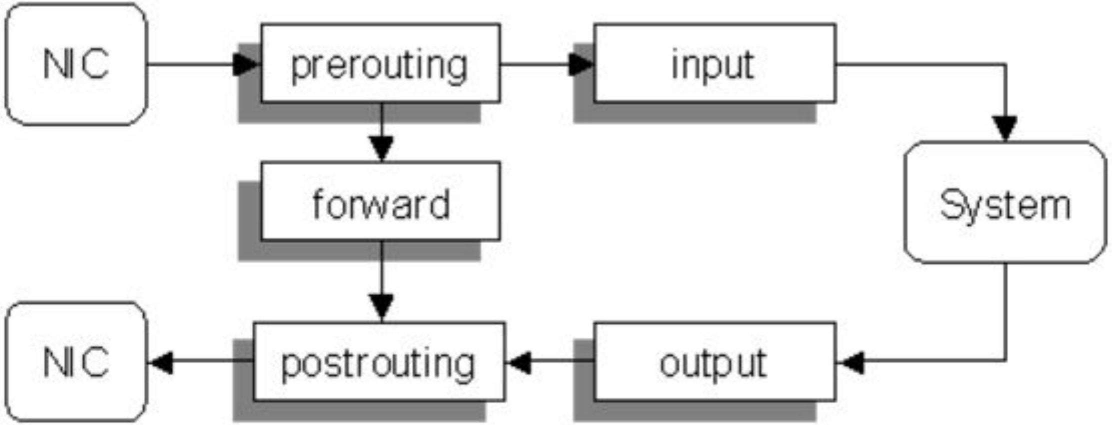
# iptables

- Host-based firewall (Edge firewall) ?
- Network firewall (net.ipv4.ip_forward = 1)



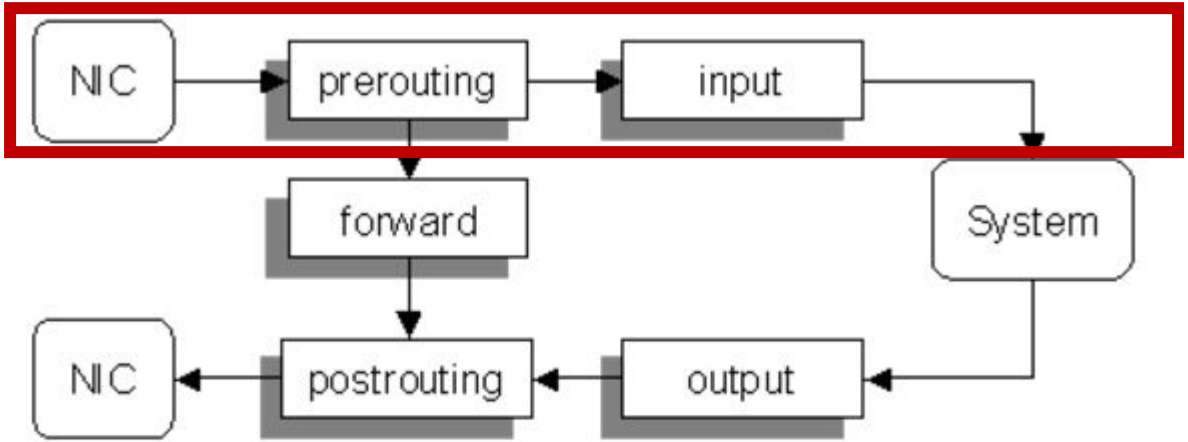
# Five Build-in Chains

There are five built-in chains, one for each point in the kernel’s processing path:



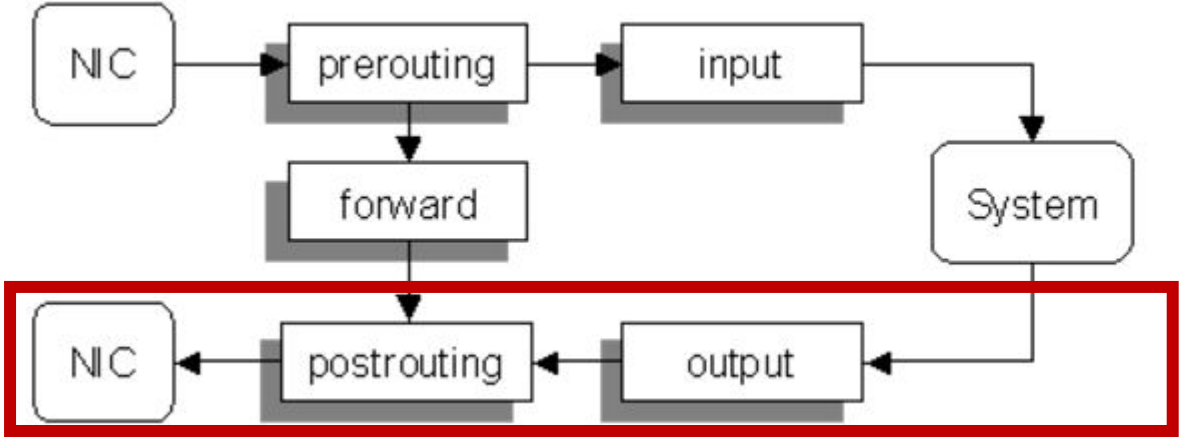
# Five Build-in Chains

The chain determines when a packet is examined. An incoming packet (from the outside to this host) arrive at a NIC and will be processed only by the rules on the **prerouting** and **input** chains.



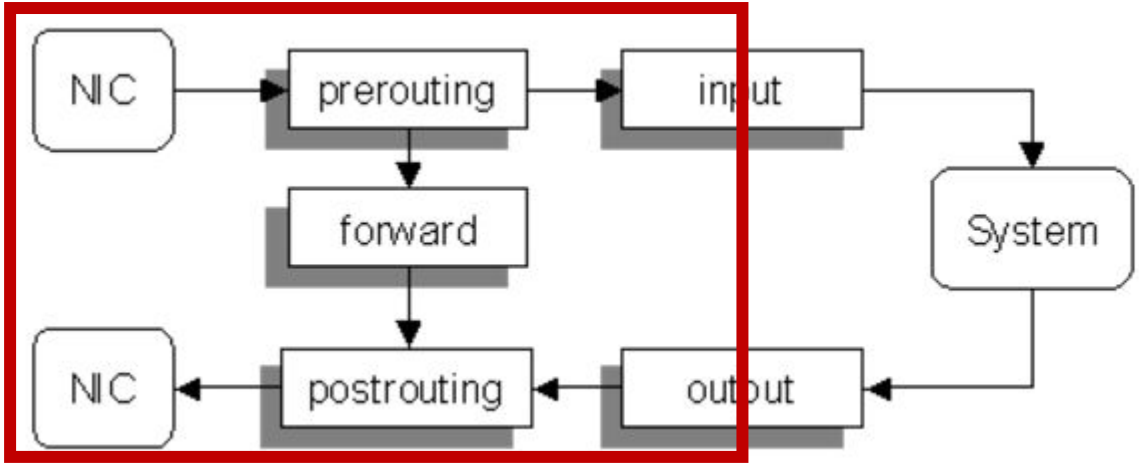
# Five Build-in Chains

Outgoing packets (from this host to another) will use the **output** and **postrouting** chains only.



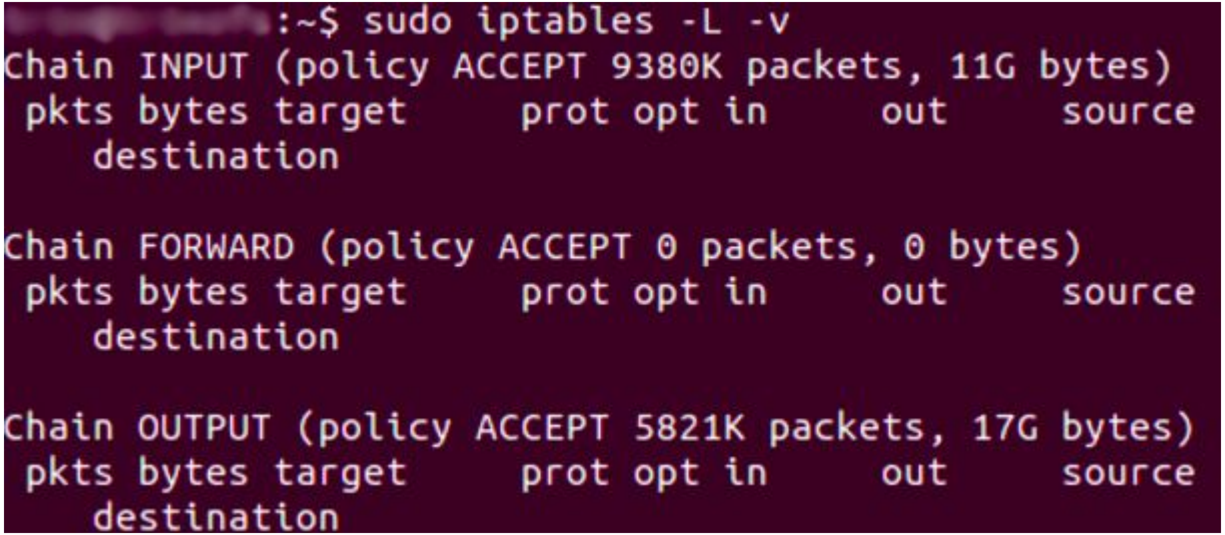
# Five Build-in Chains

When the host acts as a router it will forward some packets (from one interface to another).
These will traverse only the **prerouting**, **forward**, and **postrouting** chains.



# Five Build-in Chains

Some (or all) of these chains may be empty.
You can set a default policy for this standard chains.



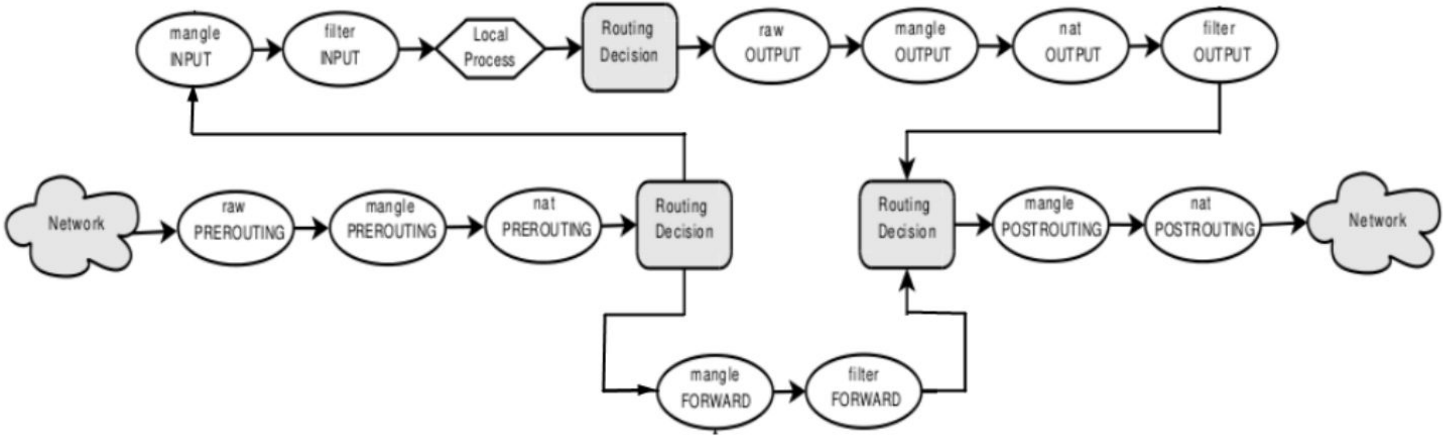
```
[redacted]:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 9380K packets, 11G bytes)
 pkts bytes target     prot opt in     out     source
   destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
   destination

Chain OUTPUT (policy ACCEPT 5821K packets, 17G bytes)
 pkts bytes target     prot opt in     out     source
   destination
```

# Four Tables

A rule's table determines what targets are valid for that rule, as well as to which built-in chains the rule can be added.  Note some targets (e.g., ACCEPT, DROP) can be used in all tables.

- **filter**: rules used for filtering packets. (the default table)
- **nat**: rules used for translating the packet's src and dst field.
- **mangle** table and a **raw** table that allow rules with targets that do other things to packets, such as re-routing them or changing header values (such as the QoS field).

# Detailed Packet Path



# iptables Command

**iptables** [**-t** *table*] **-I|A** *chain* [*rulenum*] *rule-specification*

rule-specification = [*matches...*] [*target*]

target = **-j** *targetname* [*per-target-options*]

If a packet does not match the *matches*, the packet is handed off to the next rule in the chain
If a packet meets the *matches*, then the rule is passed to the target

# iptables Command

**iptables** [**-t** *table*] **-I|A** *chain* [*rulenum*] *rule-specification*

rule-specification = [*matches...*] [*target*]

target = **-j** *targetname* [*per-target-options*]

Target types:
**filter** table: ACCEPT, DROP, REJECT, LOG, RETURN
**nat** table: DNAT, MASQ, REDIRECT, SNAT

# iptables Command

**iptables** [**-t** *table*] **-I|A** *chain* [*rulenum*] *rule-specification*

-I Add a rule to the head of a chain
-A Appends a rule to the tail of a chain
-D Deletes a rule that matches the specifiers
-R Replaces a rule in a chain

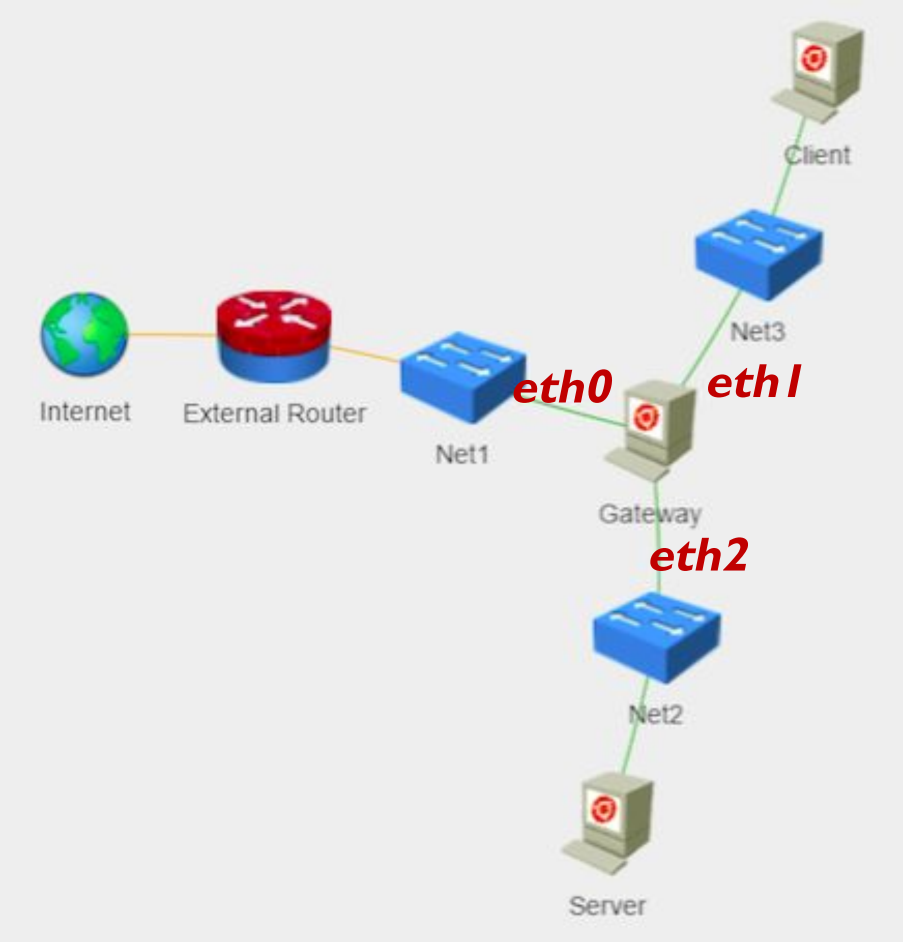
# iptables Command

**iptables** [**-t** *table*] **-I|A** *chain* [*rulenum*] *rule-specification*

Protocol, Source IP, Destination IP, Input Interface, Output interface, Frag,  TCP flags, Connection State

# Example

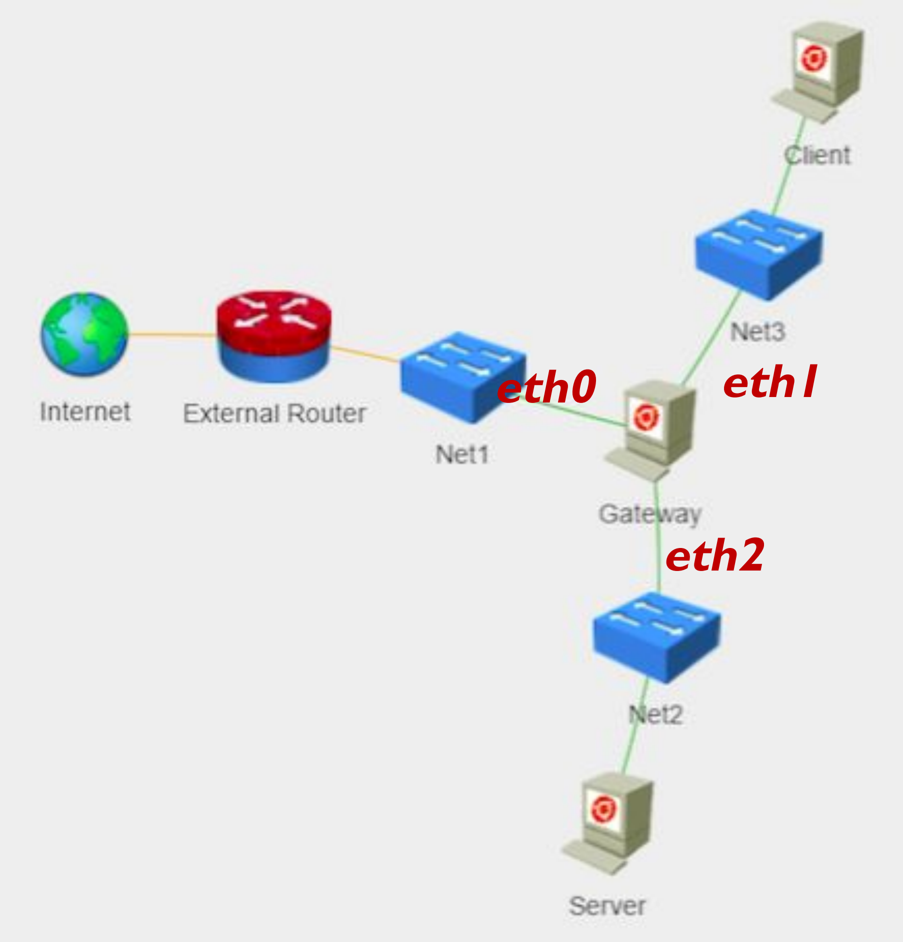
*Gateway* has three network interfaces, whereas *Client* and *Server* have only one.



# Example

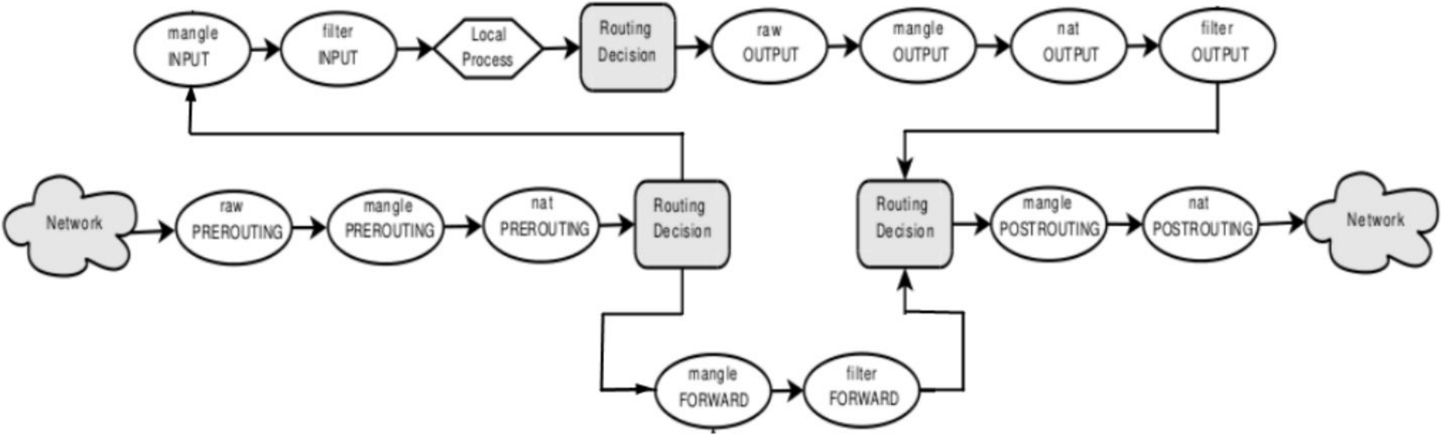
Enable IP forwarding

Enable network address translation on *Gateway, so Client* and *Server* can also talk to the Internet



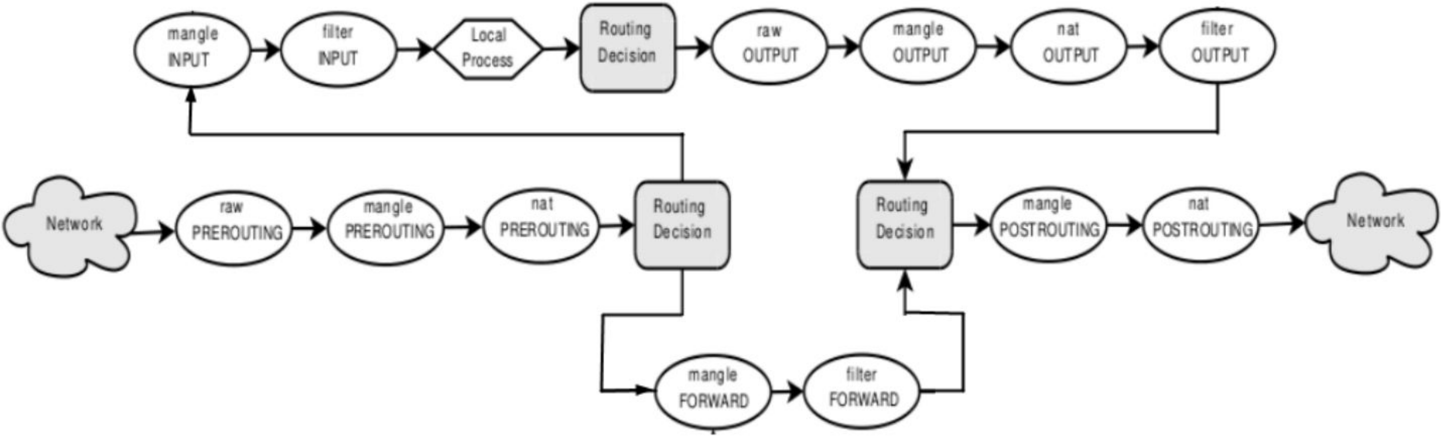
# Example

```
net.ipv4.ip_forward = 1

sudo iptables --table nat --append POSTROUTING --out-interface
eth0 -j MASQUERADE
```



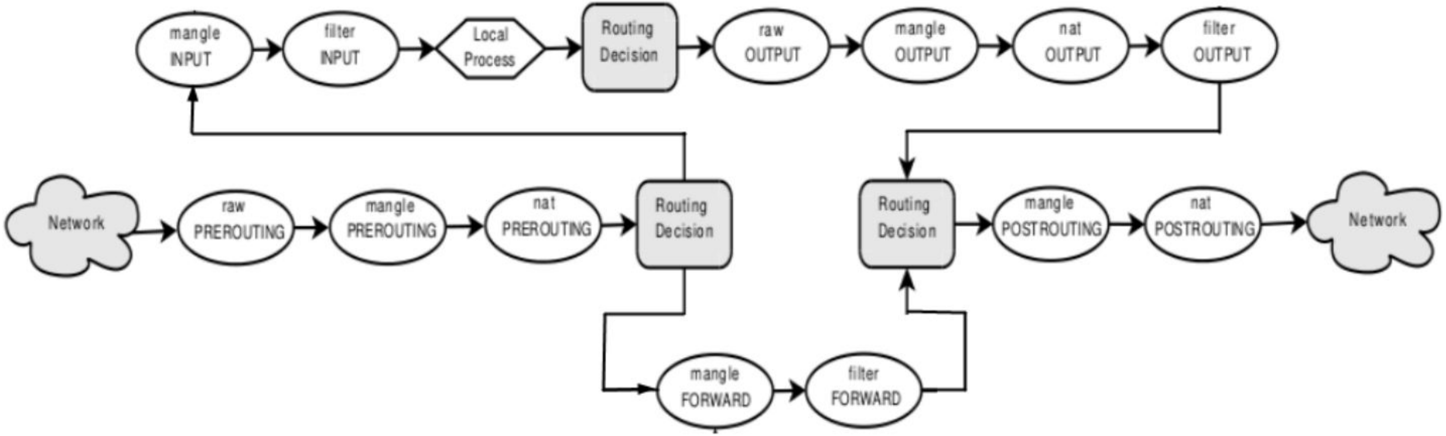
# Example

```
sudo iptables --append FORWARD --in-interface eth1 -j ACCEPT
```



# Example

```
sudo iptables --append FORWARD --in-interface eth2 -j ACCEPT
```



# Intrusion Detectionn

- Intruders
- Intrusion detection
  - Host-based
  - Network-based
  - Hybrid
  - attacks on intrusion detection systems

# Intruders

- Different types of intruders
  - Hackers
  - people who break into computers to gain status within hacking community
  - even benign intruders consume resources and must be stopped
- criminal organizations
  - more determined attackers with a target goal (e.g., to gain access to sensitive or financial data)
  - often act quickly and with fewer mistakes
  - obscure use of stolen financial data to complicate investigation

# Intruders

- Types of intruders
  - Insiders
    - employees who misuse their privileges with or without malice
    - example: access to IRS data by employees, employees who take databases upon leaving an organization
- The goal is to defend against all of the above
- Often a strong barrier is built at the network perimeter
  - firewalls, packet filtering, stricter policies, intrusion detection
  - special precautions must be made to defend against internal threats

# Intruders

- Often the following defenses are used to counter insider intrusion
  - enforce least privilege, permit access only to resources needed for the job
  - use authentication to access sensitive information
  - log accesses and other relevant information
  - upon job termination promptly revoke all privileges
  - when an employee with access to sensitive information leaves, can be useful to store information about their privileges and data for future references in case an accident happens

# Intrusion Detection

- Intrusion detection system (IDS) is a security service that monitors and analyzes system events
- IDS classification
    - host-based IDS
        - monitors events and characteristics of a single host for suspicious activity
    - network-based IDS
        - monitors data on the network for traces of suspicious activity
        - often a single monitor scans data sent to/from many machines on the network
    - hybrid IDS
        - combines information gathered from hosts and network

# Intrusion Detection Systems

- IDSs can be classified based on how they recognize suspicious activity
  - misuse detection (signature based)
    - define what constitutes an intrusion attempt through a set of rules
    - e.g., specific patterns in network traffic, a combination of events
    - can detect only known/encoded intrusion attempts
  - anomaly detection
    - train the system on clean data to understand behavior of legitimate users
    - use it to monitor real data and detect anomalous behavior
    - advantages: more flexible, can detect unknown misuses
    - disadvantages: higher error rate, difficult to tune

# Intrusion Detection Systems

- Intrusion detection is not perfect, two types of errors are
  - false positives: legitimate behavior of authorized users is classified as an intrusion
  - false negatives: an intrusion is not recognized as suspicious activity
- False negatives result in higher losses than false positives
  - thus a higher rate of false positives is normally tolerated than the rate of false negatives
  - if an error rate is very high, warnings tend to get ignored
  - proper tuning of the system is important
- The earlier intrusion is detected, the better
  - it is easier to recover while the damage is small

# Intrusion Detection Systems

- What we often want from an IDS
  - continuous operation
  - minimum human intervention
  - small overhead, ability to scale
  - ability to adapt to changes in user behavior and system characteristics over time
  - resistance to compromise (ability to monitor itself)
  - ability to be reconfigured on the fly, without restarting
- Often all of the above are extremely difficult to achieve simultaneously
  - e.g., ability to adapt in anomaly-based detection often has a higher

# Host-Based Intrusion Detection

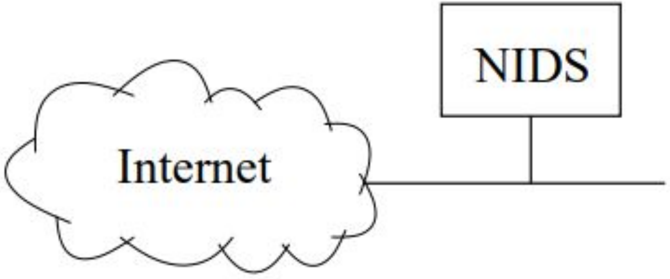
- A host-based IDS runs on a single host
  - it is best positioned to evaluate the state of the machine
- It can monitor events and activity such as
  - login and session activity
    - frequency and location, time since last login, failed login attempts
    - events of security importance can include break-in into a dead account, logins from unusual locations or unusual hours, password guessing, etc.
  - program execution activity
    - monitored activity can include execution denials, resource utilization and execution frequency

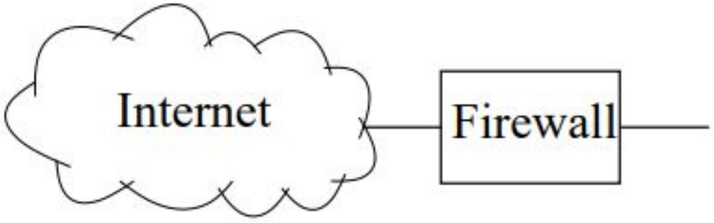
# Host-Based Intrusion Detection

- Monitored events and activity
  - file access activity
    - record frequency of different types of file access, denial of access
    - look for abnormal usage patterns, suspicious activity such as copying system programs or opening devices directly
  - some combination of the above
    - e.g., users who login after hours often access the same files they used earlier
- If a host-based IDS runs on each host, information from different machines can be collected and managed at a central facility
  - the central manager receives aggregate information and distributes updates to all machines running the IDS

# Network-Based Intrusion Detection

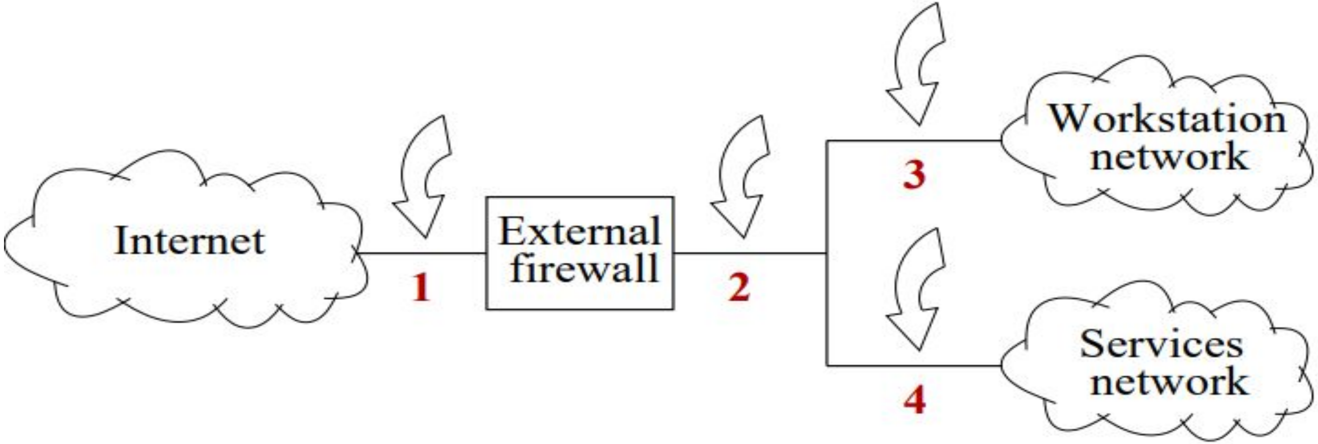
- A network-based IDS monitors traffic corresponding to many machines on a network
  - often such a monitor is passive
    - NIDS receives a copy of the traffic
  - a firewall, on the other hand, performs active filtering
    - all traffic goes directly through it
  - active filtering adds overhead and normally needs to be minimized





# Network-Based Intrusion Detection

- Where NIDS is positioned matters



  - point 1: complete picture of traffic, lots of data
  - point 2: can recognize problems with firewall, see outgoing attacks
  - points 3 and 4: increased visibility of attacks on the local network, can see internal attacks

# Network-Based Intrusion Detection

- A NIDS is often stateful and performs deep packet inspection
  - full stream reassembly
  - analysis at network, transport and/or application layers
    - network layer: IP, ICMP protocols, illegal header values, spoofed addresses
    - transport layer: analysis of TCP and UDP headers, detection of unusual packet fragmentation, floods, scans
    - application layer: understanding of DHCP, DNS, HTTP, Network File System (NSF), remote login and many other protocols; detection of buffer overflow attacks, malware propagation, etc.
  - detection of DoS attacks, scanning, malware (worms)

# Network-Based Intrusion Detection

- Example systems
  - Snort
    - can be host-based or network-based
    - can monitor traffic inline (supports intrusion prevention) or passively
    - intrusion detection/prevention is rule-based
  - Bro
    - provides passive monitoring of network traffic
    - suitable for high-speed high-volume detection
  - commercial appliances

# Network-Based Intrusion Detection

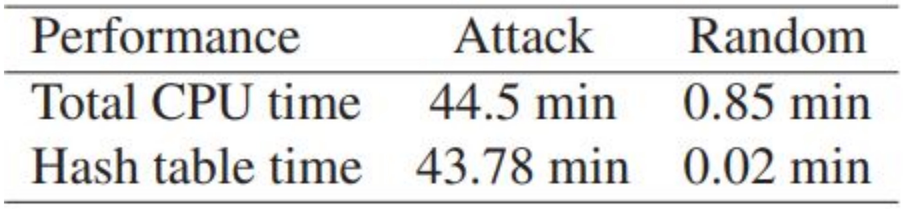
- Challenges in running NIDS
  - necessity to handle large volume of traffic
  - ability to correctly maintain the state of each machine on the network
  - ability to withstand attacks on NIDS itself
- Attacks on NIDS
  - algorithmic complexity attacks
  - evasion attacks
  - stealthy port scanning

# Attacks on NIDS

- Algorithmic complexity attacks
  - DoS attacks are already serious for denying service, but can be more severe as a component of an attack
  - DoS attack on IDS enables other attacks to remain undetected
- Example: complexity attack on hash table
  - on average, a hash table has $O(n)$ overhead to insert $n$ elements
  - in the worst case, it may have $O(n^2)$ overhead to insert $n$ elements
  - Perl implementation for 90 thousand inserts
    - random: < 2 sec
    - worst case: > 6500 sec

# Attacks on NIDS

- Complexity attack against Bro
  - Bro used simple XOR to “hash” values for hash table
    - easy to find collisions
  - for example, Bro port scanning detector keeps a hash table of destination IP addresses
    - keep the list of destination IP addresses for each (source IP, destination port)
  - using source IP spoofing one can exploit this structure to perform DoS attack



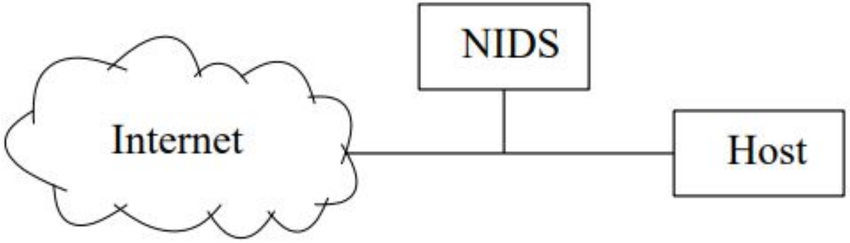
| Performance | Attack | Random |
|---|---|---|
| Total CPU time | 44.5 min | 0.85 min |
| Hash table time | 43.78 min | 0.02 min |

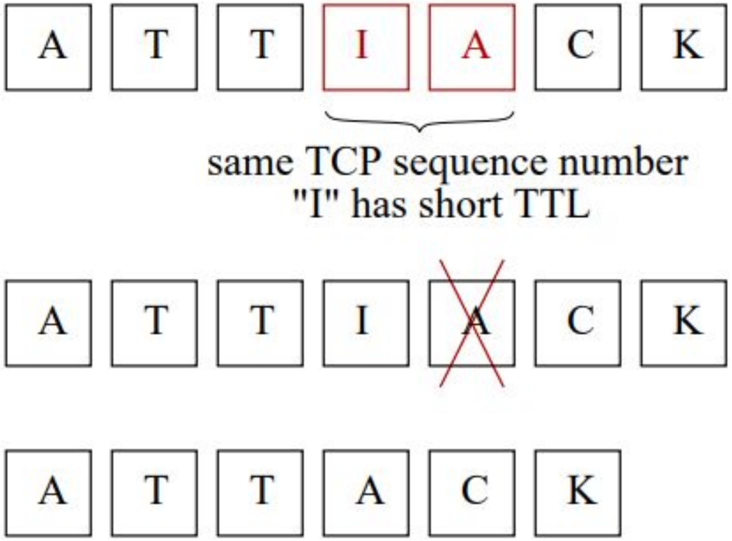
# Attacks on NIDS

- NIDS evasion
  - attack might rely on the fact that NIDS is not the target host and might have incomplete picture
  - complete fragment reassembly is necessary to detect certain attacks
  - NIDS only has partial knowledge of what the host sees
    - Time-To-Live (TTL) expires before reaching the host
    - packets that exceed the maximum transmission unit (MTU) are dropped
  - ambiguities in TCP/IP (e.g., overlapping IP and TCP fragments)
    - different OSs implement the standard differently

# Attacks on NIDS

- Small TTL attack





Attacker’s data stream: A T T I A C K

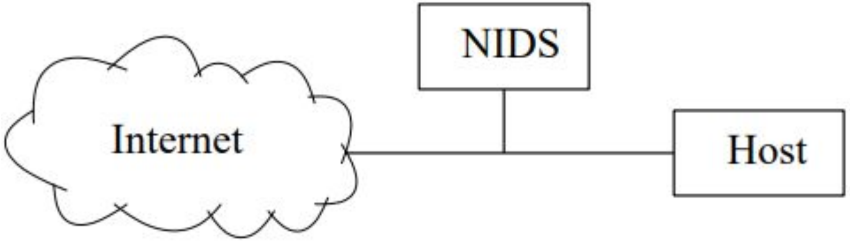
same TCP sequence number
"I" has short TTL

NIDS sees: A T T I ~~A~~ C K

End host sees: A T T A C K

# Attacks on NIDS

- Fragment overlap attack



Attacker’s data stream: A T T I A C K

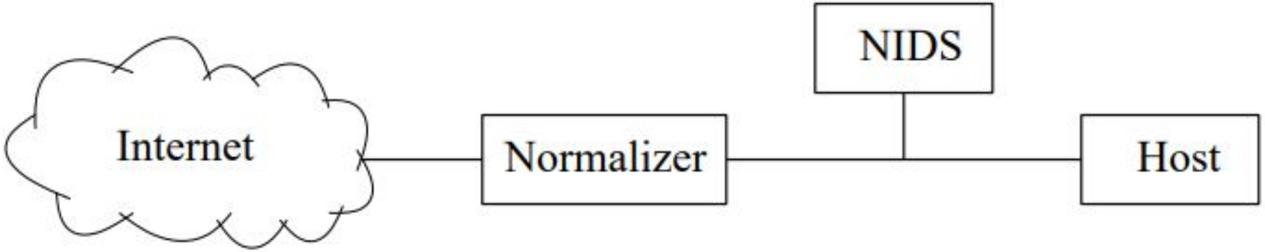
same TCP sequence number
or same IP fragment offset

NIDS sees: A T T I ~~A~~ C K

End host sees: A T T ~~I~~ A C K

# Attacks on NIDS

- How do we defend against such attacks?
  - solution: introduce traffic normalizer to avoid ambiguities



  - drop overlapping IP/TCP fragments
  - increase TTL in packets with low TTL
- But IDS evasion can still be possible
  - different interpretation of strings of characters at higher levels
  - e.g., `A T T I DEL A C K`

# Intrusion Detection

- For more reliable detection, NIDSs can be placed at different points inside the network
  - one monitor for the entire network
  - a monitor inside each subnet
  - this results in a distributed IDS
- Hybrid IDSs can be most effective
  - run IDS both on hosts and network
  - combine the data for improved decision making

# Conclusions

- Intrusion detection systems
  - signature-based: effective, but don’t recognize new attacks
  - anomaly-based: can find novel attacks, but often result in many false positives
  - host-based: best positioned to detect attacks on a machine
  - network-based: monitors traffic of the entire network
- Effort must be applied to protect the IDS itself from attacks