Advanced Networking: networking Vs. privacy and security

Leonardo Maccari

leonardo.maccari@unitn.it

18 dicembre 2012

Contents

Privacy

- What is privacy?

We can try with:

- The **protection** of **private data**

What are the private data?

For instance..

- Home address?
- Sexual preferences?

What does *protection* mean?

- Access?
- Control?

Second guess:

- Privacy is what law says privacy is

What if. . .

- You are on the *Lost* island
- You live in a non-democratic country

Does that mean you have no privacy?

Third guess:

- It's the distinction between private and public stuff.
- . . . and what's public?
  - Is what you do in your house private?
  - Is what you do in a public square public?
  - And is Facebook a public or a private space?

Let's change the approach:

What is a *privacy invasion?*

- somebody reading your email without your consent?
- somebody taking picture of you in the shower?
- ...

Summing up

- There is no clear definition of privacy
- At the same time there is a clear vision of when you are victim of a privacy invasion
- We can't define it, but we know it has a value when it is unrespected

Privacy is a new concept

- We talk about privacy since a century (when journalists started to take pictures)
- We are really interested in it since few decades (when computers started elaborating our data)
- Privacy conflicts with other well established collective values (freedom of information, security, free market . . .). So is privacy just a selfish value?

Presidio Modelo:

The Panopticon [1]:

Essentially, it was for a building on a semi-circular pattern with an 'inspection lodge' at the centre and cells around the perimeter. Prisoners, . . . were open to the gaze of the guards, or 'inspectors', but . . . by a carefully contrived system of lighting and the use of wooden blinds, officials would be invisible to the inmates. Control was to be maintained by the constant sense that prisoners were watched by unseen eyes. There was nowhere to hide, nowhere to be private. Not knowing whether or not they were watched, but obliged to assume that they were, obedience was the prisoner's only rational option. [...]. Bentham's innovation, then, was not just to inspect, or even to ensure that the gaze is asymmetrical, but to use uncertainty as a means of subordination.

[1] D. Lyon, The electronic eye: The rise of surveillance society. Univ Of Minnesota Press, 1994.

Panopticon today:

- The panopticon is not only a model of a prison, conceptually is a model of a society in which everybody give some of his personal privacy away receiving in exchange a globally safer society
- This view of society has been rediscovered lately to describe with an extremed metaphor the widespread of technological instruments of control
- Privacy is not only a selfish value, it is a fundamental value for society to develop freely

So, what's the point?

- The point here is that privacy does not exists as a well-defined concept
- It is always in conflict with other values
- It often looses the conflict
- Technology is making things worse for the majority of people

Try to put some context in

- Helen Nissenbaun says that Privacy is all a matter of Contextual-Integrity:

  Some critics have concluded that privacy is at best a culturally relative predilection rather than a universal human value. The framework of contextual integrity begins with the same observation, but draws a different conclusion; there is, indeed, great complexity and variability in the privacy constraints people expect to hold over the flow of information, but these expectations are systematically related to characteristics of the back-ground social situation. Once these characteristics are factored into an account of privacy expectations (hereafter referred to as norms of information flow), the law-like character of these privacy expectations, or norms, is much more evident.

How-to

You have to identify:

- the context. The place where the exchange happens, being it a geographical space, a building, a social place (school, hospital...)...
- the actors. More specifically, the feature of the actors and consequently the relationships that intervene between the ends of the exchange
- the attributes. The tags applied to the information that is exchanged, and consequently the level of privacy applied to it...
- the transmission principle. Some rule applied to the exchange (for instance, the obligation to not further give the information away).

This will help you identify the Social Norms that are present in this context, and understand if they have been broken by somebody.

How do we get out of here?

- We are techies, not philosophers. . .

Back to Networking and Security

Something similar happens with security.

- Communication networks exist since thousands years.
- Electronic communications exist since almost 200 years, and optical ones since end of the 18th century years
- Up to few years ago, communication networks were centralized and unavailable to the users.
- Today the same (or similar) technology that your provider uses can be used by any and-users (hardware and software)
- We have much more security problems than before

Back to Networking and Security (II)

Moreover. . .

- The communication networks we are using are not imagined to have embedded security systems into it
  - ARP, IP, DNS as used today have no security embedded
  - TCP, UDP neither
  - SMTP, HTTP, have been designed with poor security requirements
- Every time you enforce security measures you add a cost:
  - Strong passwords $\rightarrow$ user complexity
  - TLS certificates $\rightarrow$ more computation power required
  - VPN $\rightarrow$ transmission overhead

Back to Networking and Security (III)

We can ask for security the same questions that we can ask for Privacy:

- What is security?
- How do I enforce it?
- Is it in contrast with other values?
- How valuable is it?

Security assessment

Security people call this a **security assessment**

> *The baseline or reference point for any security assessment should be corporate security policies. . . . Security policy must be deployed so that it's known and accepted by employees, project managers, and management throughout the corporation. Information Security Policy World states that "the fundamental question is how to deploy the policies - how to deliver them. This is critical, as undelivered or badly delivered policies might as well not exist"* [2]

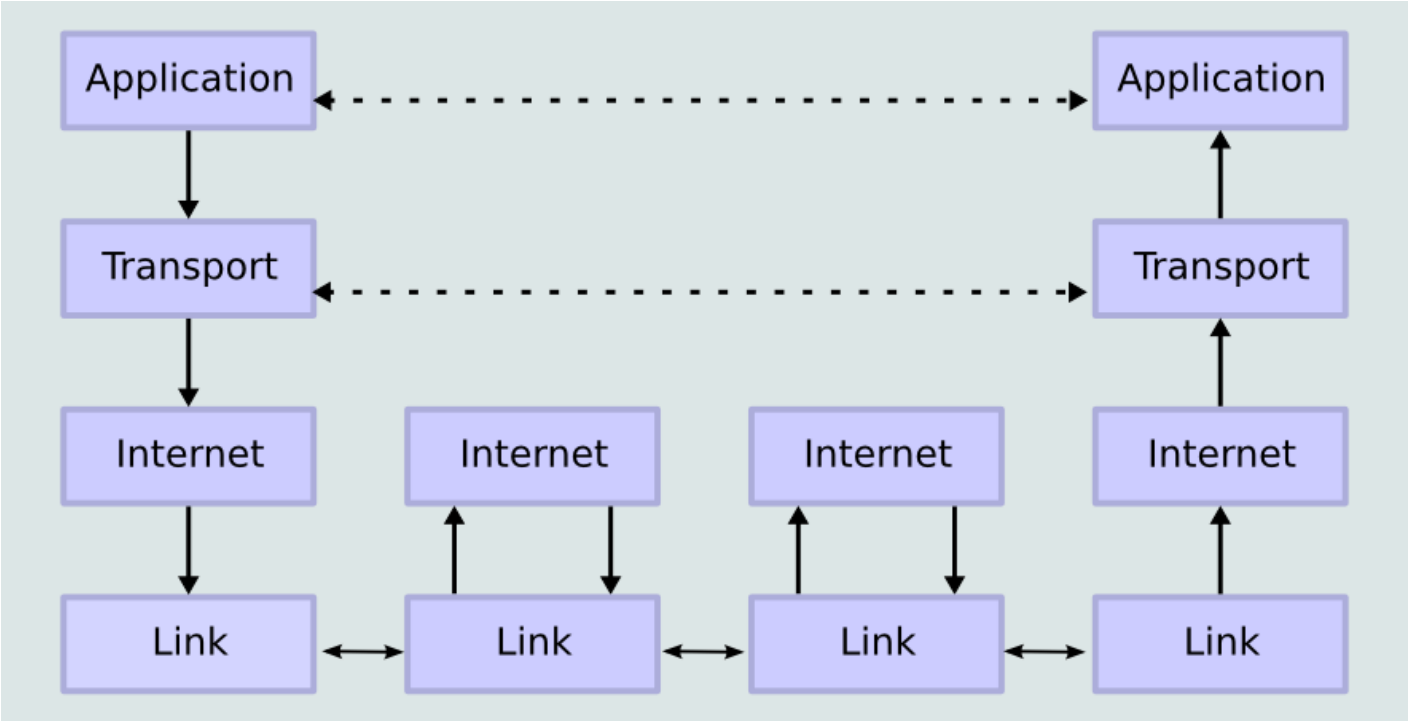
- Gather information
- Identify the requirements
- Identify the risks
- Test the security measures . . .
- Basically, this is a technical translation of what Contextual norms were for the more abstract Privacy concept

[2] Sans Institute: Implementing a Successful Security Assessment Process

What should you do?

- It's up to you to identify the correct services, measures and instruments to give your users the wanted level of privacy, using security instruments.
- There are standard methodologies and techniques to assess security. But, as with privacy it is way easier to find negative examples: are these security breaches?
  - All your communications with your mail server are intercepted by someone
  - Your computer starts to send spam
  - Your server is victim of a Denial of Service attack
- It is easier to define what is against security than what security is
- . . . it's also more fun!
- in the remaining time, we are going to focus on this, how to attack privacy and security layer by layer, and, when possible, how to limit the attacks.

Our playground



Jamming

- jamming on the physical medium can be done if the physical medium allows it, as in the case of wireless networks or wired networks with unshielded cables
- For instance, in 802.11 even if the jamming is difficult, just changing the reception of one bit will invalidate the checksum and cause the packet drop

Link layer

- DoS: flood of packets, generating collisions
- MAC address spoofing
- ARP-Spoofing

Flooding the Link layer

- flooding the available physical resources
- if the medium is shared, you can ignore the network timeouts and create collisions, or keep the channel always busy
- a flood may be masked by sending packets with spoofed source address

MAC address spoofing

- Is it possible to change the MAC address of a device?
- try this on Linux “ifconfig eth0 hw ether 00:11:11:11:11:11”
- try it on a wireless captive portal authentication

ARP-Spoofing

The ARP protocol

- the machine 192.168.2.51 needs to communicate with address 192.168.2.52 in the same subnet. To do so it must send a frame to the corresponding MAC address.
- if it is unknown it sends an ARP request in broadcast asking the machine 192.168.2.52 to notify its MAC address
- the machine 192.168.2.52 responds with an ARP Reply message indicating that its MAC address corresponds to the IP 192.168.2.52

ARP protocol

Frame ARP

ARP protocol fields:

Hardware type (ethernet)
Protocol type (IPv4)
[...]
Sender hardware address
Sender protocol address
Receiver hardware address
Receiver protocol address

Receiving ARP packets:

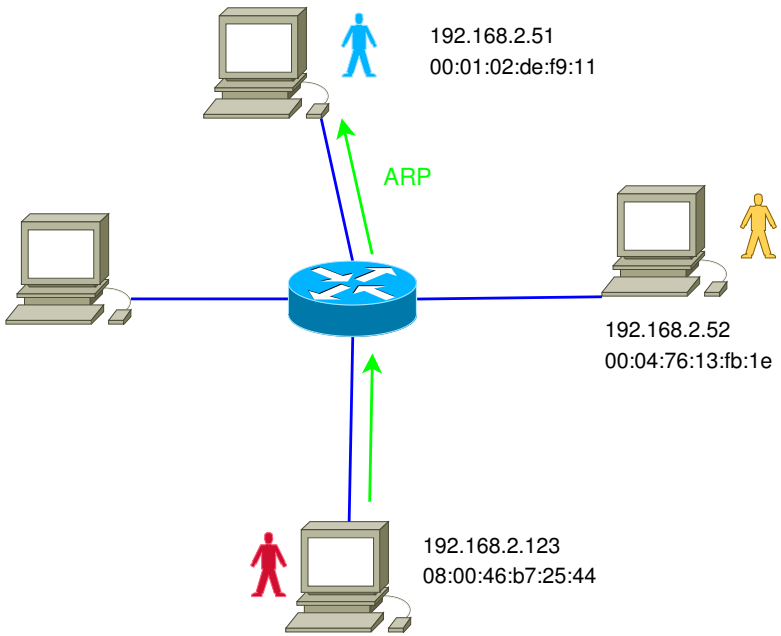
```
?Do I have that hardware type ?
Yes: (almost definitely)
  ?Do I speak that protocol ?
  Yes:
    If the pair <protocol type, sender protocol address> is
        already in my translation table, update the sender
        hardware address field of the entry with the new
        information in the packet and set Merge\_flag to true.
    ?Am I the target protocol address?
      Yes:
        [...]
```

The ARP table is updated even when you have no pending requests

ARP-Spoofing

ARP-spoofing attack

- Eve sends ARP-Reply messages to MAC address of Alice, stating that Bob's IP address corresponds to its MAC Address
- Eve also sends ARP-Reply messages to the MAC address of Bob, stating that Alice IP address corresponds to his MAC address



What happens if:

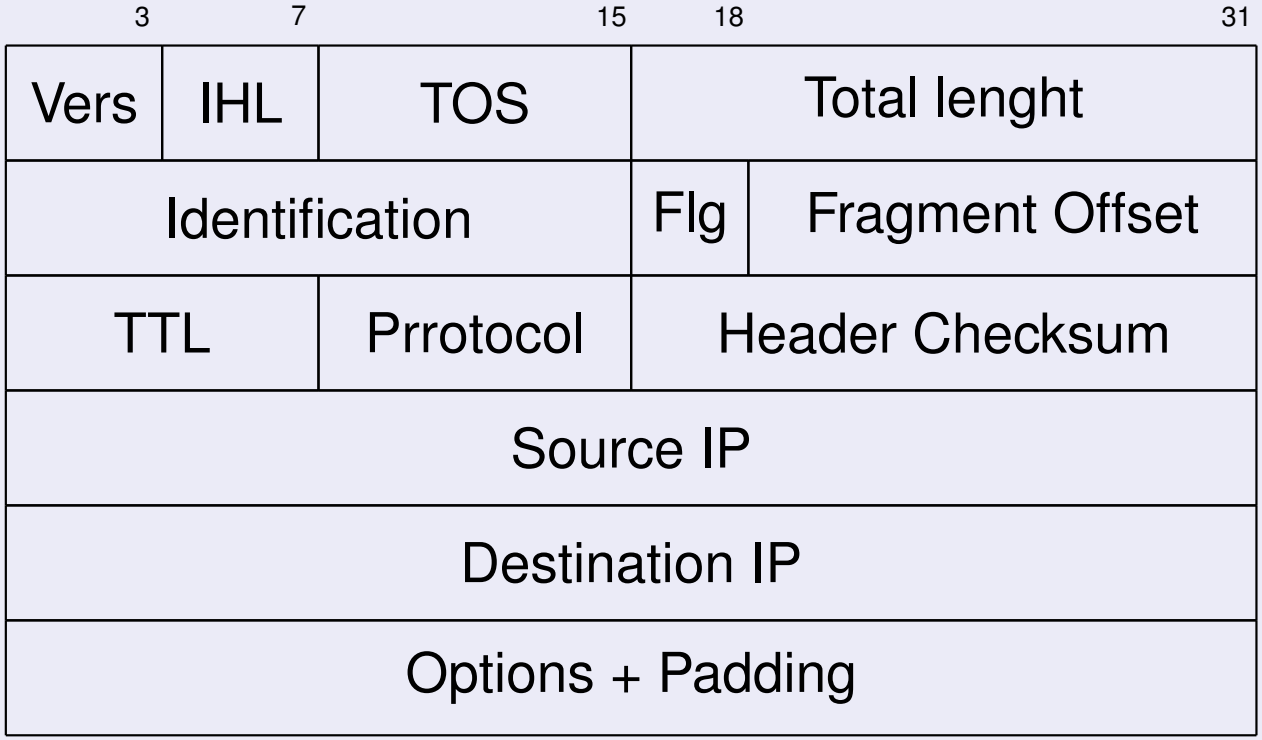
- Alice wants to communicate with another host in the network?
- He will do an ARP request, receive a response and will change Bob ARP table
- Eve may not be in the middle anymore
- Eve has to keep sending spoofed ARP requests to keep the attack going

Countermeasures

- Statically assign MAC addresses to Switch Port, do not let ARP packets pass if they do not match the static table $\rightarrow$ more expensive hardware and less dynamic network
- Try to detect ARP spoofing and report it (periodic ARP requests/reply, Multiple IPs with the same MAC address (or the inverse))
- Try to detect weird network topologies, i.e. all the packets are sent to the same host that is not a known router.

Fragmentation attacks

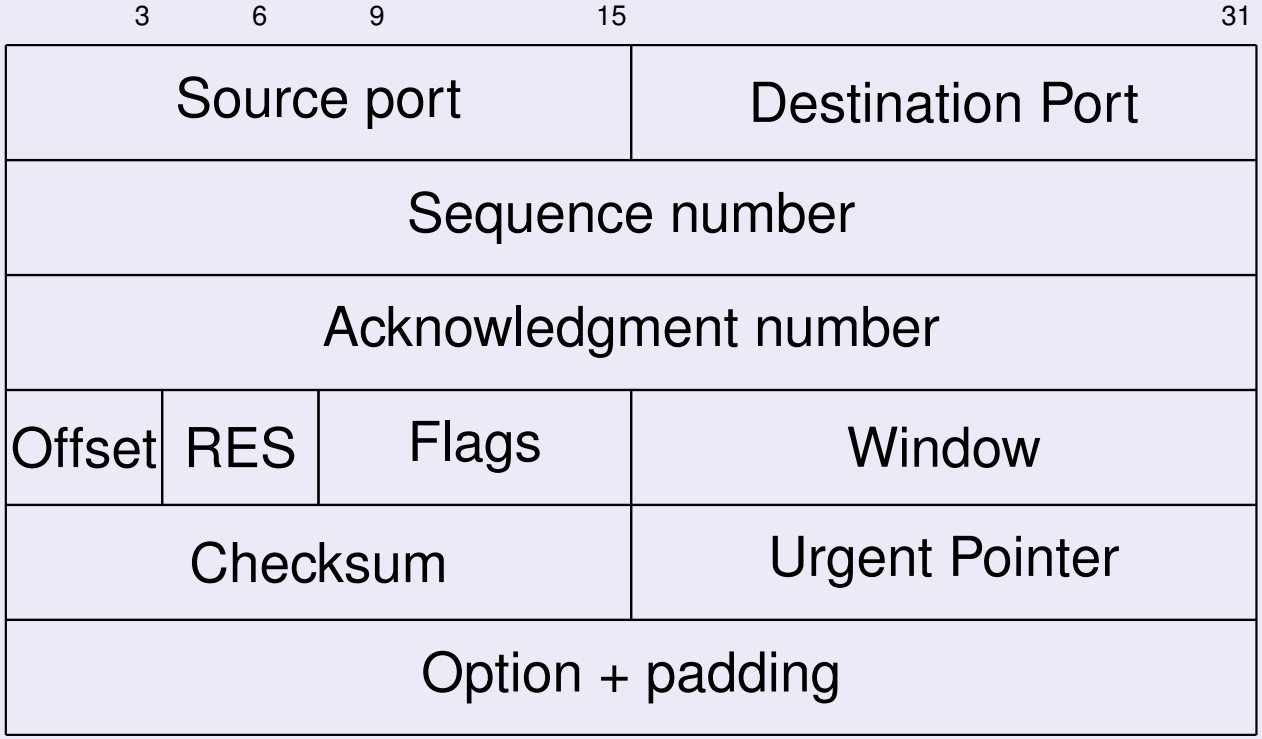
- The IP protocol allows you to break a packet in fragments to cross subnets with MTU less than the length of the packet itself.
- **IP Packet Header:**



Fragmentation attacks, segue

Overlapping fragments attack

- **TCP Packet Header:**



- The IP protocol allows to split packets into fragments smaller than a TCP header. In addition, packets that arrive later can rewrite those who came before

Fragmentation attacks:

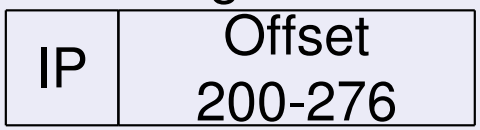
IP fragmentation:

- Original IP packet:

- First fragment:

- Second fragment:

- Third fragment:



Fragmentation attacks, why?

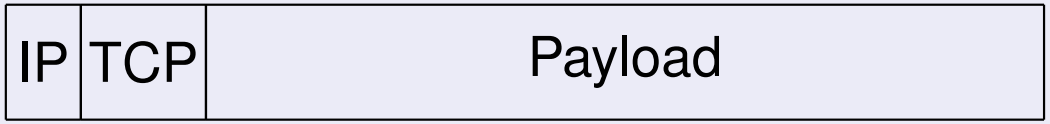
Tiny fragments attack:

- Firewalls are typically configured to block attempts to connect from outside the network to high ports (above 1024) of a server.
- But high ports are used to open connections from nodes behind a firewall, so firewall can not just filter all the packets towards high ports.
- To block attempts to **open** a connection **stateless firewalls** drop packets with the SYN TCP flag set to 1 TCP to high ports (connection initiation).
- Is it possible to circumvent such filtering?

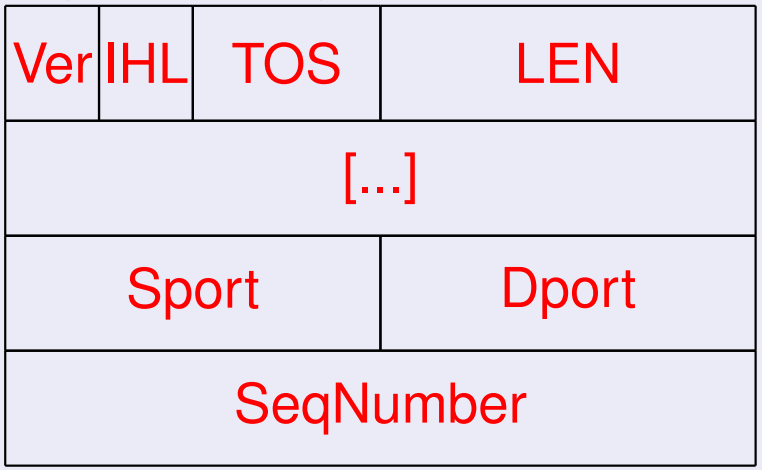
Fragmentation attacks, segue

Tiny fragments attack:

- Original IP packet:

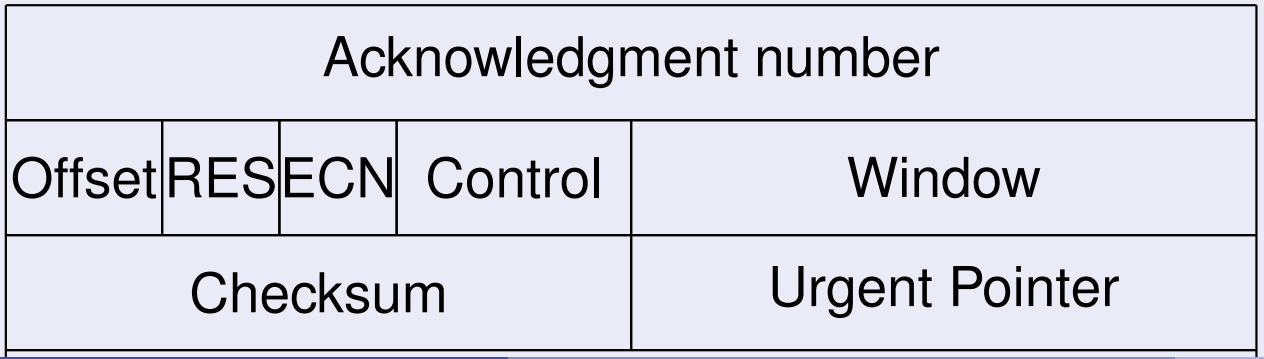


- First fragment (8 byte):



Header IP
First 8 bytes of
TCP header

- Second fragment:



Fragmentation attacks, segue:

Tiny fragments attack:

- Using a first fragment that does not contain the TCP flags the firewall allows the fragment even if directed to a port > 1024
- The second fragment is not interpreted as a TCP packet because it does not contain a valid TCP header so it is not filtered.
- When the packet arrives at the destination machine is reassembled, is directed to a port> 1024 and has the SYN flag = 1 but passed through the firewall

Fragmentation attacks, variant:

- The fragments are overlapping:
- Original IP packet

- First fragment

- Second fragment

Fragmentation attacks, segue:

Overlapping fragments attack:

- The first fragment contains a destination port> 1024 but the flag SYN = 0, then the firewall will not filter it
- The second fragment does not contain a valid TCP header therefore it is not filtered, but it rewrites a part of TCP header. Particularly corrects the flags: SYN = 1
- When they arrive at their destination the fragments are reassembled and the second overwrites the first, forming a TCP packet with valid destination port> 1024 and SYN = 1.

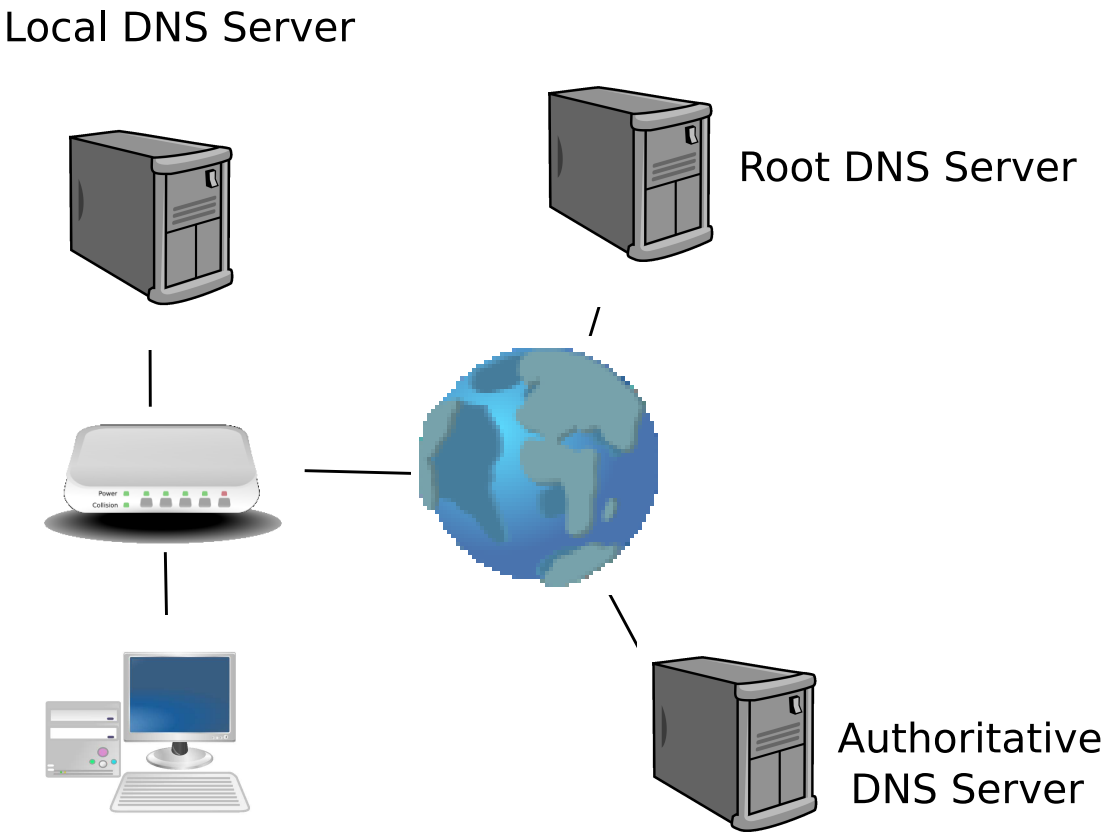
Fragmentation attacks, segue:

Fragmentation attacks have been superseded by the advent of **stateful firewalls** that rebuild the original packet before applying the filter

Recall DNS

- Each host must be able to resolve a network address from an alphanumeric string (a domain). This operation is carried out with the DNS protocol, Domain Name System.
- For each domain, there is a server which can play this task in an *authoritative* server.
- The address of an authoritative server for each domain is not known a priori, then each host is configured with an address of a local DNS server
- The local DNS server will request to the DNS root servers an authoritative server for a certain domain
- Once the name resolution domain / IP has been done, DNS maintains the association in a local cache, for a certain period.

recall DNS



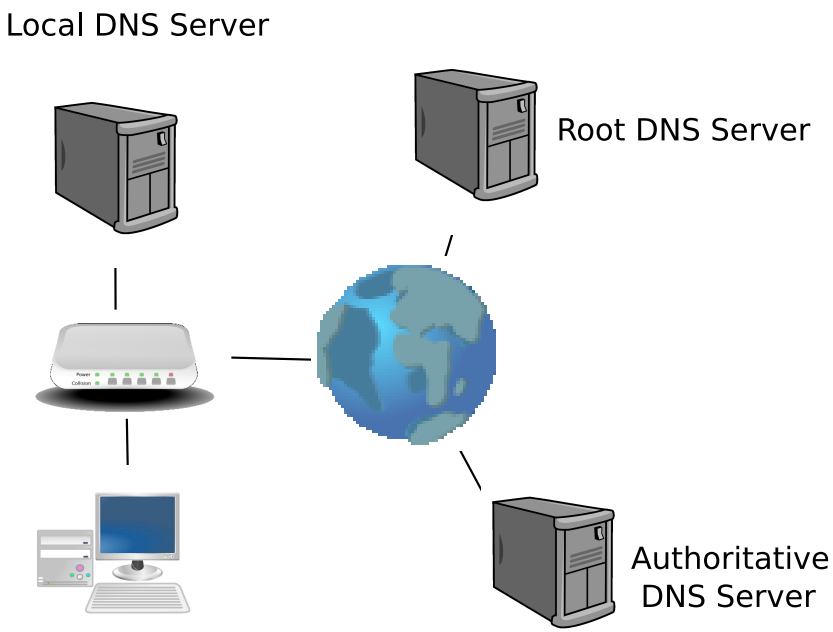
Attacks on DNS

- In general, an attack on a DNS aims to convince a certain victim host that the IP address that corresponds to a domain name is a different IP than the original one.
- The DNS protocol does not use forms of encryption to protect the packets, so you can falsify the answers of a DNS server.
- What for?
  - phishing,
  - theft of credentials
  - attacks on home-banking
  - redirection of connections and, in general, Man in the middle

DNS Attack

Where can the attacker be?

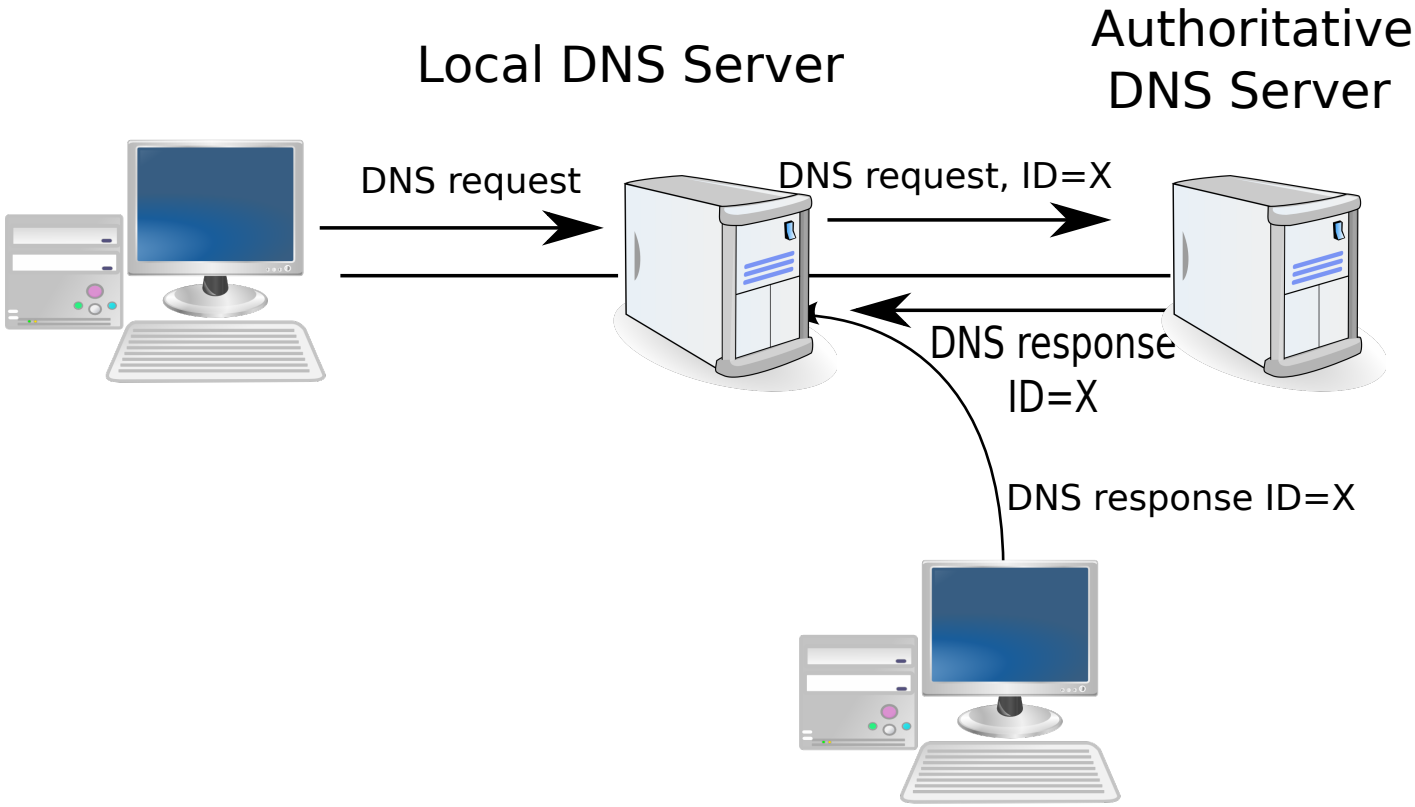
1. In the local network
2. Somewhere in the path from the client to the local DNS server
3. Somewhere in the path from the local DNS server to the authoritative one



DNS Attack - Layer II

- We have seen that using ARP spoofing we can easily perform a man in the middle attack, so if the attacker resides in the same network the attack is easy to perform
- With the same logic, the attacker can siimply spoof the DHCP replies and assign a fake DNS server to the victim
- But what if the attacker is not in the same switched network of the victim?

Attack scheme



DNS Attack

- The goal of the attacker is to be able to pollute the cache of the local DNS server.
- To do this, when the local DNS server sends a request for a remote domain, the attacker must respond with a forged packet containing the right data
  - The destination IP address $\rightarrow$ the remote server (predictable)
  - Destination UDP port number $\rightarrow$ (53)
  - Source IP address $\rightarrow$ the local server IP (predictable)
  - Source UDP port $\rightarrow$ randomly chosen by the local server (non predictable?)
  - random ID inserted in the request $\rightarrow$ (non predictable!)
- There are two fields, both of 16 bits that may not be predicted by the attacker

DNS Attack - numbers

- 32 bit $\rightarrow 2^{32}$ chances
- $2^{32} \simeq$ 4 billions packets with a brute force attack
- The attacker, provided that the he knows when to send the fake answer, it must send to the remote server an average of 2 billion packets.
- If each packet is 80 bytes long, the attacker must send 160GB traffic before the remote server answers $\rightarrow$ unfeasible.

DNS Attack - restrict the assumptions

- Some DNS do not use a random port to send requests
  - BIND didn't before 2009
  - If the DNS server itself is behind a NAT the port number will be changed by the address translation
- once the attacket knows the source port, still 2^{16} bit are unknown, which still accounts for $65000 * 80 \simeq 5MB$ of traffic.
- It is more feasible, but can we make anything better?

DNS Attack - restrict the assumptions

- Imagine that the attacker (Eve) can make requests to the DNS server (she is in the same routed network, or the DNS server (Alice) also accepts requests from the outside)
- At this point the attacker knows when the request will be sent to the root server:
  - Eve sends to Alice a request for the domain www.example.com
  - Alice resubmit the request to the DNS server for example.com (Bob)
  - Eve tries to answer before Bob, with a flood of responses,
- how likely is Eve to succeed? The *birthday paradox* can help us.

DNS Attack - Bithday paradox

Let's introduce the birthday paradox.

- What is the probability $P(n)$ that at least 2 of the n people in this room were born in the same day? We use inverse probability:
  - $\overline{P}(n)$ = *chances that none of n people were born the same day*
  - $\overline{P}(n) = \frac{364}{365} * \frac{363}{365} * \frac{362}{365} \ldots \frac{365-(n-1)}{365}$
  - Inverting: $P(n) = 1 - \overline{P}(n)$, we have $P(n) = 1 - \prod_{i=1}^{(n-1)} \frac{365-i}{365}$.
  - Surprisingly enough
    - per n=23 P(n) $\simeq$ 51%,
    - per n=30 P(n) $\simeq$ 70%,
    - per n=40 P(n) $\simeq$ 90%,

DNS Attack - Bithday paradox

How to apply the paradox to DNS poisoning attack?

- If Eve makes *n* requests for the host www.example.com, Alice will generate *n* IDs randomly chosen between 0 and 2^{16} and send them to Bob.
- Eve simultaneously send a burst of fake responses, with randomly chosen IDs
- If the ID of at least one of these false responses corresponds with the ID of at least one of the requests sent (and is received before the correct one from Bob) then Alice will have a fake entry in the cache for www.example.com
- Statistically for the birthday paradox, sending 700 requests/responses you have a probability close to 100 of guessing at least one answer.
- This way, the cache remains polluted for a time span and all subsequent requests are redirected to the host controlled by Eve.
- Note that even with a randomized port you have about 25% success probability with 50000 fake packets, which doesn't make it infeasible.

DNS poisoning, conclusions

- Poisoning a DNS server is theoretically possible but very difficult,
- Under suitable assumptions, however, the attack is perfectly feasible with the means available to anyone
- The attack can be helped if the original DNS server (Bob) is under a DoS attack, and does not respond promptly
- To prevent the attack it's important to choose the applications that use random source ports.
- Use DNSSEC (the authenticated version of DNS), root servers now support this protocol
- Do not accept unrelated answers to the DNS request

ICMP attacks (See RFC 5927)

- The Internet Control Message Protocol (ICMP) is used in the Internet architecture mainly to perform the fault-isolation function, that is, the group of actions that hosts and routers take to determine that there is some network failure
- When an intermediate router detects a network problem while trying to forward an IP packet, it will usually send an ICMP error message to the source system, to inform the source system of the network problem taking place.
- In the same way, there are a number of scenarios in which an end-system may generate an ICMP error message if it finds a problem while processing a datagram.

ICMP Attacks

- The received ICMP errors are handed to the corresponding transport-protocol instance, which will usually perform a fault recovery function.
- It is important to note that ICMP error messages are transmitted unreliably and may be discarded due to data corruption, network congestion, or rate-limiting an that there are no timeliness requirements for ICMP error messages. ICMP error messages could be delayed for various reasons, and at least in theory could be received with an arbitrarily long delay.

ICMP Attacks

- RFC1122 classifies ICMPv4 error messages into those that indicate soft errors, and those that indicate hard errors, thus roughly defining the semantics of them.
- RFC0792 also defines the ICMPv4 Source Quench message (type 4, code 0), which is meant to provide a mechanism for flow control and congestion control.
- RFC1191 defines a mechanism called Path MTU Discovery (PMTUD), which makes use of ICMPv4 error messages of type 3 (Destination Unreachable), code 4 (fragmentation needed and DF bit set) to allow systems to determine the MTU of an arbitrary internet path.

ICMP Attacks

- RFC0792 states that the IP header plus the first 64 bits of the packet that triggered the ICMPv4 message are to be included in the payload of the ICMPv4 error message
- For TCP, this means that the only fields that will be included in the ICMPv4 payload are the source port number, the destination port number, and the 32-bit TCP sequence number
- This means that there is no way of authenticating an ICMP packet:
  - You should have a security association with any host that generates it
  - The Original IP packet that generated the error can not be validated, since it is truncated in the ICMP payload (which invalidates any authentication algorithm)

ICMP Attack

- The current specifications do not impose any validity checks on the TCP segment that is contained in the ICMP payload.
- If you can guess the ports and IPs, you can forge a valid packet
- Some stacks are known to extrapolate ICMP hard errors across TCP connections, increasing the impact of this attack, as a single ICMP packet could bring down all the TCP connections between the corresponding peers

ICMP Attack - Hard errors

When these packets are received, the connections are reset.

- ICMPv4 type 3 (Destination Unreachable), code 2 (protocol unreachable)
- ICMPv4 type 3 (Destination Unreachable), code 3 (port unreachable)
- ICMPv4 type 3 (Destination Unreachable), code 4 (fragmentation needed and DF bit set)
- ICMPv6 type 1 (Destination Unreachable), code 1 (communication with destination administratively prohibited)

See RFC 5927 for specific countermeasures

ICMP Attack - Soft errors

When these packets are received, the connections are slowed down.

- ICMPv4 Source Quench message (type 4, code 0) (will reduce the TCP window)
  - Ignored by many implementations
- ICMPv4 Destination Unreachable, fragmentation needed and DF set (type 3, code 4)
  - use other protocols for MTU discovery (PLPMTUD) and safely discard such ICMP messages

Resetting TCP sessions

TCP Reset Guess

- A TCP connection can be terminated by one of the two participants by sending a packet with the RST flag = 1.
- To be accepted the packet must contain the correct values of:
  - IP address of the sender and destination
  - TCP port for sender and destination
  - Correct sequence number within the flow
- An attacker who wants to break a connection between two hosts without being along their path must know the IPs, can make a guess on the ports (one is known and the other may be predictable), but can not know the correct sequence number

RST attack

- The sequence number is a 32-bit field, again, a space of 2^{32} = 4,294,967,295 attempts.
- The TCP protocol, however, requires that to be received correctly, a packet reset should simply fall into the window of sequence numbers sequence that the machine maintains active.
- A TCP window can be up to 2^{16} bits.
- There is no need to try all the sequence numbers, but sampling with a distance of 2^{32} the attacker can be reasonably sure of being able to break the connection.
- $(2^{32}/2^{16}) = 2^{16} = 65,535$

Expected attack time

Reset times

Assuming a 2^{16} window:

Speed	# Packets	Time (1 port)	Time (50 ports)
256kbps	65,537 (*50)	81 seconds (1 min.)	4,050 (1.1 hours)
1.54Mbps	65,537 (*50)	13.6 seconds	680 (11 minutes)
45mbps	65,537 (*50)	1/2 second	25 seconds
155mbps	65,537 (*50)	1/10 second	5 seconds

Randomizing network parameters

Random ports

When developing your applications, keep in mind that all these attacks are greatly helped when source ports are not properly randomized

- When the application chooses (or negotiates) a non-random port
- When the network stack chooses a non-random port (unlikely today)
- When the network stack doing the NAT chooses a non-random port

Syn flood

Filling the host memory

- Whenever a server receives a packet with SYN = 1 allocates memory resources to handle the connection that is to be created, then sends a packet with the SYN flag = 1, ACK = 1 and starts a timeout to await the arrival of the third packet dell'handhake. The server has an *half-open* connection.
- If an attacker sends a large number of packets with SYN = 1 using a spoofed IP address, sooner or later server memory will saturate and the server begins to discard new connections.
- In this way it prevents other machines access to the service, performing a DoS attack.

Syn flood

Hard to tackle...

- This is not an attack on bandwidth resources. The attacker needs lower resources than the attacked hosts.
- The attack is perfectly asymmetrical, the attacker can use fake IPs

Syn Cookies

A potential solution

- When the server sends SYN = 1 and ACK = 1 it does not choose a random sequence number but it chooses a number that is the encoding of information regarding the connection and it does not allocate memory.

  5 bits : t mod 32, where t is a 32-bit time counter that increases every 64 seconds;

  3 bits : an encoding of an MSS selected by the server in response to the client's MSS;

  24 bits : a server-selected secret function of the client IP address and port number, the server IP address and port number, and t.

- When the server receives the third packet of the connection (ACK), it contains the incremented sequence number. The server will re-extract the encoded information. At that point the connection is open.
- Note that this reduces the sequence number to practically 24 bits

Port scanning

How to find open ports on a host

The simple way:

- perform a tcp handshake with every single port on target host

Nmap[a] port states:

- Open → port is reachable, service available
- Closed → port is reachable, no service open
- Filtered → packet has been dropped by a firewall on the way

[a] nmap.org

Scan types

Direct scans

- TCP connect scan $\rightarrow$ open a TCP connection
- Syn scan $\rightarrow$ do not send ACK (requires root privileges)

An open port will answer ACK, a closed port will answer RST, filtered port will not answer of answer with ICMP port unreachable

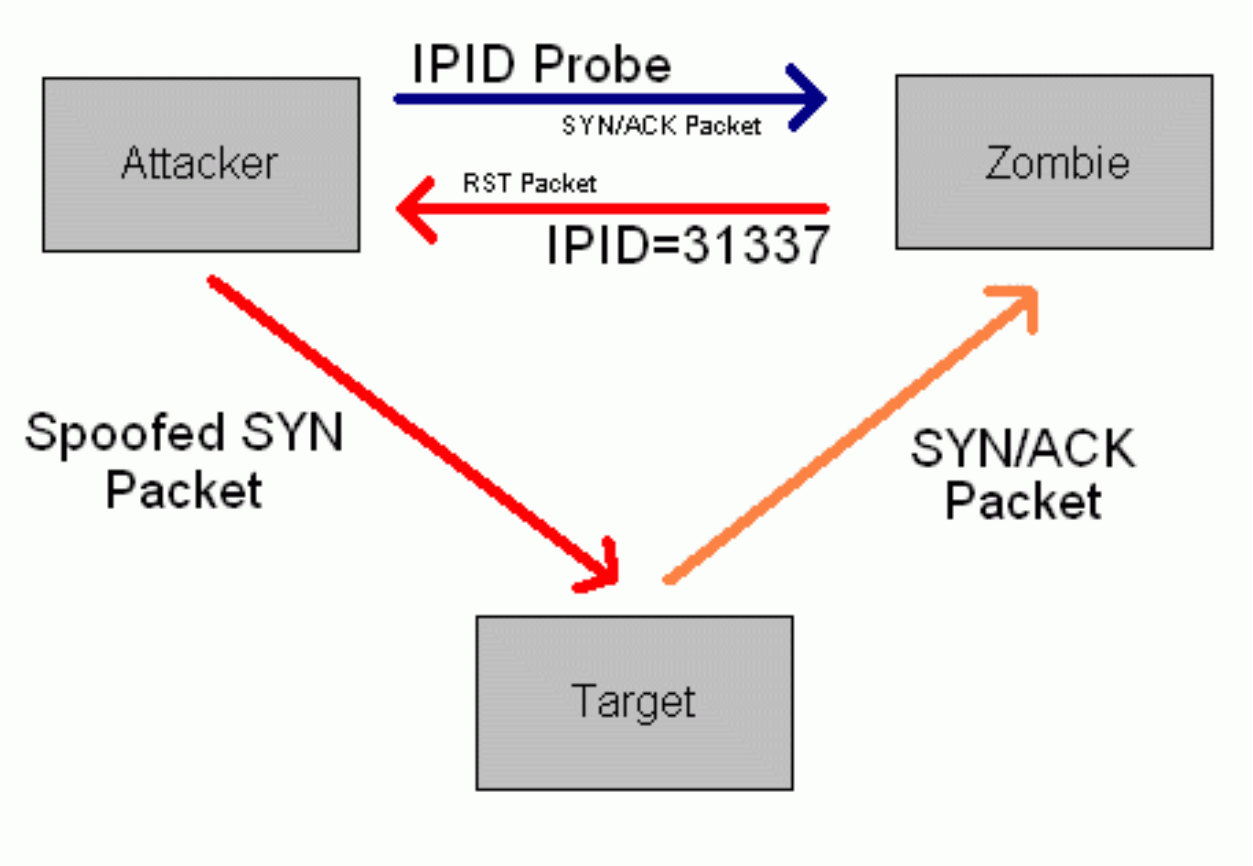
Scan types

Indirect scans

- TCP ACK scan $\rightarrow$ send ACK packets to detect if the firewall is stateful (ACK packets are not filtered by statless firewall and receive RST from destination)
- FTP Scan $\rightarrow$ use malconfigured FTP server that allows proxying requests
- Idle Scan $\rightarrow$ use an old zombie host to perform bouncing scans (one more sequence number related problem)

Idle scan[3]

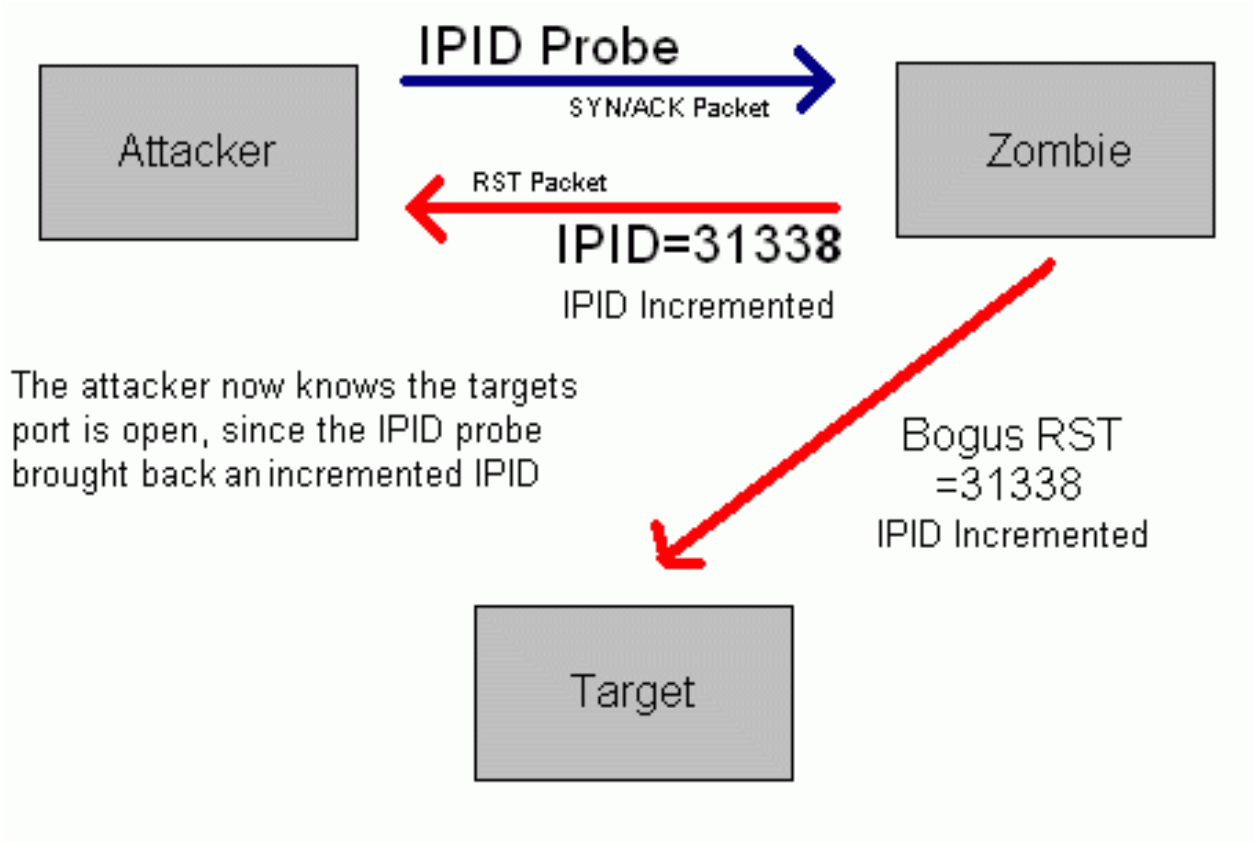
A perfectly blind scan, if you find an old zombie host



[3] http://en.wikipedia.org/wiki/Idle_scan

Idle scan

A perfectly blind scan, if you find an old zombie host



OS fingerprinting

- When the scan is done, the attacker knows which are the services available on the host
- Depending on the open ports on a host, he can try to guess its OS

Firewall

A firewall is a software or hardware device configured to allow or deny connections between two areas of a network with different level of confidence.

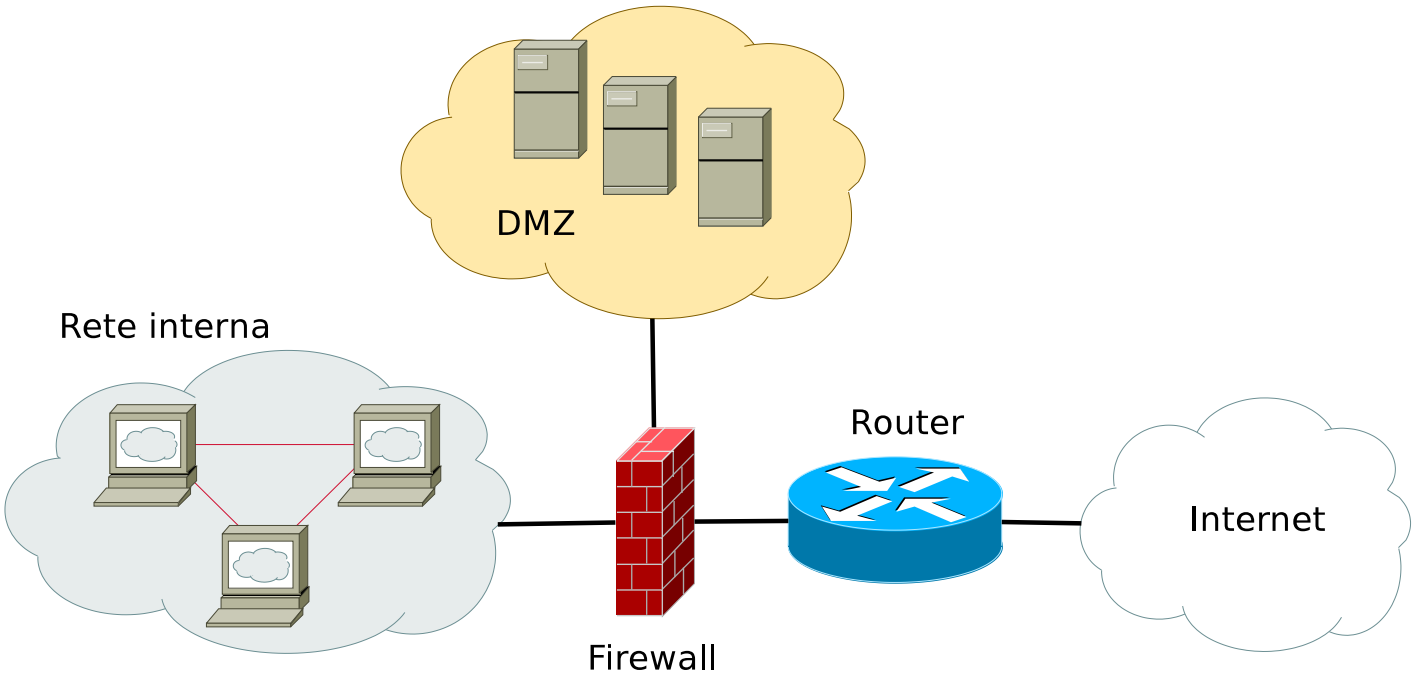
- Example: a perimeter firewall is usually placed on a gateway to separate the local network (high level of trust) from the Intenet (minimum level of trust)
- The ultimate goal of the firewall is to provide a configurable interface between two network segments with different levels of trust. The interface must be configurable through security policy based on two principles:
  - Least Privilege
  - Separation of duties
- The configuration of firewalls demands deep knowledge of network protocols and network security, an error in the configuration can make it useless.

Evolution of the firewall

- Packet filter: each packet passes through the firewall and for each of these it takes a separate decision. The decision taken at time t is not affected by the choices made at time $t - \delta$.
- Stateful firewall: the firewall implements state machines to take more complex decisions. For example, do not accept a TCP ACK packet if you have not received a SYN packet
- Application layer firewall: the firewall normally operates at the network level, where the format of packets is defined and can not change. The application firewall read information of the packet payload to decide which applications can pass. It requires more computational resources.

Placement of the firewall:

- The firewall is used to separate different areas of the network. A typical configuration is one in which the firewall separates two network segments:
  - an internal network (corporate) that hosts the user workstations, the data server and databases etc. It is the segment that contains the most important information for the corporation, and must be higly protected.
  - a network accessible from the outside, web servers, e-mail and DNS, which are in direct contact with the Internet and so exposed to higher risk. This area contains data that is accessible from the outside, then in principle lower value and less restrictions on access. We define DeMilitarized Zone (DMZ).

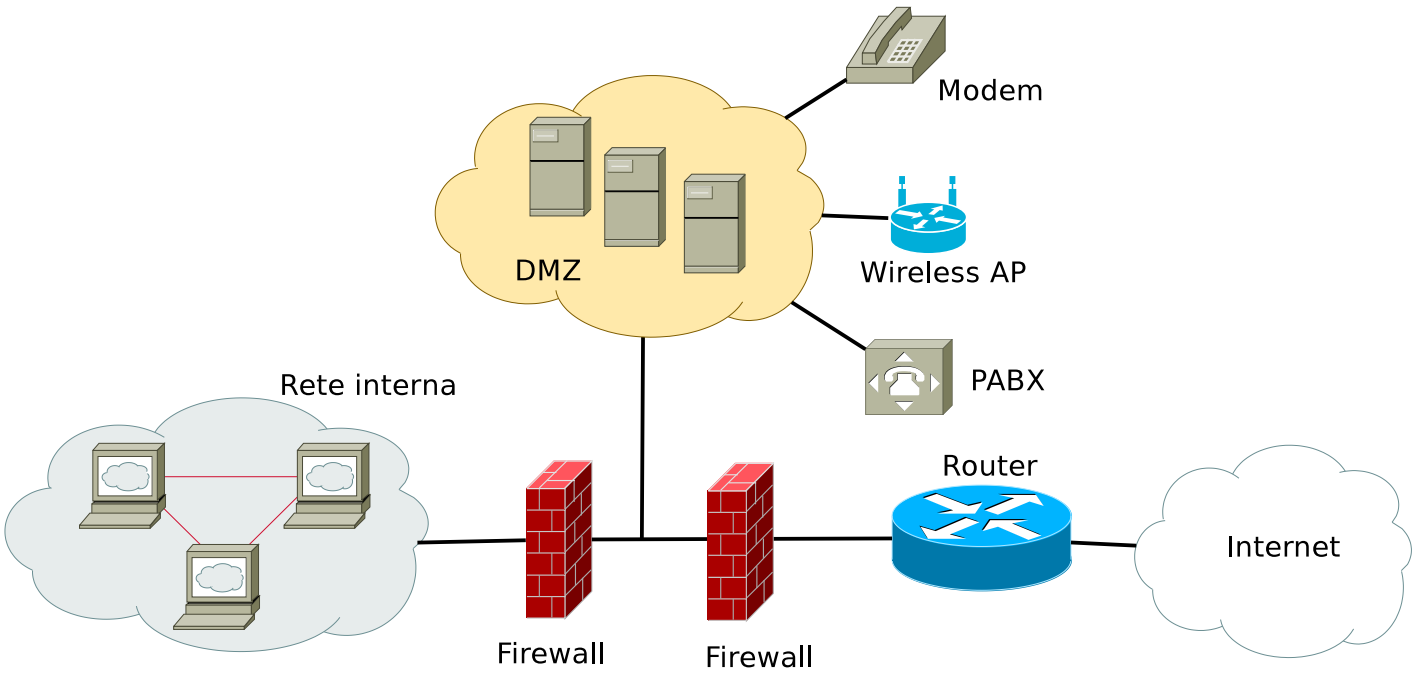

DMZ
Rete interna
Router
Internet
Firewall

Placement of the firewall:

A second configuration consists of adding an additional firewall so as to have two elements of defense before arriving to the corporat network. The configuration is more robust because:

- an attacker would pierce two firewalls before reaching the corporate network (firewalls must offer software or hardware redundancy)
- The DMZ is separated also from the inside outwards, with the same principle.
- It is easier to separate the traffic, so other connections to the outside that can be considered less secure can enter in the DMZ.

Note, this configuration doubles the costs and the effort needed


Modem
DMZ
Wireless AP
PABX
Rete interna
Router
Internet
Firewall
Firewall

Netfilter / Iptables

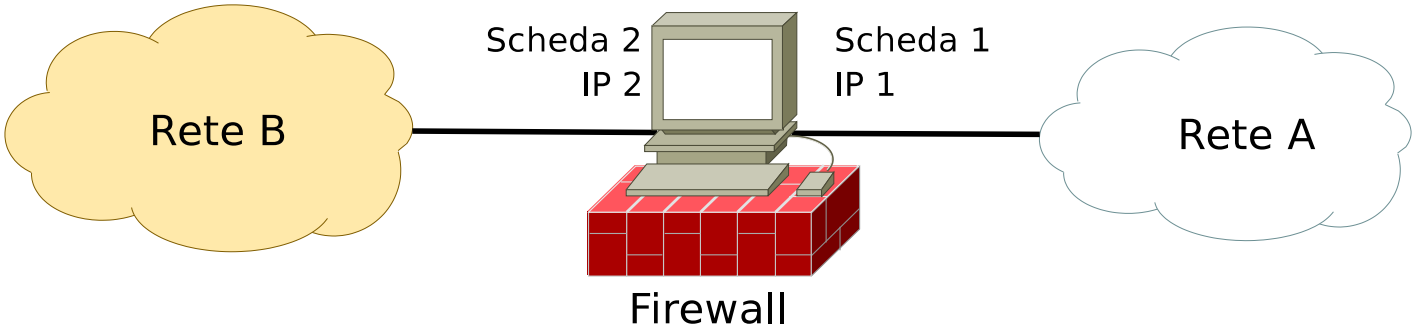
- Netfilter is the framework of the Linux kernel that allows you to perform packet filtering on a software firewall.
- Netfilter works in kernel space that is the core of the operating system and supports *hook*: attachment points in which the user can decide to filter the packets in their way inside the firewall.
- Iptables is a tool that allows you to insert, delete and organize the rules according to which the packets are filtered. An example of a rule:
  - iptables-t filter-A INPUT-dport 80-j ACCEPT
    - -T filter: table
    - -D input: chain
    - -Dport 80: policy match rule
    - -J ACCEPT: target

  translation: accept incoming packets on port 80

Chains and Tables

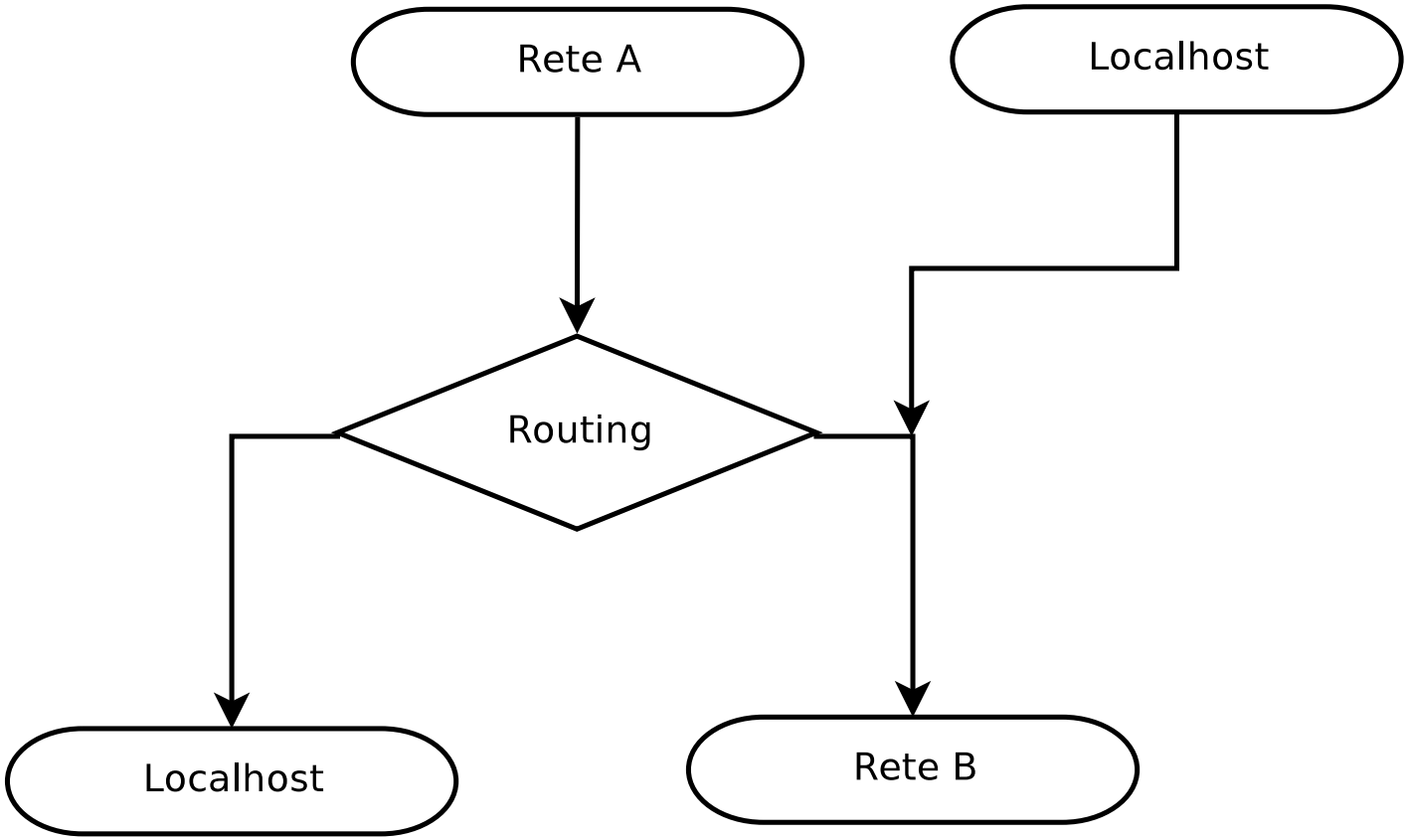
The rules are organized in chains and tables:

- A chain identifies the point in the path where the kernel is filtering.
- A table associates a function to the rule.
- What does that mean?



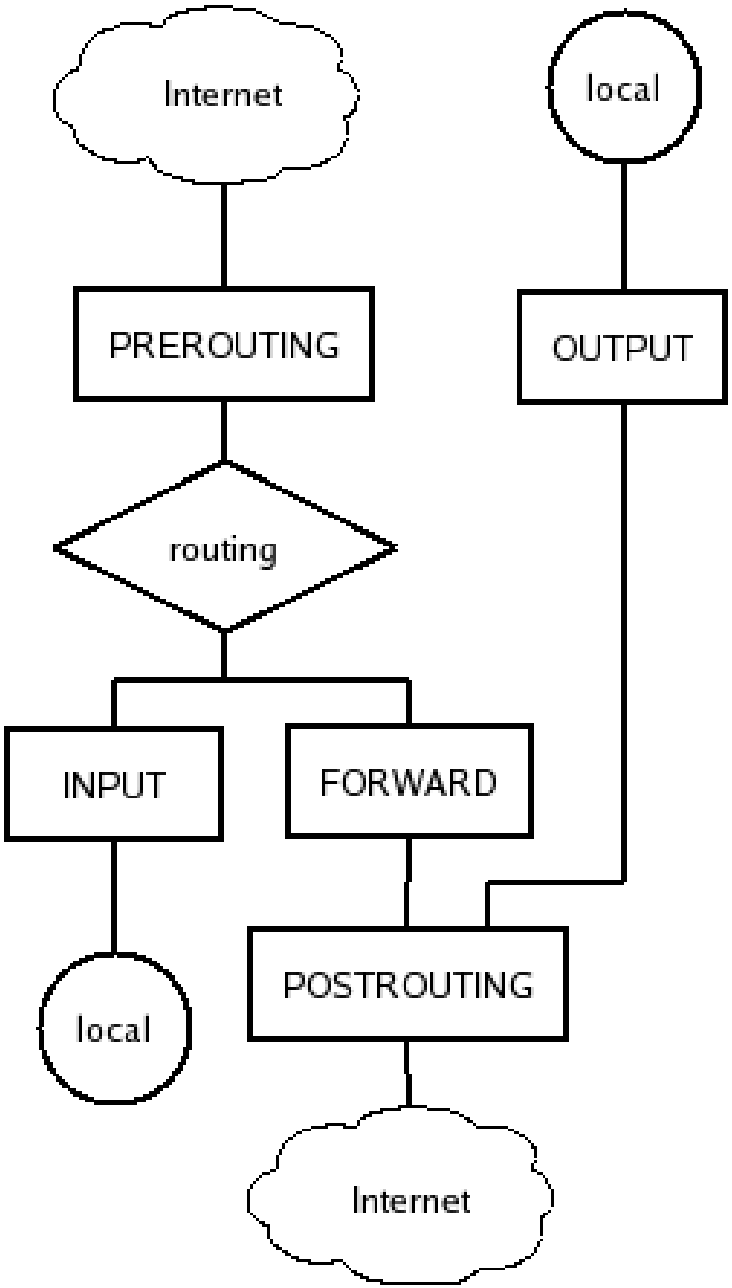
- A firewall is just a network host equipped with at least two network interfaces each of which has an IP address.
- Packets can get one of the interfaces, are filtered and forwarded, in this case the firewall behaves as a router
- If a packet received from the interface 1 is directed to IP 1, the packet is processed locally, and there there is no forwarding
- The firewall can generate packets that are sent to the network using one of the two interfaces, again, no forwarding

Firewall general structure



- Where can the packets be filtered?

Netfilter: chains

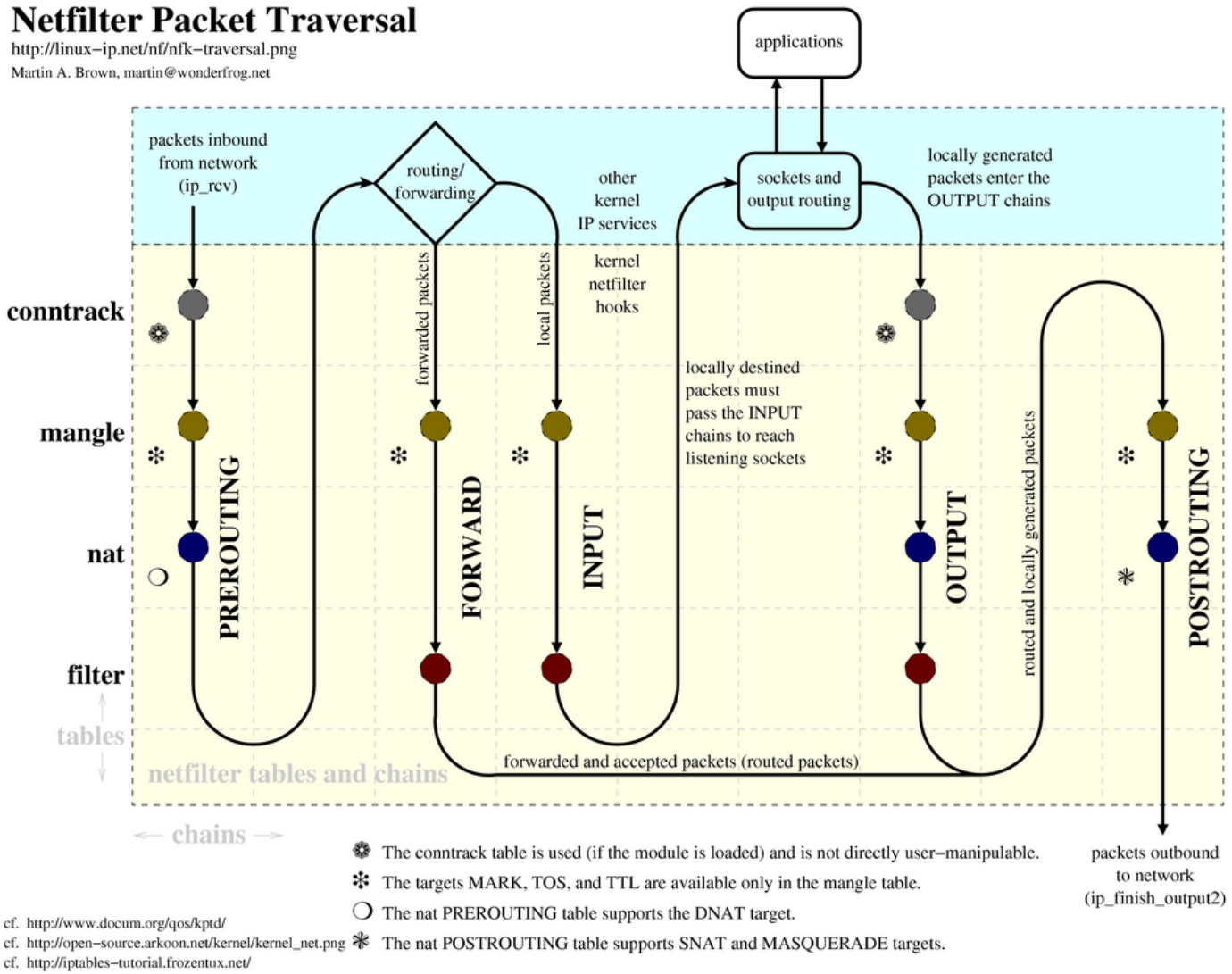


- Prerouting: all incoming packets
- Postrouting: all outgoing packets
- Output: outgoing packets generated by the firewall
- Input: incoming packets directed to the firewall
- Forward: packets passing through the firewall

What does *filter* mean?

- Discard (Drop)
- Allow (Accept)
- Modify (Mangle)
- Allow and report a message (Log)
- ...
- In addition Netfilter is stateful, so there must be a module that reconstructs the flow of related packets, for example, fragments of the same IP packet (conntrack)
- To distinguish groups of similar actions, the rules are divided into **tables**: groupings of rules that perform the same function.
  - Conntrack
  - Mangle
  - NAT
  - Filter

The complete architecture



NAT Table

NAT table: Network address translation, is used to change the address of the IP header of the packet. The targets are:

- DNAT: destination address translation, changes the target IP address. It is used, for instance, by a traffic balancer to distribute the load on a network with multiple servers.
  - iptables -t nat -A POSTROUTING -s 192.168.1.12 -j SNAT –to-source 150.217.5.123
- SNAT: source address translation. changes the source IP address. Is used to mask a private network address behind a public address.
  - iptables -t nat -A PREROUTING -d 150.217.5.123 -j DNAT –to-dest 192.168.1.12

The Filter table

Filter table: used to do the real packet filtering deciding which will pass and which will be blocked. The targets are:

- Drop: The packet is discarded without responding to the sender.
  - iptables -A FORWARD -p TCP –dport 22 -j DROP
- Reject: the packet is dropped with sending to the destination scaratato a reply
  - iptables -A FORWARD -p TCP –dport 22 -j REJECT –reject-with tcp-reset
- Log: packet generates a log (on-screen, file ...)
  - iptables -A INPUT -i eth0 -p tcp –dport 22 -j LOG

NOTE: the order of the rules is important!

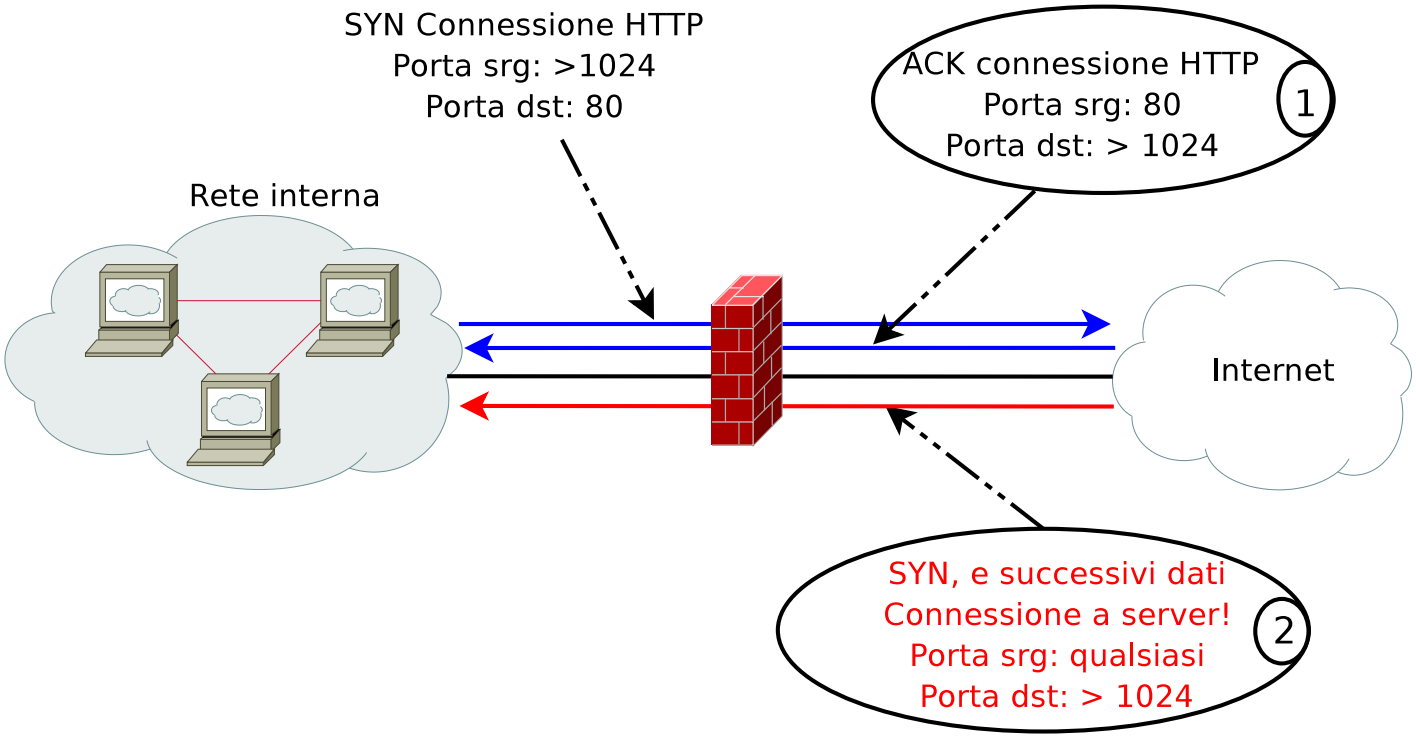
Connection tracking

The conntrack module performs key functions in the action of filtering, but should be used with caution or you risk saturating the resources of the machine. The aim is to relate different packets, according to the operation of a state machine, in order to identify:

- fragments that constitute the same IP packet
- packets that are part of the same connection
- packets that are part of distinct but related connections between them (for example FTP connections)

Il connection tracking

- Example: In a firewall that protects a private network, usually you do not want to allow connections from the outside to the high ports.

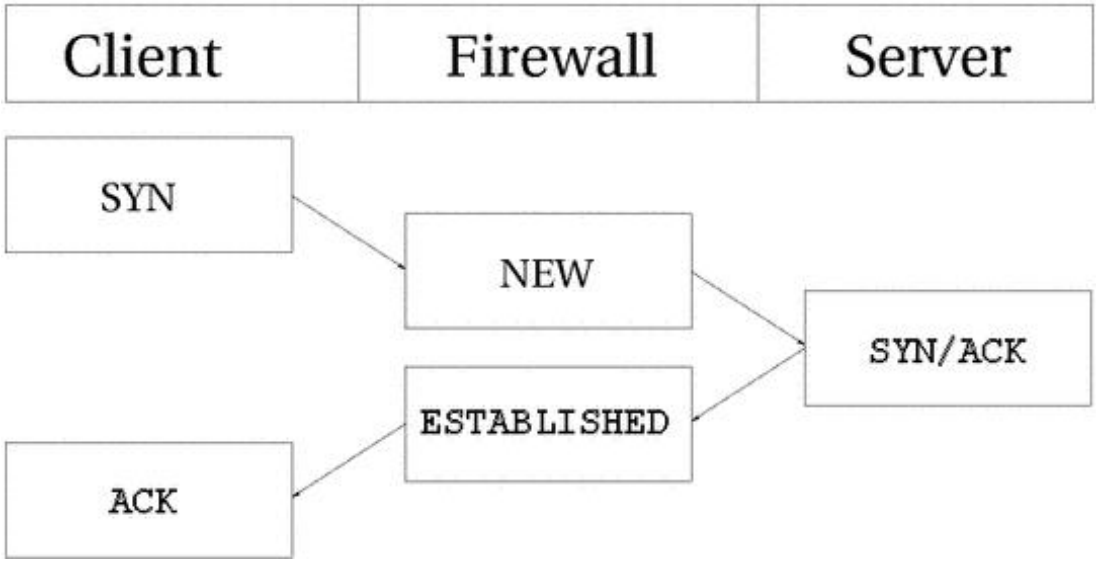


- How can the firewall distingush packets 1 e 2?

Connection tracking

- Distinguiching using the SYN is not convenient, we have seen that fragemntation attacks can prevent filering. moreover, it works for TCP only. There is a fundamental difference between the two packets:
  - packet 1 is received after sending an outgoing packet
  - packet 2 instead initiates the connection
- The conntrack module keeps track of these associations. Each packet (Of any type, UDP, TCP) is inserted into a connection that can be in 4 states:
  - NEW: The firewall has seen packets in a single directiona
  - ESTABLISHED: the firewall has seen packets in both directions
  - RELATED: for specific applications. The packet relates to another running connection
  - INVALID: none of the above

Connection tracking: state machine



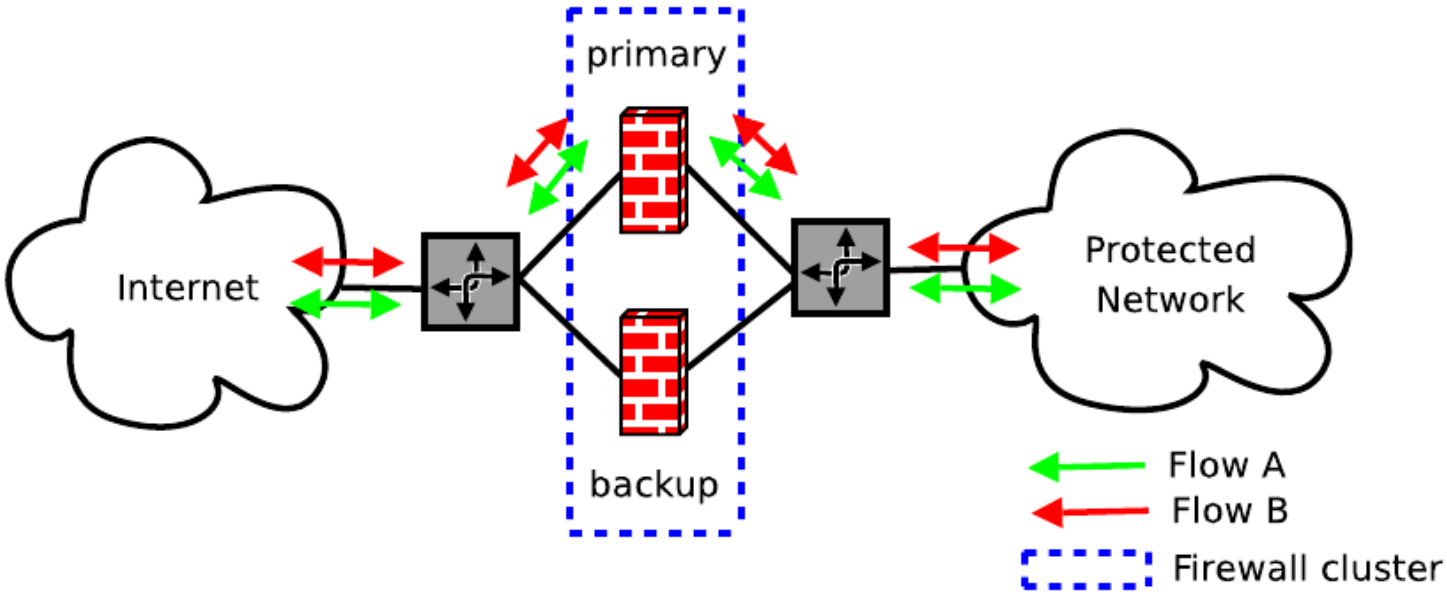
- iptables -A INPUT -j ACCEPT -p tcp -m state –state ESTABLISHED
- iptables -A OUTPUT -j ACCEPT -p tcp -m state –state NEW,ESTABLISHED
- iptables -P INPUT DROP

Fault Tolerance and Load Balancing

- The firewall is usually an input point of the network and can constitute a bottleneck.
- In networks that are subject to high volumes of traffic it is important to share the load between multiple firewalls to improve performance and resilience
  - cold swap backup: There are two firewalls, one is generally off and is switched on when the first one breaks
  - hot swap backup: The second firewall is always turned on
- All this can be organized with various configurations.

Primary-Backup configuration

- the gateway can distribute traffic to both the firewalls, the primary has a Virtual (VIP), which is what is seen from outside
- the backup firewall is generally off
- A *heartbeat* protocol (VRRP, HSRP ecc. . .) is used to check the state of the primary firewall. When it goes down for some reason the secondary firewall gets the VIP.



Primary-Backup configuration

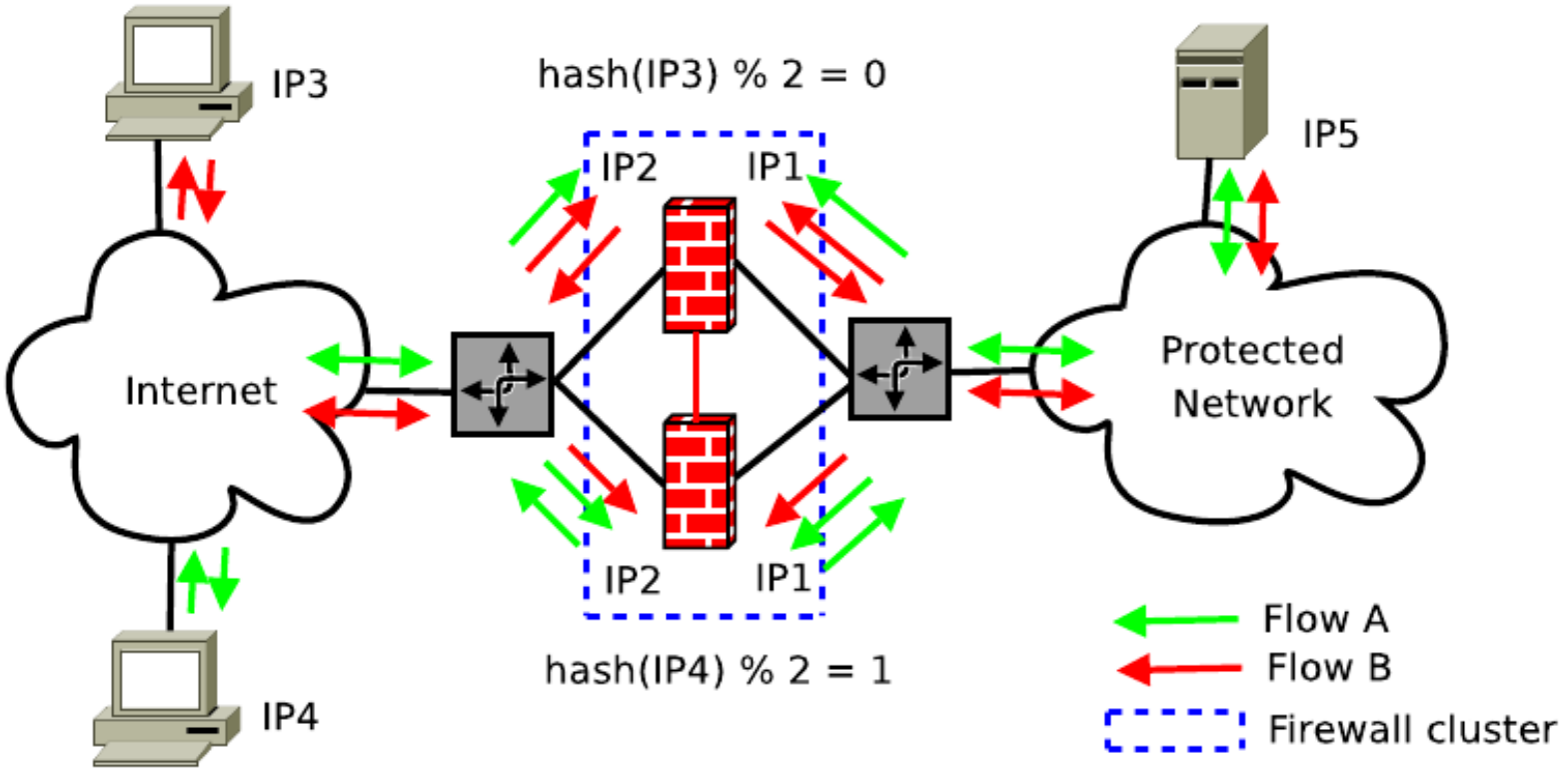
- There is no load balancing
- There is a down interval and all the running connections are lost
- There is a waste of resources since one firewall is usually off

Multi-primary multi-path firewall cluster

- It's the same as the previous case but there's a load balancer that distributes traffic flows on both firewalls, that are both on.
- There is load balancing
- If one firewall goes down, only its connections will be lost
- The redundancy problem moves to the load balancer

Multi-primary hash-based stateful firewall-clusters

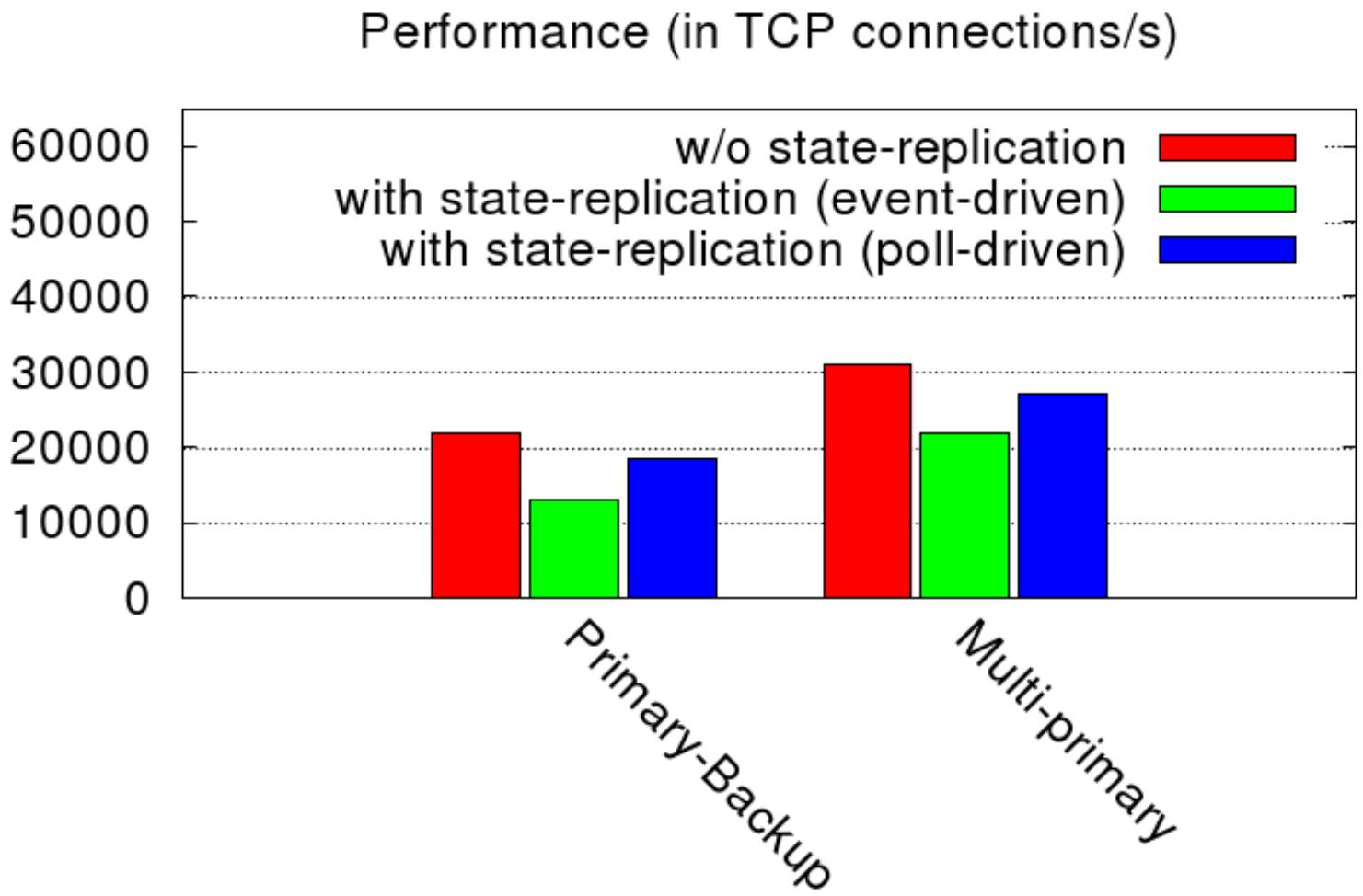
- There is no balancer. Each firewall has a numerical ID (0,1...) and filters a connection based on the evaluation of a tuple T = $IP_s, IP_d, Port_s, Port_d, Protocol$.
- For each ingoing connection if hash(T) % 2 == ID then the firweall filters the connection, else it lets it pass
- This way the firewalls share the load automatically
- You still need an heartbeat protocol to detect the failures.



State replication

- In all these situations, when a firewall fails, you lose the active connections at that moment.
- To avoid this it is necessary that the state machine for each connection, for each firewall are replicated in the other firewall. You can do this with two flavours:
  - Event based (each state change is communicated to the backup)
  - Periodic Updates

State replication



L7 Filtering

A network administrator may want to filter traffic level application several reasons:

- Logging and traffic analysis. You want to know what type of traffic is passing in your network to better organize your resources
- Traffic shaping. You want to give priority to certain flows than others.
- Block certain protocols. You want to prevent certain types of traffic to pass on your network.

L7 filterinf is necessary when it is not sufficient to use the port number in order to understand what kind of traffic is passing in your network

L7 Filtering

Filter Layer 7 protocols is very difficult:

- There are internal mechanisms of some protocols that make it difficult to relate different connections to the same session (FTP, SIP ...)
- There are protocols that intentionally try obfuscate their traffic (p2p)
- There are encrypted protocols.

Each filter must be modeled on the specific application and can have a complex state machine

L7 Filtering - challenges

- The state machines are complex enough and must operate on high speed links (gigabit links) There is a need for dedicated hosts dedicated with sufficient power.
- What if the protocol changes? you may have false positives (bad) or even false negatives (horrible!)
- A pattern matching algorithm implemented in software produces the same security concerns of other components of level 7 (i.e. buffer overflows). This is generally it is more difficult for firewalls working on fixed lenght fields for lower levels

L7 Filtering - vulnerabilities

Known vulns:

- **Snort RPC Preprocessing Vulnerability:** Researchers at Internet Security Systems (ISS) discovered a remotely exploitable buffer overflow in the Snort stream4 preprocessor module [. . .] Remote attackers may exploit the buffer overflow condition to run arbitrary code on a Snort sensor.
- **Trend Micro InterScan VirusWall Remote Overflow:** An implementation flaw in the InterScan VirusWall SMTP gateway allows a remote attacker to execute code with the privileges of the daemon.
- **Microsoft ISA Server 2000 H.323 Filter:** Remote Buffer Overflow Vulnerability. The H.323 filter used by Microsoft ISA Server 2000 is prone to remote buffer overflow vulnerability.
- **Cisco SIP Fixup Denial of Service (DoS):** The Cisco PIX Firewall may reset when receiving fragmented SIP INVITE messages.

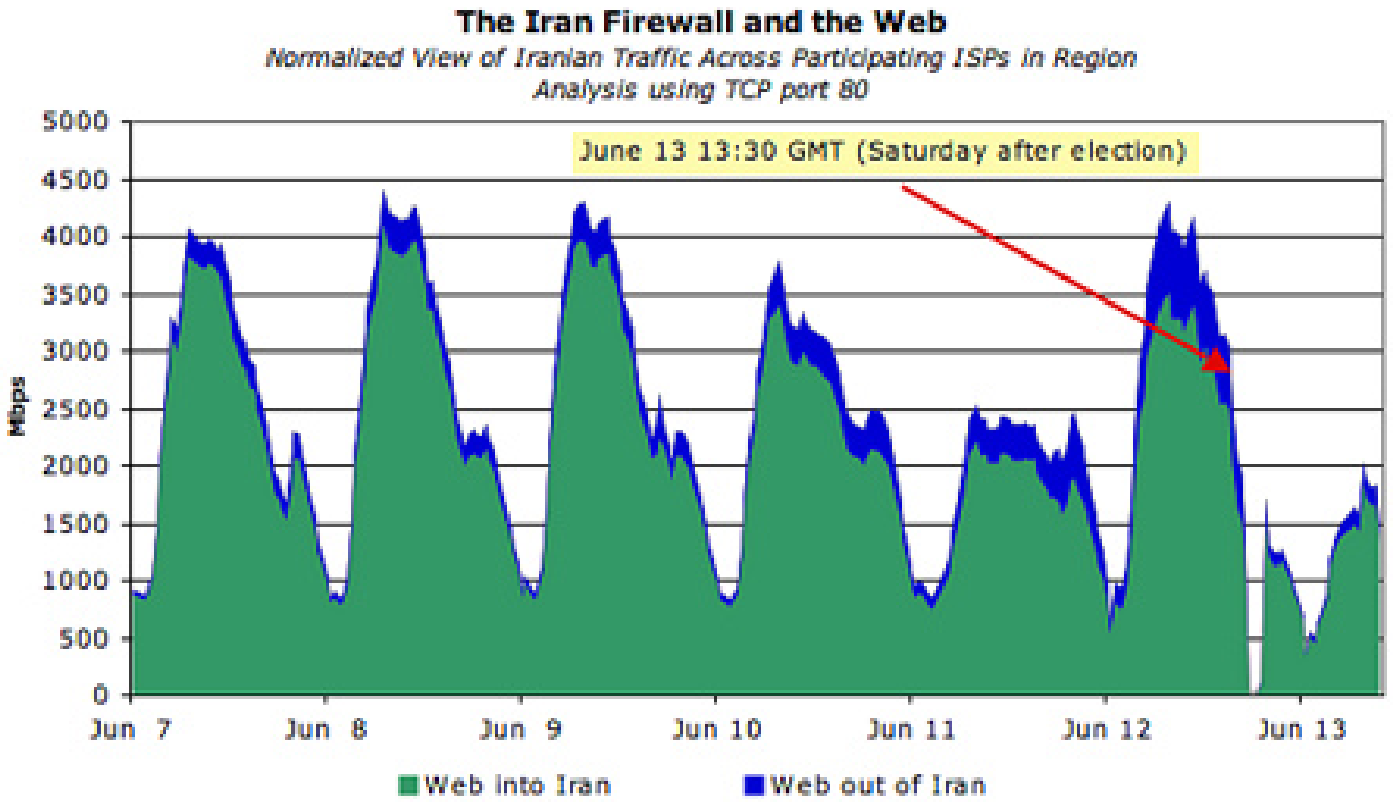
Side note, why L7 filtering?

There is a big discussion going on about *network neutrality*

- When the available bandwidth is not sufficient, you can buy more or try to do traffic shaping to give priority to some traffic kinds
- The ISP then decides which traffic kinds are more important than others, and the infrastructure is not neutral anymore
- This may lead to unfair treatment of services, or even to censorship

L7 Filtering - extreme example

June 13th, 2009: elections in Iran, images of violent repression are published on the Internet [4].



[4] images from http://asert.arbornetworks.com/2009/06/a-deeper-look-at-the-iranian-firewall/

Contact me!

@ leonardo.maccari@unitn.it

W^3 www.pervacy.eu

 @leobowski

Advanced Networking: networking Vs. privacy and security

Leonardo Maccari

leonardo.maccari@unitn.it

18 dicembre 2012