Network security

Nicolò Fornari

June 14, 2016

Chapter 1

Network layer

1.1 Subnets

Subnets are logical divisions of IP addresses. IP bits are partitioned as network,subnet,host.
A subnet mask indicates sections of IP addresses meant for network and subnet.
Eg. 255.255.255.0 means 24 bits for network and subnet and 8 bits for hosts.

CIDR
Classless Inter Domain Routing, it is a synthetic way to represent subnet masks.
Example:

- Network mask: 255.255.0.0.
- CIDR representation: 132.132.1.10/16
- Hosts = 2^{16}

Formulas: (everything as binary)

- Network = Ip AND Subnet
- Host = Ip AND Not(Subnet)

Some IPs are reserved for private networks:

- 10.0.0.0 → 10.255.255.255
- 192.168.1.1 → 192.168.255.255
- 172.16.0.0 → 172.16.255.255

Def A *datagram* is a basic transfer unit associated with a packet-switched network. The delivery, arrival time, and order of arrival need not be guaranteed by the network.

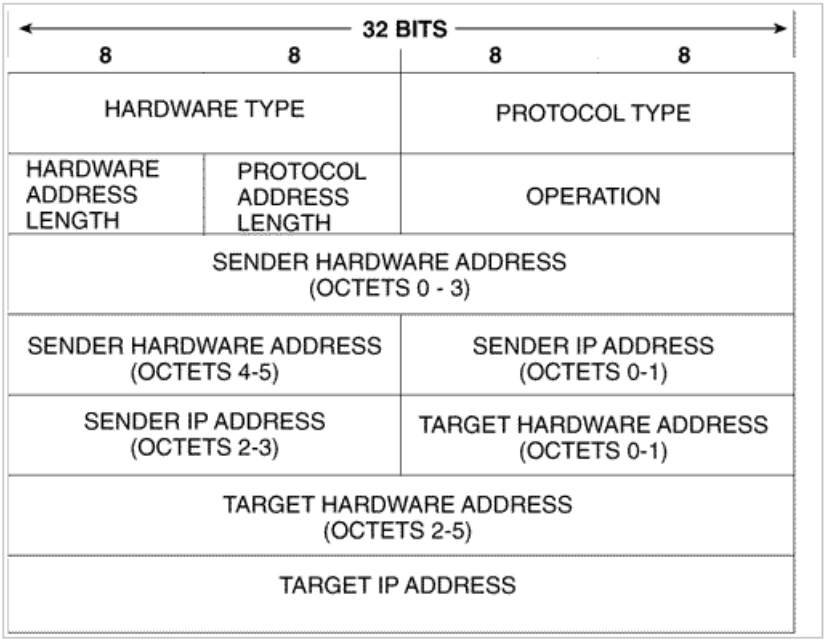
Def MTU maximum transmission unit

1.2 ARP

ARP (address resolution protocol) allows systems to associate an ip address to a MAC address. All addresses in the ARP table are added by one of these mechanisms:

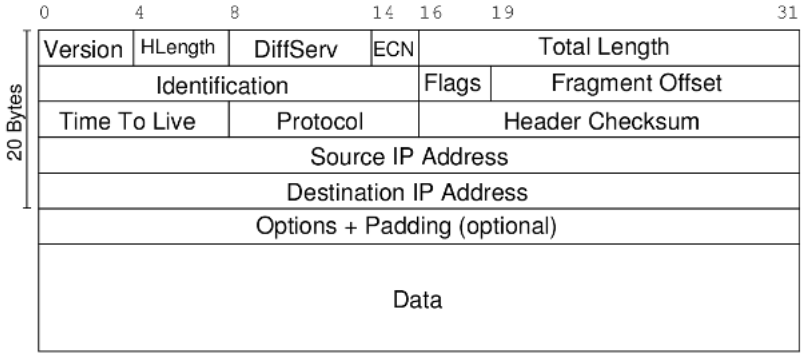
- ARP request-reply:
  who is 192.168.0.16 tell 192.168.0.1
  192.168.0.16 is at 00-10-BC-2c-11-56
- Gratuitous ARP
  192.168.0.16 is at 00-10-BC-2c-11-56

ARP frame header



ARP poisoning
The ARP protocol is declarative, it does not need an answer.
Nodes are not authenticated.
Limitations: it works only on LAN

1.3 IP

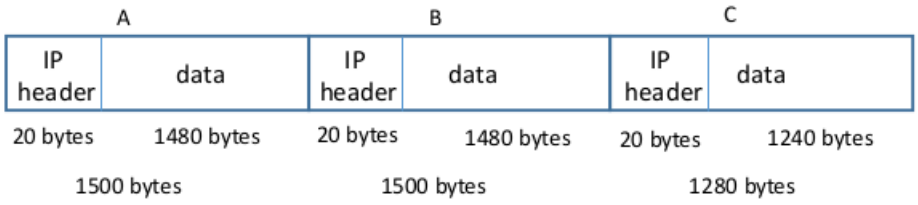


IP fragmentation *Identification:* 16 bit, is the unique identifier of the fragmented datagram. Note that all fragments have the same identification number.
Flags: 3 bits

- 0 Reserved, must be zero
- DF Don't fragment
  If set to 0 → there may be fragments
  If set to 1 → drop datagram if it has to be fragmented
- MF More fragments
  0 → last fragment
  1 → there are more fragments

Offset 13 bits, offset of this datagram wrt the first fragment with that ID

Fragmentation example

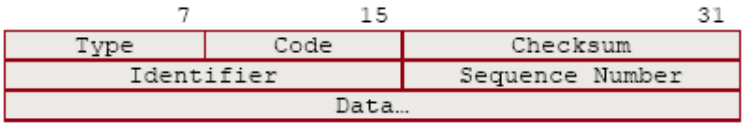


	A	B	C
Identification	4452	4452	4452
Flags	• DF=0 • MF=1	• DF=0 • MF=1	• DF=0 • MF=0
Offset	0	1480	2960

Remark 1. ***DOS with IP fragments*** *You keep sending fragments without sending the first fragment, the router keeps waiting for it until it exhausts its memory.*

1.4 ICMP

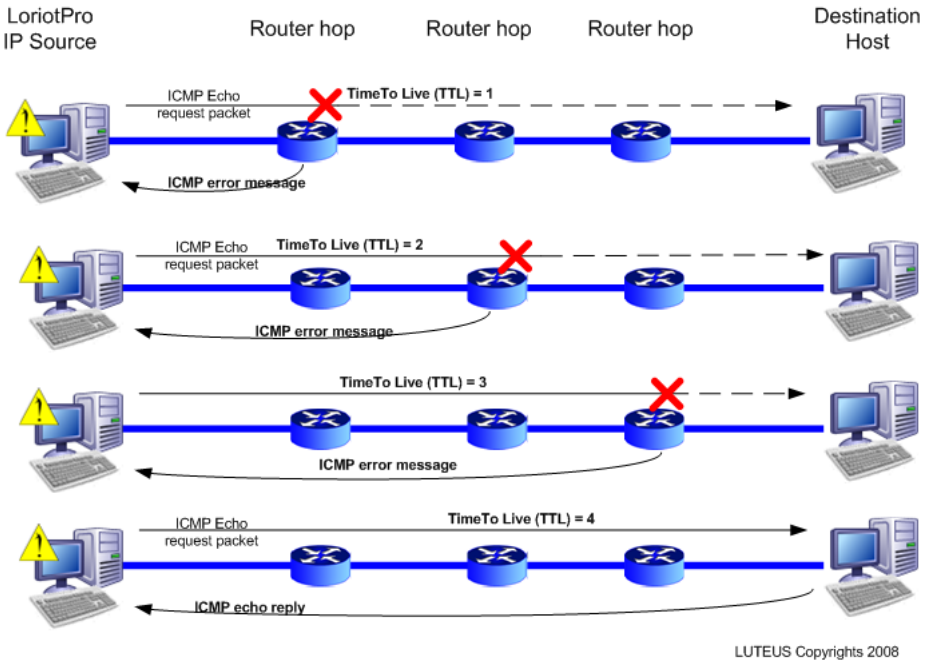
Internet control message protocol. It relies on IP and is and integral part of it.



Some message types

- 0 Echo Reply
- 3 Destination Unreachable
  Code 0 → Net unreachable
  Code 1 → Host unreachable
  Code 2 → Protocol unreachable
  Code 3 → Port unreachable
  Code 4 → Fragmentation needed and DF set
  Code 5 → Source route failed
- 8 Echo
- 11 Time Exceeded
  Code 0 → Net unreachable
  Code 1 → Host unreachable

1.5 Traceroute



1.6 Denial of Service

Def a Denial of Service is a type of attack that aims at congesting or overpowering a system's capacity by generating requests the system will have to answer.

Examples

- DoS with IP fragmentation
- Ping Flooding (the attacker exploit his wider bandwidth)
- Ping of death
- SYN DoS
- ReDos (RegExp DoS)
- DNS amplification attack
- Reflected DoS
- Coremelt

1.6.1 Ping of death

A correctly-formed ping packet is typically 56 bytes in size. However, any IPv4 packet (including pings) may be as large as 65,535 bytes. Some computer systems were never designed to properly handle a ping packet larger than the maximum packet size because it violates the Internet Protocol. Like other large but well-formed packets, a ping of death is fragmented into groups of 8 octets before transmission. However, when the target computer reassembles the malformed packet, a buffer overflow can occur, causing a system crash and potentially allowing the injection of malicious code.

1.6.2 SYN DoS

When the server receives SYN J it answers back with SYN K, ACK J+1.
The server opens a new session in a separate threat/ allocates resources. Then the server waits for the ACK K+1 from the client, it waits for the MSL (maximum segment lifetime set by default to 2 minutes). The same mechanism is on the sender side but of course the attacker controls it and can bypass it.

The attacker drops all SYN ACKs with a firewall to avoid exhausting its own resources. Note that a server typically has more bandwidth than a single client.

Remark 2. *The attack currently works in $\mathcal{O}(2n)$ because for each SYN a SYN ACK is received $\rightarrow$ less bandwidth.*

However the attack can be improved by spoofing the source ip address, now the attacker is operating in $\mathcal{O}(n)$.
In theory the attack should not work because the server should receive a RST by each zombie, consequently free the TCB making the attack fail.
Still the attacker can choose a set of IPs which do not reply.

1.6.3 Advanced DoS

Reflected DoS A reflector is a server that will reply to any receiving packet. Nowadays you would need a lot of bots to saturate all the bandwidth. instead of infecting million of bots you build a list of reflector that the slaves can contact. It is a classic IP spoofing scenario.

Amplification DoS
Eg. Network Time Protocol

1.6.4 Mitigations

Source identifications
It is a kind of traceback. It is quite difficult

Capabilities: capability is made of marks (unique hash values) set by routers on the path from sender to receiver. Routers check validity of marks upon response. Limitation: capability can be saturated

Other mitigations:

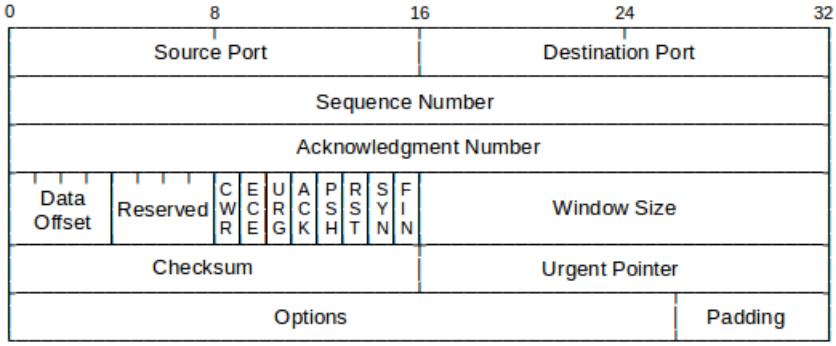
- Load balancing: distribute traffic loads evenly
- Rate limiter: deny traffic above a certain rate of SYN/sec
- Proof of work: require the source to solve a cryptopuzzle before allocating resources for the connection (note that this requires a protocol support).

Chapter 2

Transport layer

2.1 TCP

The Transmission Control Protocol builds on top of IP the notion of state. Infact IP just delivers data while TCP manages the data segments by mean of checksums and re-delivery of unreceived/corrupted packets.

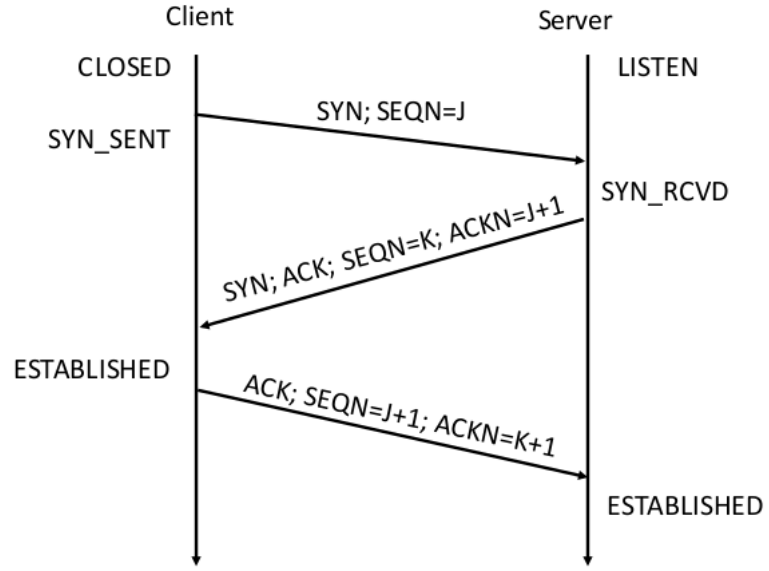


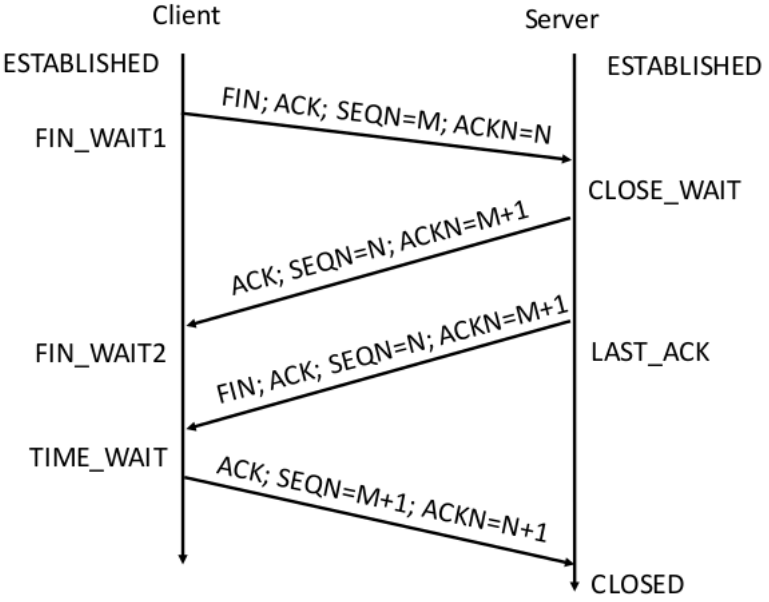
A server and a client that partecipate in a TCP connection open a *socket* which is a tuple (source-ip:source-port,destination-ip:destionation-port).
Note that a client generates a source port randomly.
TCP details - some flags:

- **SYN:** initializes the TCP session, it should be set to 1 only for the first datagram
- **ACK:** ackwnoledge the reception of the segment
- **FIN:** it signals the intention of closing the connection
- **RST:** drop the connection (reset)

Sequence number: it is a 32 bit generated by each end (note that the sequence number of the client is independent from the server's one).
Acknoledgement number: 32 bits

2.1.1 Handshakes





2.1.2 Some TCP specifics

Both client and server set up a TCB (Transmission control block) to keep track of connection. The TCB structure is freed from memory when connection reaches status CLOSED.

A packet with RST flag up does not receive an answer.
If the state is CLOSED any packet with no RST receives a RST.
If the state is LISTEN

- SYN flag up + no ACK opens a TCP session. Answer is SYN+ACK
- Only ACK receives a RST
- Drop with no answer otherwise

2.1.3 TCP scans

It is possible to exploit specifications of a network protocol (TCP,UDP,..) to learn something about a system or a network.

SYN Scan: the attacker forges TCP packets with SYN=1. It is useful to see whether the remote system accepts incoming connections on a certain port.
Note that a polite way of scanning is performed in the following way: after the server's SYN ACK reply, the attacker sends a RST so that the 3-way handshake is never finished.

Host fingerprinting: different operating systems have their own independent implementation of the TCP stack.

Examples:

- FIN = 1
- all flags set to 0
- Xmas: FIN,URG,PSH = 1

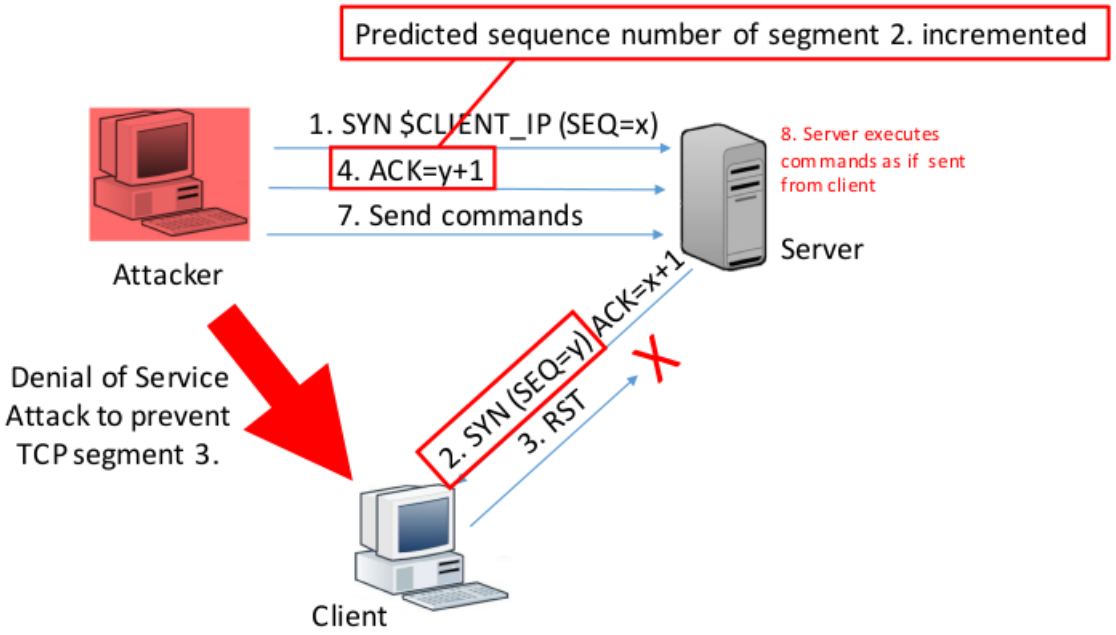
2.1.4 Mitnick attack

The attacker wants to send commands to a server they have no access to (eg. simple IP address authentication). The server has to think the attacker is the client, however the attacker does not sit in between client and server.

A TCP segment is identified and validated by:

- client ip → known
- destination ip → known
- port → known (if not standard, scan)
- client SEQ number → known(the attacker generates it)
- server SEQ number → *unknown*

Back in the old days algorithms for generating the SEQ number were really simple.



2.2 UDP

The User Datagram protocol is stateless. It is designed for fast delivery of data

- Data integrity can be controlled at application level
- It relies on the reliability of the underlying network link
- It does not guarantee delivery (there is no ACK mechanism)

Usage

- DNS servers
- NFS (network file systems)
- SNMP (simple network management protocol)
- DHCP (dinamic host configuration protocol)
- Most real time applications

2.2.1 UDP scans

It can be used to discover open ports on the network:

- CLOSE → ICMP port unreachable
- OPEN → no answer

However it is possible to configure a stealth system that does not reply to UDP requests to CLOSED ports by dropping ICMP packets with a firewall or router.

2.3 DNS

DNS (Domain Name Service) is a hierarchical system for domain name resolving. It translates human readable addresses to IP addresses the domain is reachable at. It uses UDP for fast answer (port 53).

Motivation: it is possible to assign multiple names to the same ip or viceversa. This flexibility is useful for:

- Server substitution
- Virtual hosting (a single server hosting multiple websites)
- A name is assigned to multiple IPs (so that the workload is distributed).

2.3.1 Terminology

Zone
It is a collection of hostname-IP pairs all managed together.
Example: google.com, mail.google.com, drive.google.com are all part of the google.com zone.

Nameserver
It is a server software that answers DNS questions. Sometimes the nameserver knows the answer directly ($\rightarrow$ authoritative), sometimes it has to ask for the answer ($\rightarrow$ recursive).

Authoritative nameserver
For every zone somebody has to maintain a file with the hostname-IP associations. This is generally an administrative function performed by a human and in most cases one machine called *zone master* has this file.

Resolver
This is the client part of DNS. In Linux systems the location of the servers-to-ask is $etc/resolv.conf$

Resource record
There exist different kind of DNS records, here we list a few:

- **A** correspondence name - IP(s)
- **AAAA** Same as A but works for ipv6
- **NS** ip of the DNS server to ask
- **MX** mail exchanger
- **SOA** start of authority

Delegation
When a nameserver does not have the contents of a zone, but knows how to find the owner, it is said to delegate service of that zone to another nameserver. Informally, it is a pass-the-buck mechanism.

Remark 3. *DNS has a distributed nature: no machine knows everything, but they do know how to find each other via delegation. A bad guy could set up a nameserver and configure an authoritative zone for BankOfSteve.com, but it has no effect because no higher-level nameserver ever delegates to it.*

2.3.2 Following a simple DNS query

1. The client requests an A record for a domain - unixwiz.net. The client is routed to the nameserver provided by the user's ISP. The ISP's nameserver is not authoritative for unixwiz.net so it can not look it up in its local zone db and also does not find it in its cache.
2. All recursive nameservers are preconfigured with a list of 13 root servers. The nameserver picks one a t random and sends off the query.
3. The root server does not know the answer but returns the GTLD (global top level domain) servers responsible for .net
   This is in the form of NS records of servers. Though we technically asked only for the NS records the root servers also give us the ip of each of them ($\rightarrow$ this is know as *glue*).
4. The nameserver chooses one of the authoritative servers at random and sends off a new request
5. The GTLD server does not know the answer but it knows how to get us closer: like the root servers it sends back a referral (a set of NS records) that are likely to have what we seek.
6. Once again the nameserver sends off the requests to one of the servers in the last referral
7. This time the server provides the A record for unixwiz.net, in addition the response includes a flag saying "this is an authoritative response".
8. Finally the ISP's nameservers hands the answer back to the client. Note that the reply to the client does not include the authoritative indicator because the ISP's nameserver can not pretend to be the source of authority.

2.3.3 DNS Cache poisoning

DNS only accepts responses to pending queries, unexpected responses are simply ignored. How does a nameserver know that any response packet is "expected"?

- The response arrives on the same UDP port we sent it from
- The Question section (which is duplicated in the reply) matches the Question in the pending query
- The Authority and Additional sections represent names that are within the same domain as the question (known as "bailiwick checking"). This prevents ns.unixwiz.net from replying with not only the IP address of www.unixwiz.net, but also fraudulent information about (say) BankOfSteve.com.

If all of these conditions are satisfied, a nameserver will accept a packet as a genuine response to a query, and use the results found inside.

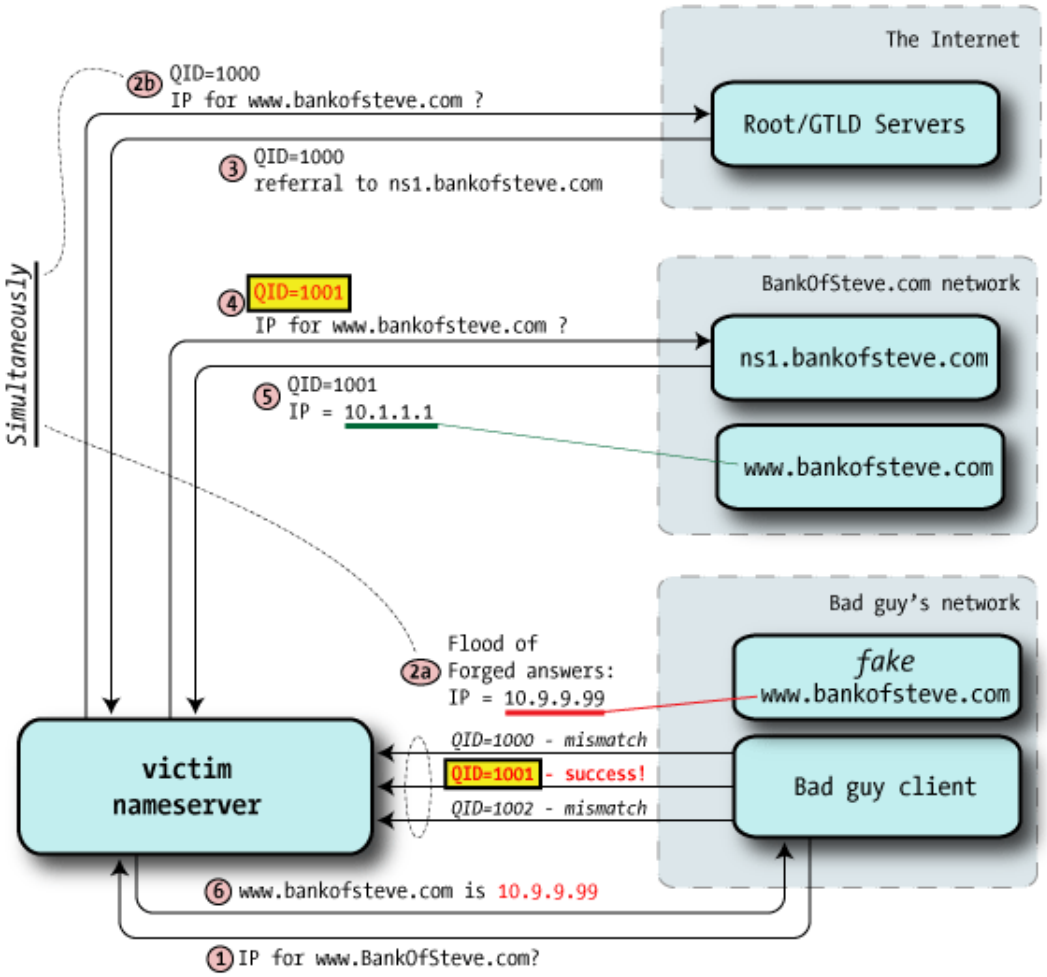
Guessing the query ID

1. The attacker asks the victim nameserver to look up a name in a zone for a nameserver he controls
2. Victim nameserver receives the request and makes the usual rounds to resolve the name starting at the root servers

3. Eventually, the victim nameserver will be directed to the attacker's nameserver
4. The attacker monitors the traffic going to its own machine and from this discovers the source port and Query ID used

Then the attacker sends a DNS query to the victim nameserver for the hostname it wishes to hijack. Knowing that the victim will shortly be asking ns1.bankofsteve.com (as directed from the root/GTLD servers) for an IP address, the attacker starts flooding the victim with forged DNS reply packets. All purport to be from ns1.bankofsteve.com, but include the answer with the IP of attacker's fraudulent webserver. The rule is: *first good answer wins.* Most of the forged answers are dropped because the Query ID doesn't match, but if just one in the flurry of fake responses gets it right, the nameserver will accept the answer as genuine. *This is a race, but only the attacker knows it's playing.*

Remark 4. *The name can not be already in the cache (otherwise the attacker has to wait for it to expire as determined by the TTL). The attacker has to guess the query ID: this is easy with (now-obsolete) nameservers that increment the Query ID by one each time. The attacker has to be faster than the real nameserver.*



2.3.4 Kaminsky attack

Dan Kaminsky found an approach that is dramatically more effective than simple cache poisoning: the key difference is the nature of the forged payload. Rather than poisoning the final answer, the A record with the IP address, Dan discovered that we can go up one level and hijack the authority records instead. Before undertaking the attack, the attacker configures a nameserver that is authoritative for the bankofsteve.com zone, including whatever resource records he likes: A records, MX for email, etc. There's nothing stopping anybody from configuring his own nameserver to be authoritative for any domain, but it's pointless because the root servers won't point to it: it's got answers, but nobody ever asks it a question.

1. The attacker requests a random name withing the target domain - something unlikely to be in the cache

2. The attacker sends a stream of forged packets to the victim, but instead of A records as part of an Answer, it instead delegates to another nameserver via Authority records. "I don't know the answer, but you can ask over there". The authority data may well contain the "real" bankofsteve.com nameserver hostnames, but the glue points those nameservers at attacker's IPs. This is the crucial poisoning, because a Query ID match means that the victim believes that attacker's nameservers are authoritative for bankofsteve.com. The attacker now owns the entire zone.

This is a devastating attack: by owning the entire target domain, the bad guy controls essentially everything with respect to that resolving nameserver. He can redirect web visitors to his own servers (imagine redirecting google.com), he can route email to his own servers via serving up bogus MX records.

Mitigation: the source of attack is low entropy with a 16 bit ID. Randomness is not enough and it is not feasible to change the protocol to 32 bits.
The solution is to randomize the source port to increase the entropy: any answer that does not match both source port and transaction ID will be dropped.

2.3.5 DNS amplification attack

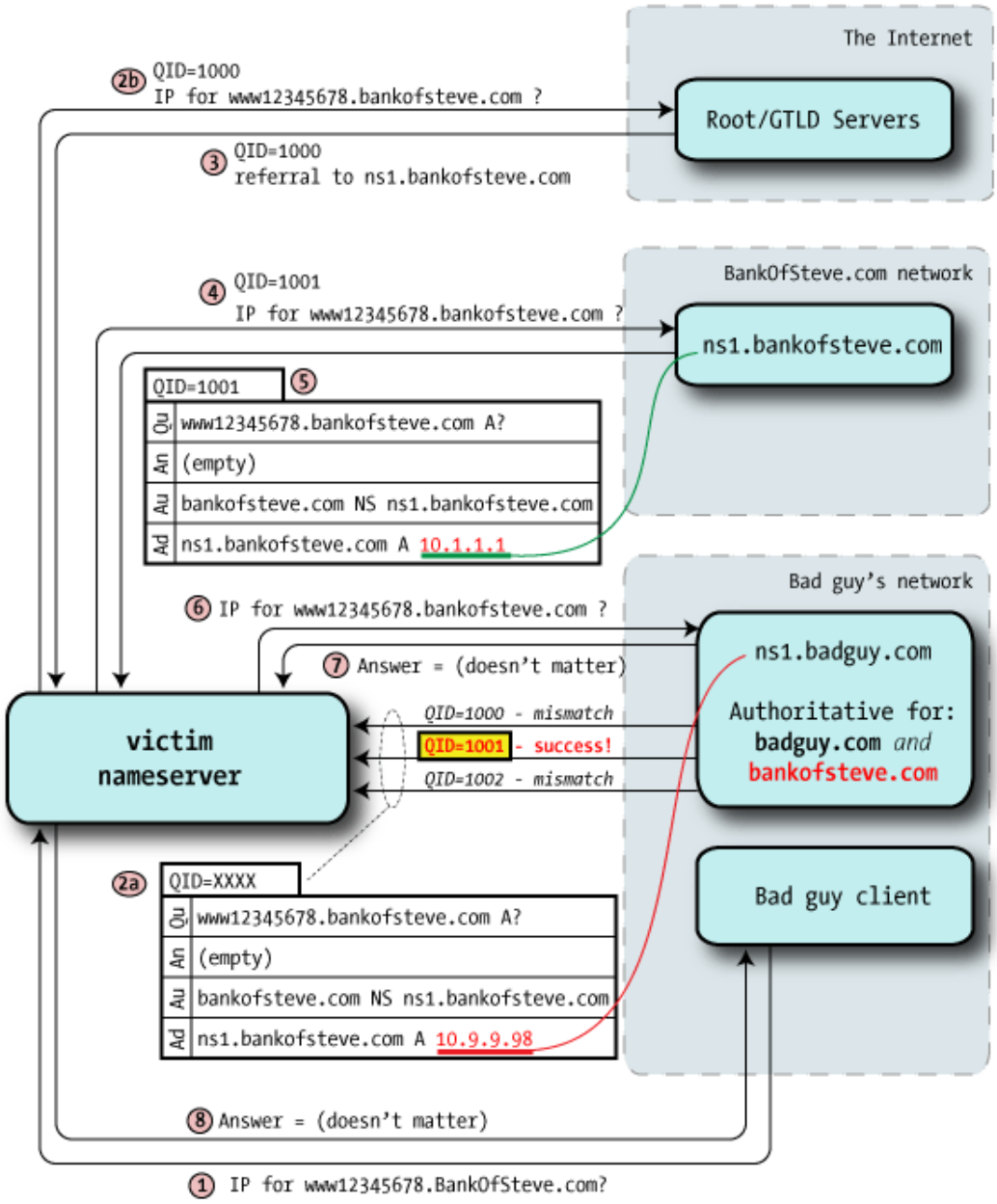
It is DoS attack that exploits certain types of DNS answers that are much bigger in size than the requests. Recall that the DNS works over UDP so the source IP is easy to spoof.

2.3.6 DNS zone transfer

A zone is a domain for which a server is authoritative. "Slave" servers can ask "authoritative" servers to copy their zone database (over TCP).
An attacker pretends to be a slave server and dump the zone DB. In this way he acquires knowledge of zone's infrastructure and it is eased in performing further attacks.

2.3.7 DNS Sec

It is a secure implementation of the DNS protocol: it implements DNS auth on top of normal DNS. Note that it protects just integrity and not confidentiality.


The Internet
Root/GTLD Servers
2b QID=1000
IP for www12345678.bankofsteve.com ?
3 QID=1000
referral to ns1.bankofsteve.com
BankOfSteve.com network
ns1.bankofsteve.com
4 QID=1001
IP for www12345678.bankofsteve.com ?
QID=1001
5
Qu www12345678.bankofsteve.com A?
An (empty)
Au bankofsteve.com NS ns1.bankofsteve.com
Ad ns1.bankofsteve.com A 10.1.1.1
Bad guy's network
6 IP for www12345678.bankofsteve.com ?
7 Answer = (doesn't matter)
ns1.badguy.com
Authoritative for:
badguy.com and
bankofsteve.com
victim
nameserver
QID=1000 - mismatch
QID=1001 - success!
QID=1002 - mismatch
Bad guy client
2a QID=XXXX
Qu www12345678.bankofsteve.com A?
An (empty)
Au bankofsteve.com NS ns1.bankofsteve.com
Ad ns1.bankofsteve.com A 10.9.9.98
8 Answer = (doesn't matter)
1 IP for www12345678.BankOfSteve.com?

Chapter 3

Firewall

Firewalls are perimetral network components that filter incoming (outgoing) traffic from (to) the network. They have different policies:
Default deny: nothing is allowed to pass expect what is permitted
Default permit: everything is allowed except for what is denied. It is suitable for open organizations like universities or home systems.

3.1 Static packet filtering

Header information is used for filtering (eg. protocol number,source and destination ip and port numbers). Stateless: each IP packet is examined isolated from what is happened in the past.
It is often implemented by routers, it is simple and fast and has a low demand on resources.
Standard ACL: $ number $ action $ src (wildcard)

3.2 Stateful packet filtering

It looks at the history of the packets passing through the firewall.
Stateful firewalls allow user to define states over stateless protocols. For these protocols there is no termination sequence (eg. TCP FIN 4-way handshake) so a timeout is implemented, depending on the application.
Possible states

- NEW → packet trying to open a new connection
- ESTABLISHED → incoming packet is realtive to a connection already initiated
- RELATED → new connection related to an existing one (eg. FTP)
- INVALID → none of the above (eg. incoming packet with ACK but not belonging to ESTABLISHED connection)

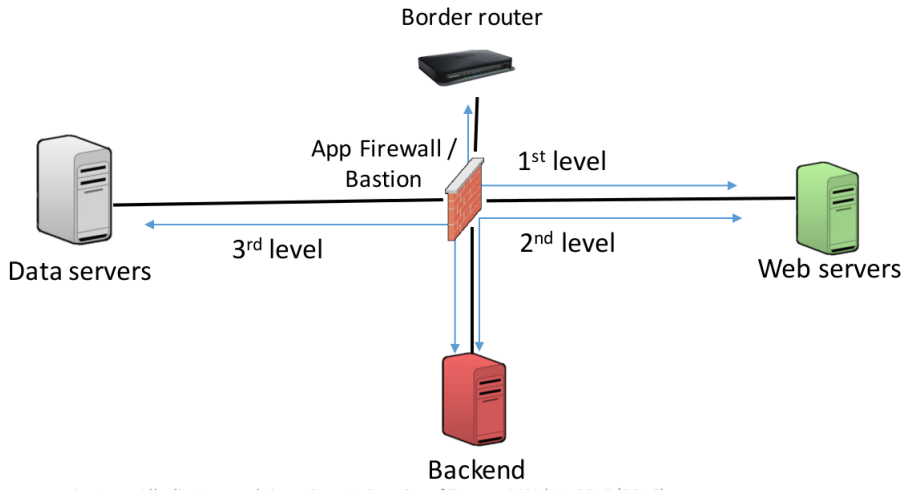
With a stateful firewall you can prevent ACK scans (OS fingerprinting).
Pros: more powerful rules, policy definition is much simpler, very diffused.
Cons: severe impact on firewall performance, deep packet inspection slows down packet check, application support may be very complicated.

3.3 Network topologies

If you want a firewall to work you have to put it in the right place

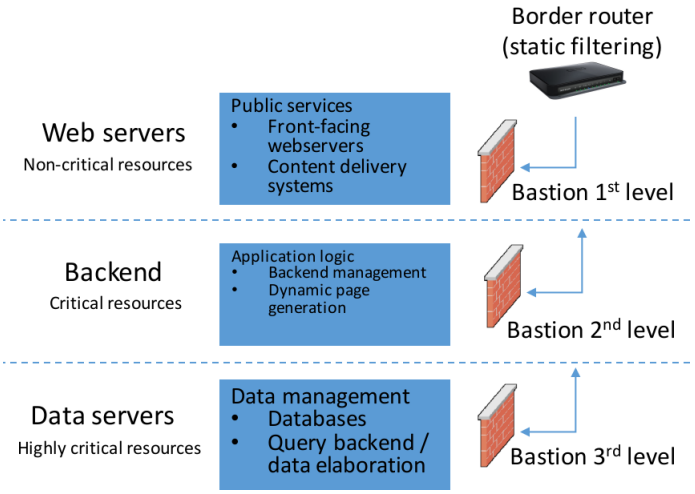
3.3.1 Simple configuration

Border router implements static inward and outward filtering. The best policy is to drop packets with no answer. Eg. do not allow packets whose final destination is the firewall. The firewall has several inward facing network interfaces. Each interface is dedicated to a network level. It is a single point of failure.

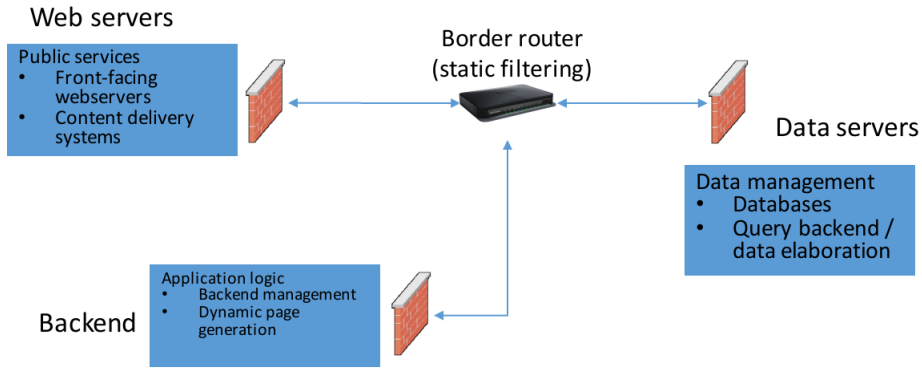


3.3.2 Divide et impera: cascade

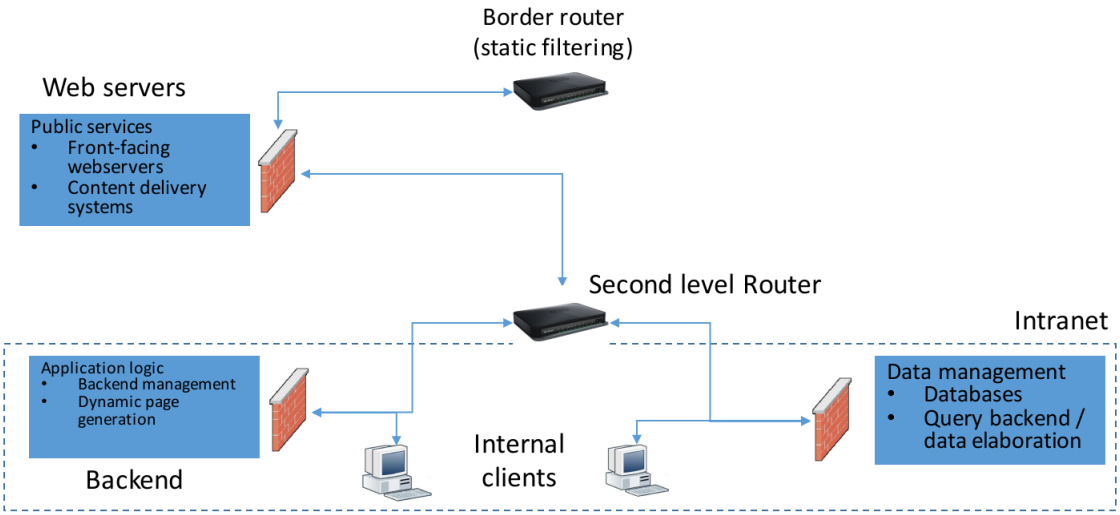
Interdependent firewall policies. Eg. firewall at level 1 must allow packets eventually directed toward level 2. It requires a good mixture of NAT and PAT policies,firewall configurations and good separation of services. It has high design, management and mainteinance costs.



3.3.3 Divide et impera: parallel



3.3.4 Mixed



3.4 Proxy

Def. It is a network component that mediates network communications. Proxy acts both as a client and server.

Open proxy
Proxy connects any client on the internet to any server on the internet. It enables the user to achieve some level of anonymity on the network. Note that there are techniques that allow the server to know the client's ip thorugh third parties (eg. flash).
Trust issues → all trust is put on proxy service (it is ok to bypass organisation's blacklist) but not trustworthy for sensible internet traffic. It may also be used for malware distribution.

Reverse proxy
The situation is "double blind": the client thinks he is talking to the (lets say) FTP server but actually he is talking to the reverse proxy.
Pros: hide properties of internal servers (IP, non custom service ports, versioning - if too aggressive it may cause disservice eg. declaring a fake server version breaking the protocol). It may be used for load balancing (several internal replicas of a webserver) or caching server's content. Furthermore, a reverse proxy can provide SSL encryption for an arbitrary number of hosts; removing the need for a separate SSL Server Certificate for each host, with the downside that all hosts behind the SSL proxy have to share a common DNS name or IP address for SSL connections.

Application level gateway
Pros. information hiding, robuts authentication and logging, cost effective,less complex
Cons. keeping up with new applications, custom implementations may be expensive.

Circuit level gateway
Circuit-level gateways and have the advantage of hiding information about the private network they protect. On the other hand, they do not filter individual packets.

Network address translation
NAT operates at level 3 and acts as a reverse proxy. It maps (sourceIP, sourcePort) to (destIP,destPort). Port address translation (PAT) to resolve conflicts.

Remark 5. *Most of the time proxy refers to a layer-7 application while NAT operates at layer-3.*

3.5 Bastion host

It is a system identified by the firewall administrator as a critical strong point in the network security.

- Secure version of OS (hardened system)
- Only essential services
- Each proxy maintains detail audit information
- Each proxy runs as a non-privileged separate user

Administration

- Access with secure connecion
- Logging: on a remote logging server
- Strategies of disaster recovery
- Security incidents

3.6 Firewall configuration errors

Possible configuration errors:

- **No stealth rule:** from anywhere to the firewall with any service DROP
- **Implicit rules:** control both DNS and ICMP
- **Insecure firewall management:** access to the fw over insecure,unencrypted and poorly authenticated protocols.
- **Too many management machines:** fw should be managed from a small number of machines
- **External management machines**
- **NetBios service**
- **Portmapper/Remote procedure call**
- **Zone spanning objects:** addresses that reside on more than one "side" of the firewall. For a firewall with more than two interfaces each interface defines a "side"
- **Any service on inbound rules:** allowing any service to enter the network
- **Any destination on outbound rules:**

Chapter 4

Intrusion detection system

There are three phases

- Data collection
  It can be host based or network based
- Data analysis
  Two distinct approaches: misuse detection (list unwanted behaviour) or anomaly detection (build an average profile, report if current activity is significantly different from average)
- Action
  Report, log entry and (just IPS) block/alert

Signature based
It is the IDS equivalent of "default allow" policies. There are "blacklist" patterns that are believed to be related to malicious activities (eg. system calls and payloads in network protocol).
Pros. straight from the box solution
Cons. it is easy to fool signature based solutions. The more advanced the signature db is, the more computional power is needed and more false positive are triggered.

Anomaly based
It assumes intruder behaviour differs from legitimate profile. Problem: users evolve, a new legitimate activity may look suspicious.
Pros. this method is particularly good at identifying sweeps and probes towards network hardware giving early warnings of potential intrusions.
Cons. requires more hardware spread further across the network than is required with signature based IDS.

External
- Analysis of all incoming traffic
- Only general signatures
- All "detected attacks" are logged

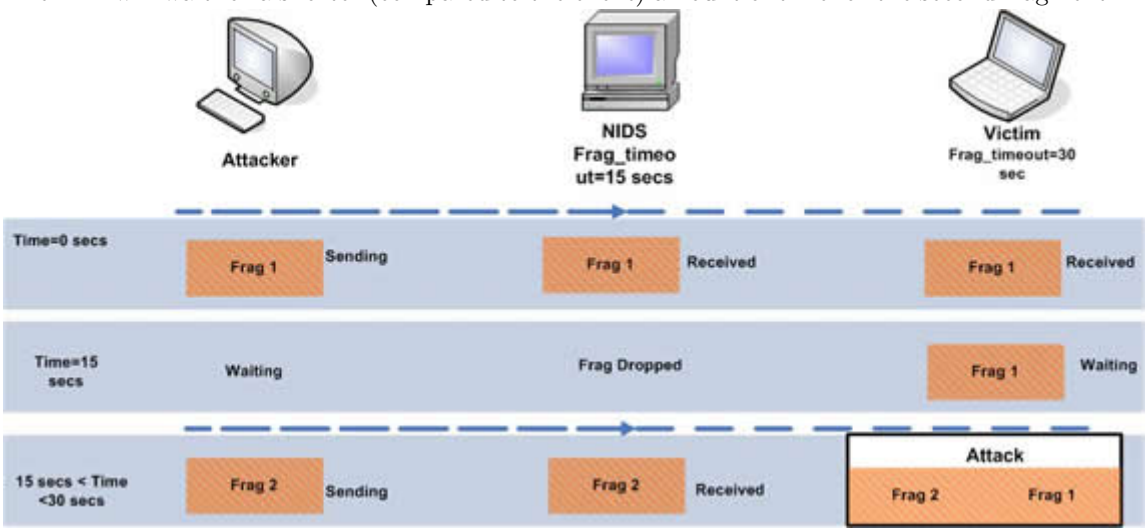
Internal
- Analysis only of traffic allowed by the firewall
- More specific signatures are possible
- It says nothing about attacks attempted but blocked by the firewall

4.1 NIDS evasion

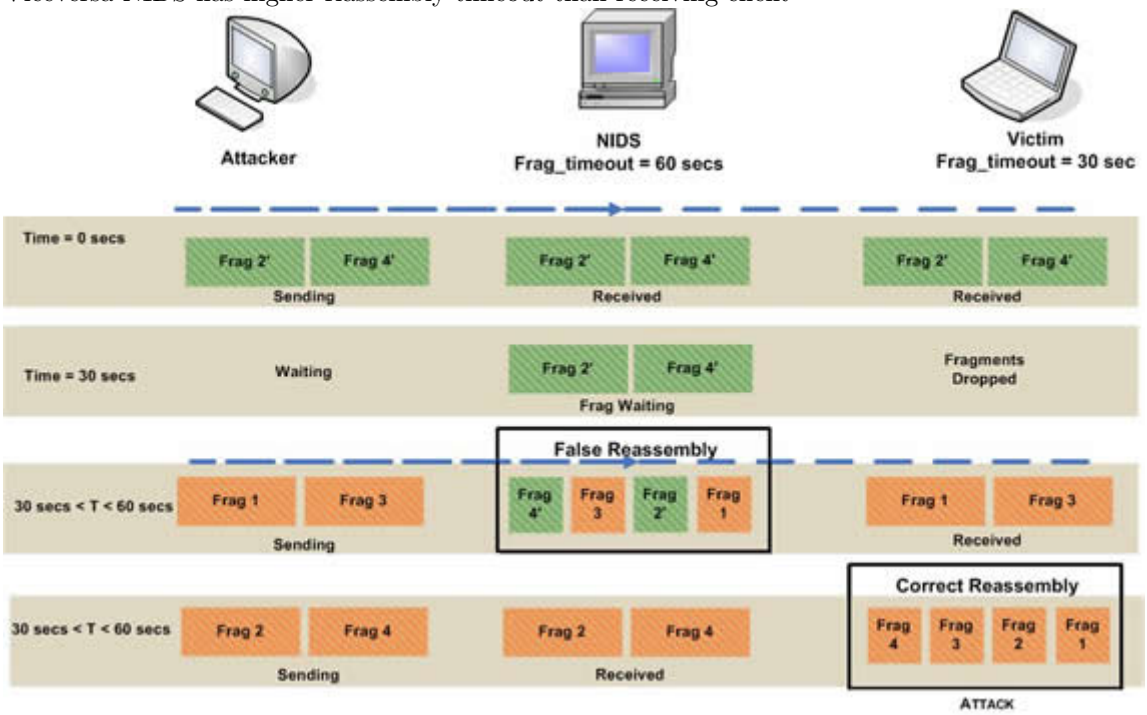
Signature based evasion can be fairly trivial: for instance "*/bin/bash*" will be detected while "*/etc/../bin/bash*" will not. This is a proof of concept: of course NIDS could use regexp or wildcard. More advanced techniques are typically based on IP fragmentation.

4.1.1 Reassembly time-out

The NID will wait for a shorter (compared to the client) amount of time for the second fragment.

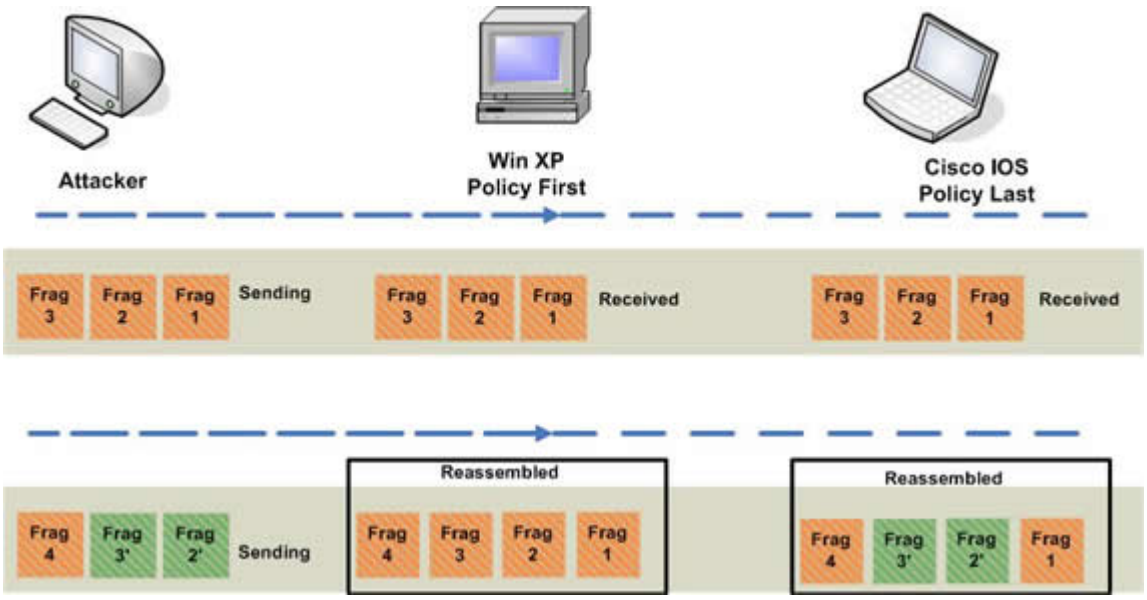


Viceversa NIDS has higher riassembly timeout than receiving client



4.1.2 Time to live

4.1.3 Fragment replacement



4.1.4 Snort

SNORT is a free and open source network intrusion detection and prevention system. It has the ability to perform real-time traffic analysis and packet logging on Internet Protocol networks. It performs protocol analysis, content searching, and content matching. The program can also be used to detect probes or attacks, including, but not limited to, operating system fingerprinting attempts, common gateway interface, buffer overflows, server message block probes, and stealth port scans. There are three main modes in which Snort can be configured: sniffer, packet logger, and network intrusion detection.

Componets.

- Packet decoder: it takes packets from different network interfaces and prepares them for the detection engine
- Preprocessors: used to normalize protocols headers, detect anomalies, packet reassembling and TCP stream reassembly
- Detection engine: it employs Snort rules to detect intrusion activities.
- Logging and alerting system
- Output modules: process alerts and logs and generate final output

Example: simple rule to prevent shellcodes

```
alert TCP any any -> any any (msg:"Payload detected"; content:"/etc/passwd"; sid:500001;)
```

4.1.5 Bro

Bro is an open-source, Unix-based Network Intrusion Detection System (NIDS) that passively monitors network traffic and looks for suspicious activity.
First Bro filters the traffic, discarding elements of minimal importance to its analysis. The remaining information is sent to its "event" engine, where Bro interprets the structure of the network packets and abstracts them into higher-level events describing the activity. Finally, Bro executes policy scripts against the stream of events, looking for activity that the rules indicate should generate alerts or actions, such as possible intrusions.

Componets.

- libpcap: used to capture packets from the network interfaces
- Event engine: assembles the packets coming from libpcap
- Policy script interpreter: it takes the high-level events generated by the Event Engine and compares these with the policy scripts in the system

Comparison

Parameter	Bro	Snort
Signatures	yes	no
Flexible customization	high	medium
High speed network capability	high	medium
Large user community	no	yes
Configuration GUI	no	yes
Analysis GUI	a few	a lot
Installation/Deployment	difficult	easy
OS compatibility	Unix	Any

Chapter 5

Privacy

Two types of attacker:
Outright malicious attacker
Goal: reading/modifying communication
Example: in this contest the attacker is typically called man in the middle or man in the browser.

Honest but curious attacker
Goal: use client's information after correctly handling a service
Example: database profiles the user

5.1 Content security

Cookies

- Name
- Content
- Host (name of the server - same origin policy)
- Path (server path onto which the cookie is valid)
- Expires

Different cookies by *attributes*

- Temporary (typically deleted at the end of session)
  Expires: NULL
- Persistent Expires: some date
- Secure
  Sent just over secure channel (SSL/TLS)

Different cookies by *setting*

- Third parties
  Set by domains other than the one requested. They can be used to track user.

- Supercookies
  They are associated to first level domain name. The super cookie is sent to anybody who has eg. ".com".
  They are Permanent, can store more info (< 100 kb as opposed to < 4 kb of standard cookies. They are saved also in private browsing mode.
  Example in the wild: the ISP Verizon assigned a unique cookie to each of its users. Therefore the user is completely tracked as far as that single TLD.

Private browsing
It does not prevent user tracking or identification. It only disassociates past browsing history from future.

Browser extension
Basically third party code that is executed by the browser.
Plugins do not have to respect any of the browsing policy.

Browser fingerprinting It is a technique that can uniquely identify a browser over a set of rather stable metrics:

- User agent
- Header HTTP
- Screen resolution
- Plugins/fonts
- Supercookie settings

The email client loads html so it is subject to web-based attacks!

5.2 Channel security

5.2.1 Certification Authority

CAs act as a third party that certifies the tuple (identity,public key).

- They verify the subject's identity
- Create a digital certificate with associated (identity,public key)
- Sign association with their private key.

Browsers are shipped with a list of public keys of several CAs.

HTTPS limitations

- Channel encryption is not fully end-to-end: the server can forward the traffic to third parties.
- HTTPS can be downgraded to HTTP → sslstrip
- Routing still happens over the TCP/IP stack: traffic sniffing can reveal source of traffic, domain to which request is directed, timings

5.3 VPN services

What can you do if you can not trust a VPN server or if it is blocked by the ISP? Onion routing puts multiple layers of encryption (as in an onion) around the protocol. Layers are removed as subsequent hops → no hop can know both whom sent the packet and what it is in the packet.

5.3.1 TOR

TOR is a virtual distributed network that allows the user to achieve high privacy levels thanks. to onion routing.

Chapter 6

Vulnerabilities

Software bug a bug is a problem in the execution of the software that leads to unexpected behaviour (eg. crashes,infinite loops, wrong entries of db displayed)
Characteristics of a bug

- Replicability
- Logic/configuration/design/implementation
- Fix priority
- If it is documented it is a feature

6.1 Vulnerabilities types

- Configuration v.
  eg. SSH accepts root connections from any ip
- Insfrastructural v.
  eg. sensitive db in a network's DMZ
- Software v.
  eg. authorization mechanism can be bypassed

6.2 Vulnerability discovery

Vulnerabilities are different in nature

- Often implementation dependent
- May require deep understanding of sw module interaction
- Necessary in-depth knowledge of system design (kernel structure, memory allocation)

Discovery techniques

- Code lookups (searches for known patterns)
  either you are a developer or the software is open source

- Fuzzing (semi-automatic random input generation)
- Google hacking (outdated)
  you look for software that you already know it is vulnerable

Vulnerabilites can be found either internally or externally to a company.

6.3 Vulnerability handling

The ISO 30111 is a standard to handle vulnerabilities.
The initial investigation

- The reported problem is a security vulnerability
- The vulnerability affects a supported version of the software the vendors maintains, not third parties modules (else: exit).
- The vulnerability is exploitable with know techniques (else: exit)
- Root cause analysis
- Prioritisation (evaluate potential threat)

Resolution decision
Vendor must decide how to resolve the vulnerability.

- Configuration v. $\rightarrow$ advisory may be enough
- Code v. $\rightarrow$ patch
- Critical v. $\rightarrow$ release a mitigation before full patch

Remediation development
Every solution must be tested before being delivered. This is very expensive both for customer and vendor

Release and Post-Release

6.4 Vulnerability mitigation

DEP: Data execution protection - data in memory is marked as non executable. It defeats code execution via stack corruption. Still with DEP attacker can redirect execution of code areas in memory. Eg. write a stack frame in memory and point to libc. Most memory corruption attacks rely on the attacker being able to guess start address of stack frame/heap/other areas of memory.
ASLR: address space layout randomization - it randomizes location in memory of stack, heap and libraries.

6.5 Vulnerability patching

Problems related to patches:

- Reboot is often required
- SW functionalities may change
- Deprecated third-party libraries
- Production systems need to be up and running
- A patch needs to be tested

Remark 6. *Counting vulnerabilities ≠ Security assessment. More vulnerabilites do not translate directly into risk of attacks. CVSS measures severity NOT risk.*

6.6 Different issues

Vulnerability advisories are typically published after patching.
Security researchers expect economic return and or credit.
Issue: communication between security researcher and the vendor: tradeoff between saying little and being verbose.
Often involves development of Proof of concept exploit to show the vulnerability is exploitable.

6.7 Social Engineering

The biggest threat to the security of a company is not a computer virus, an unpatched hole in a key program or a badly installed firewall. In fact, the biggest threat could be you. What I found personally to be true was that it's easier to manipulate people rather than technology. Most of the time organisations overlook that human element
Kevin Mitnick

Social engineering identifies a set of techniques that attack weaknesses in human psychology.
Situational theory of publics: why people take action, or feel part of a collective

- Problem recognition → subject thinks the problem is relevant to them
- Active involvement → subject thinks he will suffer the consequences of the threat
- Constraint recognition → subject thinks his actions are limited by factors outside of their control

6.7.1 ELM - Elaboration Likelihood Model

ELM describes the ways humans change their attitudes or decide to perform actions they would not perform without external "stimuli".
Two routes to persuasions:

- Central route
  Stimuli are weighted by the subject and the final decision is carefully elaborated. Persuasions happens through careful elaboration of information

- Peripheral route
  Subject is convinced by under analyzing apparently relevant clues that are in reality unrelated to the subject matter. Examples of adjunct elements of the communication: likeability of subject,attractiveness,trust,etc.

Remark 7. *Social engineering differs from marketing in that attackers typically do not try to sell products. Rather social engineers must persuade victims to disclose sensitive or private network.*

6.7.2 Hacking a human

- **Reciprocation**
  Normative Commitment: subject will perform an action because it is customary or mandated by law or contract. It is based on the notion of reciprocation of benefits. When subject receives something he values he feels *cognitive dissonance* (eg. sun cream free tester example). People tend to comply because they feel gratitude for the unsolicited proposal.
- **Consistency**
  Continuance commitment: subject keeps doing a determined action even if it is clearly bad (eg. lottery, keep spending money, eventually it will work).
  Upfront costs are low wrt promised benefit.
- **Social proof**
  Affective commitment: people are influenced by the opinion of those they esteem or like. Example: pretend you are on a vacation with a friend of the victim and ask money to solve an emergency
- **Likeability**
  If you like somebody you will trust him and viceversa
  Example: actors in advertisment
- **Authority**
  Subjects fear punishment and will comply
  Example: Milgram's experiment with electric shocks
- **Scarcity**
  Subject think freedom of choice is a function of time: similarly to fear scarcity leads people to take quick potentially uninformed decisions in fear of losing an opprtunity that will disappear in time or scarce in quantity.

6.8 CVSS

Why to grade vulnerability?
Vulnerability counting can not be a measure of severity. Vulnerabilities are not the same (eg. XSS vs BoF). Clients and user should be informed too: security researcher → vendor → user. A standard is needed.

CVSS is an open framework for communicating the characteristics and severity of sw vulnerability. The goal is different users score the same vulnerability in the same way → severity assessment and different people read the same vulnerability in the same way → severity communication.

Remark 8. *CVSS v2 and v3 are different*

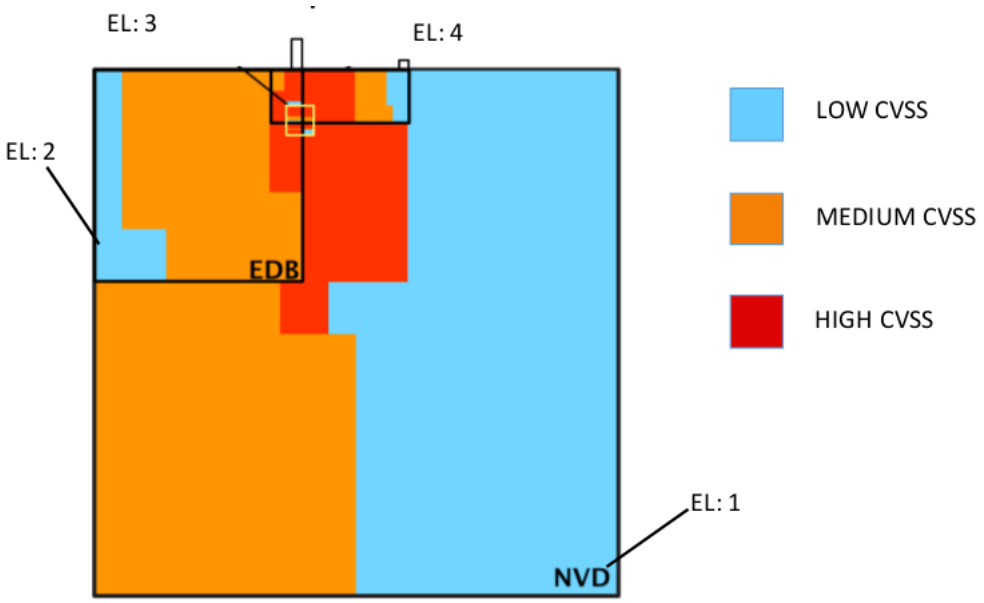
CVSS v3 has three metrics:

- **Base metric**
  Exploitability metrics: attack vector, attack complexity, privileges required, user interaction
  Scope metric: where is the vulnerability and wrt whom is affected
  Impact metrics: CIA
- **Temporal metric**
  Exploitability, remediation level,report confidence
- **Environmental metric**
  CIA requirements,mitigated base metrics

Example: think of ssh open to the wild or available just in the LAN, the environmental metrics measures this difference.

6.8.1 CVSS vs Exploitation levels

- EL1: ∃ vulnerability
- EL2: ∃ PoC
- EL3: the exploit is traded
- EL4: the vulnerability is reported and exploited
- EL5: real measure of the frequency of exploitation



Chapter 7

Malware

Def a software that run without consciousness of user and or system.
Two different kinds: program that need a host program to operate and self-independent ones.
Moreover a malware can self replicate or not.

Taxonomy

- Virus → modifies legitimate software
- Worm → self-replicates
- Trojan horse → allows remote control of machine
- Keyloggers → send typed info to attacker
- Rootkit → hook to libraries or system files
- Zombie,bot → remote coordinated control of multiple machines

7.1 Virus

Software that replicate and install itself without user consent. The copies can be installed into programs,data files,boot sector,macro files.
Components:

- Infection mechanism: enables replication
- Trigger: event that makes payload active
- Payload: (malicious or benign)

The executable changes in size so the virus is easy to spot with a hash table.
However the virus could place itself in an used PE header so there is no change in size.
Concealment mechanism

- Encrypted virus
- Polymorphic virus
- Metamorphic virus

A virus can modifiy the MBR so it is loaded in memory even before the operating system.

7.2 Rootkit

The kernel is organized in protection rings. Eg. going from inside to the outside: hardware,drivers,application,user space.
The rootkit can inject to kernel and defeat disk encryption eg. stone bootkit.
It can be persistent or memory based, in user level or kernel mode.

7.3 Counter measures

Prevention is ideal but difficult. Realistically you need: detection,identification,removal. Virus and antivirus tech have both evolved:

- Signature scanners
  Every time the malware is loaded in memory it will look the same
  It is a purely reactive strategy → unknown malware does not have a signature
- Heuristics: checks for common features
  Instead of having a perfect match it uses common features.
- Identify actions: behavioral fingerprint
- Machine learning

Evolution against heuristics → polymorphic virus
It changes how it looks in memory through encryption for every new instance.
Defense → generic decryption (aka sandboxing).
At execution time code will be the same, at that point signature and heuristics work. Still malware can recognize if it is executed in an emulated environment, if so it will not decrypt its payload.
Evolution → metamorphic virus: it rewrites the machine code in an automatic way. Fortunately this is very difficult to perform (eg. 14.000 lines of assembly code)
Defense → behavioural detection: the basic idea is that malware behaves differently from legitimate software (eg. system calls, interaction with drivers, system interrupts). The risk of false positives is higher compared to signature and heuristics (for which an hash collision is needed).

7.4 Botnets

Def. a botnet is a virtual network of infected machines under the control of a "bot herder".
It is managed through a command and control server which can push configuration files and update functionalities.

7.4.1 Centralised archtecture

Bots communicate with the bot herder via IRC or HTTP. For the latter there a re two approaches: either the bot contact a fixed (set of) IP or it resolves domain dynamically.
C& C server is a single point of failure: who controls the C&C controls the botnet.
Centralised botnets take downs happen by *sinkholing* the C&C server which needs to be protected:

Fast-flux: one domain is mapped to different IP addresses

Domain-flux: each bot generates valid domain names periodically with a DGA (domain generation algorithm) and resolves them. If bots get no answer from a generated domain they pass to the next in the list. If the DGA generates a domain which already exist then botnets may cause an unintentional DoS attack.

7.4.2 P2P architecture

It is more robust than centralised. Commands are spread through the network, bots can act as both slaves and masters. When a new machine is infected:

- Hard coded list of peer are contacted
- Mixed p2p/centralised approach (eg. a centralised web cache with list of peers)
- Infected bot inherits peer list from infector

There are three types of p2p botnets:

- **Parasite**
  All bots are selected from vulnerable hosts within an existing p2p network
- **Leeching**
  Bot candidates may be vulnerable hosts that were either inside or outside an existing p2p network.
- **Bot-only**
  It builds its own network in which all members are bots.

Usage: botnets are used to perform DDoS,send spam,exploit computational power, steal sensitive information, be rented to other criminals.

Chapter 8

Web attacks

Attacker's evolution
In the 90's attackers were security enthusiastics with high technical competence. In the 00's attackers were mainly script kiddies running automated tools while in '10s attackers are economic agents: malware is an investment.
A malware author operates in a competitive and adversarial environment:

- Adversaries: security researchers,security firms building AV signatures for malwares
- Competitors: many players in the market and a finite number of vulnerable systems.

8.1 Propagation vs operation

Different strategies:

- High propagation rate (self-replication) → higher chances to handle a sample to security researcher
- Low propagation rate (controlled deployment) → higher stealthiness

There is an equilibrium point whereby all competing players maximize their expectations in terms of return to investment.

8.2 Malware distribution network

Modern internet malware does not employ self propagation mechanism (worms are not of concern anymore). Nowadays the attacker deploys the malware on a machine he controls, the infected machines contact the malware server.
Sources of risk

- Domain compromisation
- Content compromisation (eg. XSS)
  Execution can be upon page loading, events triggering, user actions.
- Malicious third party

Drive by downloads: it is a common infection mechanism. when contacted the server delivers content that tries to exploit local vulnerabilities on the machine (eg. buffer overflow against the browser or browser plugins). If successful shellcode calls home, downloads malware and executes it.

Third party traffic: an attacker buys connections from specific users with specific configurations (eg. js check local configuration,sends it to remote server, the user loads the webpage the attacker compromised and if characteristics match traffic is redirected.

8.3 Exploit kits

Exploits typically attack vulnerabilities on adobe flash, acrobat reader, IE, Java, plugins etc.
Offensive components

- Detect browser and operating system
- Check whether the system has been already attacked by ip checking as a cookie is not so stealthy and moreover cookies can not be retrieved from new domains.
  This is a business model, it wants to keep a low profile under the radar
- Check if the system is vulnerable
- It launches appropriate attack

Defensive components

- Many exploit kits defend themselves against AV/robot detection
- Payload and malware obfuscation
- Block ip to avoid probes
- Evasion robots and crawlers
- Some even check whether the domain on which the exploit kit is hosted is included in antimalware list.

8.4 Cyber crime market

There can be many type of markets: financial,work/job position etc.
Cyber markets were originally in IRC (but being unorganized it failed completely) and then moved to web-forums. What is traded? Data, knowledge and services: eg. attacking tools, exploits, vulnerabilitites, accounts, money laundry, CCNs etc. There are two kind of market:

- TOR based market
  It is mostly on illegal goods in traditional sense (eg. drugs,weapons)
- Closed market

These markets are closed by entry selection. Some of them are typically national: eg. russian,chinese. Note that this "authentication" mechanism based on the difficulty of learning the language is not secure.
Markets can offer low-tech services (spam advertise,stolen credentials) or high tech: attack delivery technologies,malware/specialized payloads.

8.5 Composition of an attack

Elements of an attack

- User connections
- Attack delivery
- Malware infection
- Monetisation mechanism

Most products sold in the markets enable the attacker to perform only one of these steps:

- Traffic brokers,stolen accounts → connections
- Exploit kits,stand alone exploits → attack delivery
- Ransomware,banking malware → malware
- CCN,banking info → monetisation

Each of these steps can be combined by an attacker to obtain a set of characteristics that suits them

- Traffic from northern europe
- Exploit kits for recent IE versions
- Malware that does not infect CSI countries
- Attack UK bank customers

Chapter 9

System hardening

All systems have a default configuration (pc, servers, mainframes). Moreover fresh installations of an operating system offer default services (DHCP, NetBios, SSH, VNC).
System hardening is the process by which a system's configuration is tuned to improve its security without compromising its functionality.

9.1 Attack surfaces

- Weak passwords
- SW vulnerabilities
- Misconfigurations
- Services listening on the network
- Inaccurate access control

Minimality principle: no user and no system component or process should be authorised to perform actions that are not strictly necessary for their normal operations.
It can be applied to both users and processes.
A *system* should be configured such that it does not embed or enable functionalities that are not needed for normal operation (eg. microkernel)
A *user* should be authorised to only access and modify resources that are necessary for their normal operations.
Example: compile the kernel with just what is needed.

9.2 User authentication

There are two steps: *identification* and *verification*. The final goal is to link the physical user with its representation in the system. Here we list four means of authentication:

- Knows: password,PIN,graphical password
- Posses: key,token,smartcard
- Is (static biometric): fingerprint,retina
- Does (dynamic biometrics): voice,sign

They can be used alone or combined and of course they all have problems.

9.2.1 Knows

Password aging "Frequently" change passwords decreases attack surfaces (less time for attacker to crack hash file). However users will have to create and remember more passwords for one account. It is important to give users time to think of good password (eg. do not force users to change the password before they can log in). There must be a balance between security and usability.

Draw a secret scheme
It is easier to remember and there are low error rates but still adjacent coordinates are more likely to be used in sequence.

Graphical password
The user clicks on some pixels among a set of highlighted ones and then to log in the users will have to click (possibly in the same sequence). Drawbacks: shoulder surfing; is it easy for the user to change?

9.2.2 Posses

Token authentication
Examples: magnetic stripe card,memory card,smartcard etc.
Token based: it is used to compute a response to challenge. It can be used to prevent shoulder surfing (even if the attacker sees the current value he can not predict the next valid one).
The most common challenge-response mechanism is based on time syncronization: token and server have a syncronized clock so they will generate the same number at a given time.

Memory card
It stores but it does not process data. It easy to read and replicate the magnetic strip.

Smartcard
They have their own processor,memory,I/O ports. They include a crypto chip, they are tamper resistant.

9.2.3 Biometric

Physiological (static): iris,fingerprint, hand,retinal and face recognition
See picture for graph/accuracy:

- Hand → sweat palms, temperature
- Face → glasses,beard,light conditions,picture of a face,make up etc.
- Voice → cold

Behavioural (dynamic): voice,typing pattern,hand signature,gesture,gait.