1 Confidentiality: guarantees that private information is not disclosed to unauthorized people. guarantees that subjects control what information related to them can be stored and analyzed and by whom

Integrity: guarantees that resources (information, program and systems) are changed only in an authorized manner.

Confidentiality loss: an attacker read your mails by intercepting your plain SMTP traffic. You mail provider (e.g. KTH) execute statistical analyses to discover how many users discuss about some political topics.

Integrity loss: if mails are not signed, any user is able to send e-mail with a fake SMTP from header, impersonating a different user

2a $\frac{number\ of\ possible\ keys}{time\ to\ decrypt\ a\ block} = \frac{2^{64}}{2^{20}} = 2^{44}$ microseconds

2b Message Authentication Code is used to guarantee integrity. The schema assumes that two parties A and B agree on a common secret key K_{AB}. If party A wants to send a message M to B, A computes $mac = F(K_{AB}, M)$ (where F is function such that is not feasible to find a pair K'_{AB}, M' given mac), appends this code to the message and sends the results to B The other party extract the message M and the code mac from the received data, computes its own code using the message and the same key and compares the results with the code mac. If the match successes then the receiver is assured that

- the message has been generated by A
- the message has not been altered

2c If a function F is not weak collision resistant then for some x it is easy to find y such that $F(x) = F(y)$. If such a function is used to authenticate messages using encryption, the attacker can intercept a message M, its encrypted hash code $K(F(M))$ and he can forge a new message M' such that $F(M) = F(M')$. The attacker can then send the envelope $M'||K(F(M))$ that is authenticated by the receiver.

2d

- It is not one-way. Given h it easy to find an x such that $f(x) = h$: $x = h, 0, \ldots, 0$
- It is not weak collision resistant. E.g. let $h = f(x_1, \ldots, x_n)$ then for both $h, 0, \ldots, 0$ and $0, \ldots, h$ the function yields the value h.

- It is not strong collision resistant, since it is not weak collision resistant.

3

- The attacker can clearly identify users that have the same password. If the attacker is an internal user, he can discover the passwords of the other users.
- The attacker can use statistical reasoning to discover the frequency of the passwords and then associate them to the most common password used on internet. Moreover if a “rainbow table” containing for each possible salt value the hash of the most common password used on internet, and this password is part of the most common passwords in the database, the attacker can easily identify the salt value.

4 A host based intrusion detection system monitors the activities and the events occurring on a single host. Usually a host intrusion detection system is interact with the host OS to intercept the events. A network based IDS monitors network traffic that transit a particular network region, thus checking the network activities of several hosts, but being not able to check internal host activities (e.g. writing or reading on USB flash memories).

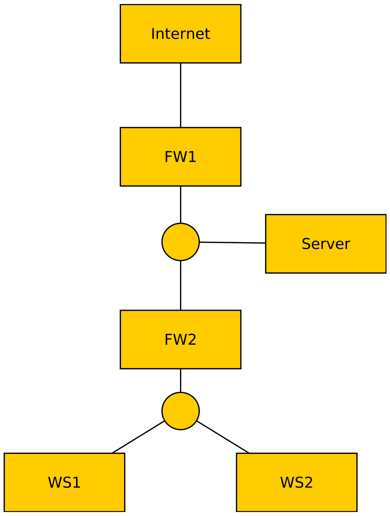
5a The user must be able to receive back traffic for the connections opened by the user itself (otherwise no TCP external service can be made available), thus the firewall must be a stateful inspection firewall to keep track of opened TCP connections.

5b We use two firewall (FW1 and FW2) to implement a DMZ. If we do not consider supporting protocols (e.g. DNS) the network configuration can be:

- FW1: packet filtering firewall: it has no state, thus reducing its overhead (all traffic cross this firewall)
  - Default discards all packets
  - Allows STMP traffic from/to the server
    - scr_ip=server_ip, dst_port=25
    - scr_ip=server_ip, src_port=25
    - dst_ip=server_ip, dst_port=25
    - dst_ip=server_ip, src_port=25

  - Allows HTTP traffic from/to the workstations
    - scr_ip=ws, dst_port=80
    - dst_ip=ws, src_port=80

- FW2: is a stateful inspection firewall, thus allowing us to track opened connections
  - Default discards all packets
  - Allows STMP traffic from/to the server
    - scr_ip=server_ip, src_port=25: established
    - dst_ip=server_ip, dst_port=25: new,established
  - Allows HTTP traffic from/to the workstations
    - scr_ip=ws, dst_port=80: new,established
    - dst_ip=ws, src_port=80: established



6a The system performs activities without the explicit agreement of the user.

- If the CD-ROM contains a malware then its code can be executed without notifying the user. The user can use an antivirus that checks all programs that are executed. The user can disable the autorun functionality. The functionality can be patched with a dialog box that asks confirmation to the user

- The CDROM vendor can be attacked and forced to spread a malware. If the vendor system is compromised, its manufacturing process can be altered to attach a malware to each produced CDROM. The manufacturer and the OS developer can implement a mechanism such that only signed program are allowed to be executed.

6b

- Viruses spread by attaching their executable code to other executable programs. To spread among different host one of the affected program must be copied by some other mechanism (e.g. the user itself)
- Worms spread by attaching vulnerability of running programs and services. They do not necessarily change the binary code saved in the storage. A Worm can spread among multiple hosts by attacking the services running on the target host.

7 IP spoofing ss the ability for any Internet host to forge fake IP source fields, thus impersonating other hosts. In a heterogeneous and federated network like Internet, this is possible because the intermediate gateways have no way to discover if the information delivered to them by other ASs is trustfulness. The only way to prevent this is to guarantee that the gateways nearest to the host (which is the gateway that assigns the IP to the host itself) prevents the connected hosts to send faked IP packets.

8 The star-property prevents a subject to write into an object of less security level. If this property is not guarantee, a subject that can read classified information can copy this information in an unclassified object (e.g. a public web page). This object can be later accessed by user with low security level, allowing the classified information to be leaked.

9 There are two buffer overflows:

- If the database passwords are longer that seven characters than `loadPwd(str1)` can cause an overflow. This problem can not be fixed if we do not know the maximum size of allowed passwords and we can not change the signature of the function to inform the corresponding implementation about the size of the target bugger `str1`.
- If the argument variable `pwd` is longer that seven character (or does not contain the null terminator) than

- the first while loop can be forced to read to high memory address, possibly raising a page fault and an availability problem
- the second while loop writes outside the memory area of `str2`

Assuming that the order of the addresses of the variables in the stack is inverse to the variable declaration (e.g. GCC) then submitting the argument `"1234567812345678\0"` allows an attacker to be always authenticated.

Patch 1:

```
>>> while (pwd[size] != 0)
<<< while (size < 8 && pwd[size] != 0)
```

This fixes the second buffer overflow, but if a user has a password `"1234567\0"` then also the argument `"7654321AAAAAA\0"` is accepted.

Patch 2:

```
>>> while (pwd[size] != 0)
<<< while (size < 8 && pwd[size] != 0)
...
>>>
<<< if (pwd[size] != 0)
<<<    return false
int i = size;
```