Chapter 1

horizontal line

# Integrity

Hashing provides assurance that data has not been corrupted, and also provides integrity by verifying that data has not changed.

Digital signatures can verify integrity of emails and files, and provide authentication and non repudiation. However they require certificates.

# Availability

Goal is to remove SPOF (single point of failure) using duplication in a system, ensuring that systems are up and running when they need to be. You can increase availability by adding fault tolerance and redundancies.

# Controls

1. **Technical controls**

Encryption, antivirus, IDS/IPS, firewalls

1. **Administrative controls**

Risk assessment, vulnerability assessment, pen testing

1. **Physical controls**

Security guards etc

1. **Preventative controls**

Hardening, Security training, and security guards, change management

1. **Detective controls**

Log monitoring, trend analysis, security audits, cctv (can also act as deterrent control)

1. **Corrective controls: attempt to reverse impact of incident**

IPS detects attack and then modify environment to stop attack. Backups and system recovery allows recovery from incidents or failure.

1. **Deterrent controls (can also be described as preventative controls)**

Cable and hardware locks deter thieves.

1. **Compensating controls**

Alternate controls used instead of primary ones, when they are not available

**Preventative vs detective controls**

Detective controls can’t predict when an attack will happen and can’t prevent it, but preventative controls prevent them from happening.

Video surveillance: Camera without recording abilities acts as a deterrent, but full CCTV acts as deterrent and also preventative control. Can also act as detective control, as security guards can review footage to detect incidents

Guards: Preventative

# 

# Virtualisation

**VMs**

Hypervisor: creates, runs and manages VM

Host: Physical system hosting VM

Guest: OSs running on host.

Host elasticity and scalability: Ability to resize capacity based on demand

If we have multiple servers running at 20% capacity, we can convert the servers to virtual hosts and run guest servers on them to upscale.

Consumes less resources e.g. electricity, cost of physical servers and increases availability

**Comparing hypervisors**

1. **Type 1 hypervisor:** run directly on system hardware, and don’t need an OS. Each VM has its own kernel.
2. **Type 2 hypervisor:** run as software within host OS

Large scale virtualisation uses **type 1** virtualisation!

**Application cell / Container virtualisation (Type 2)**

Application cell virtualisation runs services or applications within isolated application cells/containers which do not have their own kernels; they share the host kernel.

The containers don’t host an entire OS, but the host OS and kernel run the apps within the containers. Apps running in separate containers can’t interact with each other.

Uses fewer resources and is more efficient that type 2 hypervisor virtualisation. ISPs use it for customers who need specific applications.

Containers must use OS of host, e.g. if host is running linux, all containers must run linux

**Secure network architecture**

VMs can provide separation of systems, by disabling the NIC(network interface card) in the VM, preventing it from transmitting any data in or out of VM.

**Snapshots**: Provides snapshot of VM, so you can revert back to that state if anything goes wrong.

**VDI/VDE (Virtual desktop infrastructure/environment) and non persistence**: User’s desktop OS runs as VM on server.

If PC can connect to server over a network, it can run a full featured desktop OS from the server

**Persistent virtual desktop**: Each user has customisable desktop image and can save data within desktop.

Lots of disk space required to support unique desktop images for all users.

Will VDE support persistence?

When user accesses a non persistent desktop, he is provided with desktop OS from a preconfigured snapshot. Changes can be made but it reverts to known state after logoff.

**VMs as files**

VMs are groups of files, so if one server becomes overloaded you can move virtual servers to another physical server.

Restoring failed server: If you create a backup of the virtual server files and the original fails, you can restore the files, which takes minutes.

Rebuilding a physical server, on the other hand, can take hours.

You can manage all virtual servers running on different physical servers using a management tool.

**Risks of virtualisation**

VM Escape: Attacker can run code on a virtual system and access the hypervisor. This gives them access to the host and each virtual system in the host.

VM escape vulnerabilities are fixed via patches.

VM sprawl: When an organisation has many poorly managed VMs, which are not patched or updated.

Unauthorised VMs increase load and consume system resources, making server slower.

Loss of confidentiality: VMs contain lots of information, which can be copied and stolen.

**Kali linux in VM:**

1. Hyper-V allows you to run VMs on windows, and also gives a virtual switch, allowing inter VM communication
2. VM workstation player: if system doesn’t support hyper V.
3. Oracle VMVirtualBox

**Using command line tools**

**Ping**

Tests connectivity for remote systems, and test whether they can resolve domain names, test NIC.

It sends ICMP (Internet control message protocol) echo request packets, and if you receive reply packets then system is operational.

On windows: Sends 4 ICMP echo requests.

Can do this on linux by typing **ping -c 192.168.1.1.**

On linux: Keeps sending until you press ctrl c.

Can do this on windows by typing **ping -t 192.168.1.1.**

**Using ping to check name resolution**

Malware can change the name resolution process to prevent systems from accessing the windows update server and updating. You can ping host name of remote system to check that name resolution is working.

Firewalls: DoS attacks use ICMP to disrupt services, so firewalls may block ICMP traffic.

**Using ping to check security posture**

You can check if firewalls are blocking ping traffic by sending ICMP echo requests.

Admins use ping to check connectivity of remote system, verify that name resolution is working, and to check security posture of systems by verifying that IPSs and firewalls block ICMP (ping) traffic.

**Ipconfig**

Shows TCP/IP configuration information for a system, i.e/ IP address, subnet mask, MAC address, and config info for NICs. ipconfig commands are used for troubleshooting.

**Linux: ifconfig**

Can do all the above, and also configure NICs.

**ipconfig/displaydns**: displays content of DNS cache. **Flushdns** flushes dns cache, useful if there is incorrect info inside it.

**Ifconfig eth0:** shows configuration of first ethernet interface (NIC). **promisc** processes all traffic the NIC receives, regardless of IP addressed to. **Allmulti** allows NIC to process all multicast traffic received by NIC.

**Linux: ip**

Can’t enable promisc mode.

**Ip link show:** Shows interface and details

**Ip link set eth0 up:** Enables network interface

**Ip link show:** Shows stats of network interface

Windows systems use ipconfig to view network interfaces. Linux systems use ifconfig/ip, which can also manipulate settings on network interface, but ifconfig can enable promisc mode.

**Netstat**

Shows stats of TCP/IP configuration, and view active connections. If an attack has established a connection to an infected server, you can verify this with netstat.

**Netstat**: Lists all open TCP connections

**Netstat -a**: Lists all open TCP connections and UDP (user datagram ports) that a system is listening on, e.g. if you see an IP address followed by :80, it is most likely a web server.

**Netstat -r**: Lists routing table

**Netstat -e**: Displays network stats

**Netstat -s**: Displays stats of packets received, e.g. tcp/ip/icmp

**Netstat -n**: Lists addresses in numerical order

**Netstat -p *protocol***: Shows stats for specific *protocol*

We can also display the state of connections…

**ESTABLISHED**: Active open connection

**LISTEN**: System is waiting for connection

**CLOSE\_WAIT**: System waiting for connection termination request

**TIME\_WAIT:** Waiting for enough time to pass to be sure that remote system received TCP acknowledgement.

**SYN\_SENT:** System sent TCP SYN packet to start the SYN - SYN/ACK - ACK handshake process.

**SYN\_RECEIVED:** System sent SYN/ACK packet after receiving SYN, and is waiting for ACK to establish connection. Excessive no of SYN\_RECEIVED states indicates SYN attack where attacker is flooding with SYN packets but never responding with ACK to finalise connection!

**Traceroute**

Lists routers between 2 connections, their IP addresses and RTTS. Windows = tracert and linux = traceroute.

Can be used to identify faulty routers on a network if ping fails, useful for troubleshooting on a WAN. Can also be used to identify modified paths, e.g. if an attacker inserted an unauthorized router between systems.

**Arp**

Related to address resolution protocol (**ARP**). ARP resolves IP addresses to MAC addresses and stores it in a cache. Arp command can identify a server’s ip address and then arp -a can show the cache which includes the MAC address for that IP address.

**Arp:** Shows help on windows

**Arp:** Shows ARP cache on linux

**Arp -a:** Shows ARP cache on windows

**Arp -a [ip address]:** Shows ARP cache for specified IP address

Can be used to check for an ARP cache poisoning attack!