☑ PREQUISITRY — Samenvatting

OSI Model

“On what layer is a MAC Address?”

Een **MAC Address** bevindt zich op **Layer 2 – Data Link** van het **OSI Model**. Dit is het niveau voor fysieke adressering binnen hetzelfde netwerksegment.

“On what layer is an IP Address?”

Een **IP Address** zit op **Layer 3 – Network**, waar **routing** en **logical addressing** plaatsvinden.

“On what layer is ARP?”

ARP (address resolution protocol) werkt tussen Layer 2 en Layer 3 — maar wordt officieel geplaatst op **Layer 2**, omdat het MAC adressen opzoekt.

“On what layer is TCP?”

TCP (transmission control protocol) zit op **Layer 4 – Transport**.

“On what layer is UDP?”

UDP (user datagram protocol) zit eveneens op **Layer 4 – Transport**.

“On what layer is HTTP?”

HTTP (hypertext transfer protocol) zit op **Layer 7 – Application**, waar applicaties met het netwerk communiceren.

“On what layer is ICMP?”

ICMP (internet control message protocol) zit op **Layer 3 – Network**, want het wordt gebruikt voor routering en diagnose.

Routing and Switching

“Should the default gateway be part of your network/subnet?”

Ja. Een **default gateway** moet zich **binnen dezelfde subnet range** bevinden als het device, anders kan je toestel er niet naartoe ARP’en.

“How do you see the default gateway on a Windows/Linux machine?”

Windows (CLI):

```
ipconfig
route print
```

Linux (CLI):

```
ip route
ip r
route -n
```

“Should the DNS server be part of your own network/subnet?”

Niet noodzakelijk. Een **DNS Server** kan *intern* of *extern* zijn (bijv. 1.1.1.1 of 8.8.8.8). Je hoeft er niet rechtstreeks naartoe te ARP’en, want DNS requests gaan via de default gateway.

“How do you retrieve DNS configuration on Windows/Linux?”

Windows:

```
ipconfig /all
```

Linux:

```
cat /etc/resolv.conf
systemd-resolve --status
nmcli dev show
```

Subnetting

“You should be able to determine subnet ranges and see if devices belong to the same subnet.”

Je moet kunnen:

- het **network address** bepalen
- het **broadcast address** bepalen
- het **aantal hosts** berekenen
- bepalen of twee IP's in dezelfde **subnet range** zitten via **bitmask** of **CIDR** (bijv. /24)

VirtualBox Network Types

“Know the types: NAT, NAT Network, Bridged, Host-Only, Internal”

- **NAT** → VM heeft internet, maar is niet bereikbaar vanaf het LAN.
- **NAT Network** → NAT + onderlinge communicatie tussen VM’s.
- **Bridged** → VM zit in het echte LAN, als een volwaardige host.
- **Host-Only** → VM kan enkel met de host praten.
- **Internal Network** → Enkel VM’s onderling; host heeft geen toegang.

SYSTEM ADMINISTRATION

Basic scripting (PowerShell & Bash)

“Create, manage and run basic scripts”

Je moet eenvoudige scripts kunnen uitvoeren zoals loops, variabelen, en simpele taken automatiseren.

Systemd (Linux)

“Managing systemd services using systemctl”

Belangrijkste commando’s:

- `systemctl start service`
- `systemctl stop service`
- `systemctl status service`
- `systemctl enable service`
- `systemctl cat service` (config bekijken)

Managing files on Linux

“Editing: nano, vi(m)”

Je moet basisbewerkingen kennen (openen, opslaan, sluiten).

“Copying files: scp, sftp, wget, curl”

- **scp** en **sftp** → secure copy over SSH
- **wget** en **curl** → downloaden via HTTP/HTTPS

“Ownership and rights”

Belangrijk:

- `chmod` → permissions aanpassen
- `chown` → eigenaar aanpassen
- `ls -l` → rechten bekijken

Remote management using SSH

"How to SSH into a machine?"

```
ssh user@ip
```

"What is the authorized_keys file?"

Bevat **public keys** van clients die mogen inloggen zonder wachtwoord.

"What is the known_hosts file?"

Bevat **fingerprints** van servers waarmee je ooit verbinding hebt gemaakt (man-in-the-middle bescherming).

"Where are the public and private keys stored?"

Standaard:

```
~/.ssh/id_rsa        → private key
~/.ssh/id_rsa.pub    → public key
```

THEORETICAL CONCEPTS

"What is the difference between symmetric and asymmetric encryption?"

- **Symmetric encryption** → 1 key voor encryptie en decryptie (snel).
- **Asymmetric encryption** → public key + private key (veiliger voor communicatie).

"How do private and public keys work?"

- **Public key** → mag je delen
- **Private key** → blijft geheim en beschermd
- Encryptie: public key gebruikt
- Decryptie: private key gebruikt

“What is the CIA triangle?”

- **Confidentiality** → gegevens enkel voor wie toegang heeft
- **Integrity** → gegevens mogen niet ongezien wijzigen
- **Availability** → systemen en data moeten beschikbaar zijn

---

“Understand NAT”

NAT (network address translation) herschrijft IP adressen. Wordt gebruikt om private IP’s met het internet te verbinden.

“Understand the concept of a firewall”

Een **firewall** filtert netwerkverkeer op basis van regels (ports, IP’s, protocols) en beslist wat door mag.

▧ LES 1 — Samenvatting

1. “What is the Swiss cheese model and how can it be applied to cybersecurity?”

Het **Swiss Cheese Model** toont hoe meerdere beveiligingslagen elkaar aanvullen. Elke laag (firewall, antivirus, logging, MFA, netwerksegmentatie…) heeft “gaten”: zwaktes.

Cybersecurity gebruikt meerdere lagen zodat de gaten nooit perfect uitlijnen → zo voorkom je dat één fout tot een incident leidt.

---

2. “What different types of network attacks exist?”

“What is a (D)DoS attack?”

Een **DoS** of **DDoS (distributed denial of service)** overstroomt een target met zoveel verkeer dat het niet meer kan antwoorden. Doel: **beschikbaarheid (availability)** breken.

“How can DNS be considered an attack vector as well?”

DNS kan misbruikt worden voor:

- **DNS Amplification** (DDoS via grote antwoorden)
- **DNS Spoofing / Poisoning** (foute IP's teruggeven)
- **Data exfiltration** via DNS tunnels
- **Misconfiguraties** zoals open resolvers

---

3. DNS

“What information can be enumerated from a DNS server?”

Met DNS enumeration kun je opvragen:

- **A, AAAA records** (IP’s)
- **MX records** (mail servers)
- **NS records** (authoritative servers)
- **TXT/SOA/SPF records**
- Mogelijke **subdomains**

“When is it intended? What is a normal DNS resolve?”

Normaal gebruik:

```
nslookup example.com
dig example.com
```

DNS hoort alleen antwoorden op publieke records. Details zoals interne zones horen *niet* publiek te zijn.

“What is, and how can you perform, a reverse lookup?”

Reverse lookup zoekt een **hostname op basis van een IP**:

```
dig -x 8.8.8.8
```

“What is meant by authoritative nameservers?”

Een **authoritative nameserver** bevat de **officiële** zone-informatie voor een domein (records die de domain owner beheert).

“What is a zone transfer attack and why is it called an attack?”

Een **zone transfer (AXFR)** stuurt *de volledige DNS zone* naar de vrager.

- **Normaal:** alleen tussen primaire en secundaire DNS servers.
- **Aanval:** wanneer een externe attacker een volledige lijst van hosts/subdomeinen kan downloaden door slechte configuratie.

Niet altijd schadelijk → maar wél een *ernstige informatielek*.

---

4. tcpdump (or alternatives)

“How can you create a network dump using the CLI?”

```
tcpdump -i eth0 -w capture.pcap
```

“How can you exclude SSH traffic?”

```
tcpdump -i eth0 not port 22
```

“How can you only include HTTP traffic?”

```
tcpdump -i eth0 port 80
```

---

5. Wireshark

“What can an analyst learn from the Conversations window?”

- Wie communiceert met wie
- Aantal pakketten
- Data size
- Direction

“What can an analyst learn from Statistics?”

- Top talkers
- RTT
- Packet lengths
- Protocol usage

“What can an analyst learn from Protocol Hierarchy?”

Overzicht van alle gebruikte **protocols** en hun percentage in de capture.

LES 2 — Samenvatting

1. “In terms of the OSI model, what does a firewall do?”

Een traditionele firewall werkt op:

- **Layer 3 (Network)**: filtering op IP
- **Layer 4 (Transport)**: filtering op TCP/UDP ports

Next-gen firewalls kunnen ook Layer 7 inspectie doen.

---

2. Review the advantages and disadvantages of a host-based firewall vs a network-based firewall

Host-based firewall

Voordelen:

- Beschermt individuele hosts
- Fijngranulaire controle

Nadelen:

- Moeilijk te beheren op veel toestellen
- Kan door malware uitgeschakeld worden

Network-based firewall

Voordelen:

- Eén centraal controlepunt
- Goed voor segmentatie

Nadelen:

- Ziet geen interne hostprocessen
- Meer complex bij grote netwerken

---

3. “What is meant by network segmentation? What are network/firewall zones?”

Network segmentation Het splitsen van een netwerk in kleinere delen zodat verkeer en toegang gescheiden worden. Dit vermindert risico’s, beperkt aanvallen en verhoogt controle.

Network/Firewall zones: Groepen van netwerksegmenten met een eigen beveiligingsniveau (bv. DMZ, intern, secure zone). Firewalls bepalen welke zones met elkaar mogen communiceren.

Voordeel: een breach blijft beperkt.

---

4. DMZ

“What is a DMZ? How can you build this with 1 firewall vs 2 firewalls? Are there advantages?”

Een **DMZ (demilitarized zone)** is een bufferzone tussen internet en intern netwerk, voor publieke services zoals:

- Webservers

- Reverse proxies
- Mail gateways

1 firewall:

- Eén toestel heeft 3 interfaces: WAN, DMZ, LAN
- Goedkoper maar single point of failure

2 firewalls:

- WAN → firewall 1 → DMZ → firewall 2 → LAN
- Betere beveiliging, maar duurder en complexer

Nmap

"When scanning a TCP or UDP port with nmap, what is the difference between open, filtered and closed? How is nmap able to make this conclusion?"

- **Open** → Port luistert actief
- **Closed** → Host antwoordt maar geen service
- **Filtered** → Firewall blokkeert/geen antwoord Nmap beslist dit op basis van **TCP flags**, **ICMP antwoorden** of *geen* antwoord.

"What is a banner grab and how do you do this with nmap?"

Banner grabbing = service-informatie uitlezen:

```
nmap -sV target
nmap --script=banner target
```

Systemd

"Where can you find systemd configuration files?"

- `/etc/systemd/system` → custom overrides
- `/usr/lib/systemd/system` → standaard unit files
- `/run/systemd/system` → runtime units

Belangrijke systemctl commando's

- `systemctl cat service` → config tonen
- `systemctl show service` → properties tonen
- `systemctl edit service` → override file maken
- `systemctl list-units` → actieve units
- `systemctl list-unit-files` → alle units

- `systemctl daemon-reload` → herladen config

“What are systemd timers?”

Alternatief voor cron jobs. Je kunt taken plannen via `.timer` files.

Proxy

“What is a forward proxy?”

Client → proxy → internet Wordt gebruikt voor:

- filtering
- caching
- anonimiteit

“What is a reverse proxy?”

Client → **reverse proxy** → interne servers

Voordelen:

- Load balancing
- TLS termination
- Caching
- Verbergen van interne structuur

“How are (reverse) proxies related to load balancers? Do you know software that can do both?”

Reverse proxies *kunnen* ook load balancers zijn.

Voorbeelden:

- **NGINX**
- **HAProxy**
- **Traefik**

▨ LES 3 — Samenvatting

Essential SSH configuration

“Logging in remotely using keys instead of username/password”

Je genereert keypair:

```
ssh-keygen
```

Public key → `authorized_keys` Private key → blijft lokaal.

"authorized_keys file?"

Bevat public keys van gebruikers die mogen inloggen zonder password.

"known_hosts file?"

Bevat fingerprints van servers waar je eerder verbinding mee maakte → MITM bescherming.

"SSH config for client and server?"

Client: `~/.ssh/config` Voorbeeld:

```
Host web
    HostName 192.168.1.10
    User admin
    IdentityFile ~/.ssh/id_rsa
```

Nu kun je gewoon `ssh web`.

Server: `/etc/ssh/sshd_config` Belangrijke opties:

- `PasswordAuthentication no`
- `PubkeyAuthentication yes`
- `Port 22` (of custom)

"The difference between a passphrase on a key vs user/password"

- **Passphrase** beveiligt je *private key*
- **User/password** is een loginmethode zonder key → Passphrase = tweede laag voor je sleutel.

"What is meant by a Jump/Bastion host and why would a company use this?"

Een **bastion host** is een tussenstation om interne servers te bereiken. Doel: **toegang centraliseren en beveiligen**.

Een bastion host is een streng beveiligde server die als enige toegestane toegangspoort naar een intern netwerk dient. Bedrijven gebruiken het om:

- Beveiliging te verhogen
- Toegang centraal te beheren

- Logging/auditing te garanderen
- Schade bij hacks te beperken
- Zero Trust en netwerksegmentatie te ondersteunen

“What is the difference between local and remote port forwarding using SSH? You should be able to explain the difference using a proper use case.”

Local port forwarding → Je opent een lokale port die doorstuurt naar een remote service. Voorbeeld: interne database toegankelijk maken via localhost.

Remote port forwarding → Server opent een port naar jouw machine. Gebruikt voor: toegang tot jouw lokale service vanop een remote server.

“What is the SOCKS protocol conceptually and give an example where or when this might be interesting. ”

SOCKS is een **generic proxy protocol** via SSH:

```
ssh -D 1080 user@server
```

Je creëert een **SOCKS proxy** → volledige browsertraffic door een SSH-tunnel. Interessant voor:

- beveiligde verbindingen op publiek wifi
- geavanceerde pivoting in pentesting

▨ LES 4 — Samenvatting

1. “What is a honeypot?”

Een **honeypot** is een systeem dat opzettelijk kwetsbaar lijkt om aanvallers te lokken, te monitoren en te analyseren.

---

2. “What type of honeypots exist? Make a distinction between function, use and what they try to achieve?”

Op basis van functie

- **Low-interaction honeypots** → simuleren services
- **High-interaction honeypots** → echte OS + echte services

Op basis van doel

- **Research honeypot** → gedrag van aanvallers bestuderen
- **Production honeypot** → aanvallers afleiden van echte systemen

---

3. “How do honeypots differ from honey/canary-tokens?”

- **Honeypot** → volledig systeem
- **Honeytoken** → stukje data dat nooit gebruikt hoort te worden Voorbeeld: valse API key, bestandje “passwords.xlsx”.

Wanneer het gebruikt wordt → **alert**.

---

4. “What is a canary/honeytoken?”

Data die enkel bestaat om misbruik te detecteren.

---

5. “Review Docker in terms of security”

“Is Docker virtualisation (type 1 or type 2), emulation, simulation?”

Docker is **OS-level virtualization** (containerization). Geen type 1 of type 2 hypervisor → deelt kernel met de host.

“What are some security implications when using Docker? Is it considered to be more secure compared to virtual machines? Why (not)?”

- Containers delen kernel → minder isolatie dan VM’s
- Privileged containers zijn gevaarlijk
- Access tot Docker socket = **root access** op host

Nee → minder isolatie. VM’s hebben **eigen kernel**, containers niet.

“Would you deploy a honeypot in Docker in production?”

Niet ideaal. Als iemand de container breekt, breekt hij mogelijk de host mee.

“If a Docker container requires access to the Docker socket from within the container, what security implications does this have?”**

Toegang tot de **Docker socket (`/var/run/docker.sock`)** geeft een container **volledige root-controle over de host**.

Waarom?

- De Docker daemon draait als **root**.
- De socket biedt toegang tot de volledige **Docker API**.
- Alles wat via die API gebeurt → wordt uitgevoerd als **root op de host**.

Risico’s:

- Container escape → volledige toegang tot host filesystem
- Containers starten/stoppen/modificeren
- Privileged containers starten
- SELinux/AppArmor/cgroup-beperkingen omzeilen
- Malware persistent maken
- Laterale beweging in het netwerk

Kortste samenvatting:

> **Access tot de Docker socket = root access op de host → zeer gevaarlijk in productie.**

LES 5 — Backups (Theory)

“What different 'rules' exist when talking about a backup strategy? Explain what you need to take into consideration.”

Belangrijke “rules” / principes:

- **3-2-1 rule**: 3 copies, op 2 verschillende media, 1 off-site.
- **RPO (Recovery Point Objective)**: hoeveel dataverlies is acceptabel (hoe vaak backup maken).
- **RTO (Recovery Time Objective)**: hoe snel moet herstel gebeuren.
- **Regular testing**: backups moeten regelmatig getest worden (restore-tests).
- **Versioning / retention**: bewaar meerdere versies / retentiebeleid (retention policy).
- **Encryption**: backups moeten encrypted zijn, zowel in transit als at rest.
- **Access control**: beperk wie backups kan lezen/wissen (least privilege).
- **Automation & monitoring**: automatische jobs + alerts bij fouten.
- **Immutable backups / WORM**: bescherming tegen ransomware (cannot be altered).
- **Separation of duties**: opsplitsen van rollen voor security/compliance.

Kort: bepaal RPO/RTO, volg 3-2-1, versleutel, test regelmatig en bescherm tegen ransomware.

“What is the difference between a full vs an incremental backup? Give for both advantages and downsides.”

Full backup

- **Wat:** kopieert alle data.
- **Voordelen:** eenvoudig te herstellen; één file-set bevat alles.
- **Nadelen:** veel opslag nodig; lange duur; hoge netwerkbelasting.

Incremental backup

- **Wat:** kopieert alleen veranderingen sinds de laatste backup (full of incremental).
- **Voordelen:** sneller, bespaart opslag en bandbreedte.
- **Nadelen:** herstel is complexer (je hebt base full + alle increments nodig); hogere kans dat één corrupte increment herstel breekt.

Alternatief: Differential backup (wijst naar veranderingen sinds laatste full): sneller dan full, herstelt eenvoudiger dan incremental maar groeit in grootte na verloop van tijd.

“Why do some people state that 'synchronisation with a cloud service (OneDrive, Dropbox, Google Drive) is not a synonym for backups'?”

Synchronisatie is **two-way** en reflecteert wijzigingen onmiddellijk:

- Als je per ongeluk een bestand delete of corrupt maakt, synchroniseert de cloud die wijziging terug naar alle synced clients.
- Synchronisatie focust op **availability/convenience**, niet op versiebeheer of immutability.
- Backups bieden **versioning**, **retentiebeleid**, **immutable copies** en vaak **off-site/air-gapped** kopieën — synchronisatie alleen doet dat niet standaard.

“Why is putting 100% trust on a cloud provider a (potential) bad idea?”

Risico’s van enkel op één cloud provider vertrouwen:

- **Provider outage**: cloud kan tijdelijk offline zijn (availability risico).
- **Account compromise**: als credentials of admin-account worden gehackt, ben je alles kwijt.
- **Accidental deletion / ransomware**: single point of failure als geen extra backups bestaan.
- **Vendor lock-in & data portability**: migratie kan moeilijk/duur zijn.
- **Legal / compliance & jurisdiction**: geopolitieke of juridische beperkingen op data.
- **Provider bug / policy change**: onvoorziene wijzigingen of fouten.

Best practice: gebruik meerdere lagen (on-prem + cloud + off-site), encryptie en immutable/air-gapped kopieën.

LES 6 — PKI & TLS (Theory + Lab thought)

“If your browser warns about an expired certificate and you accept the risk and continue — is traffic still encrypted?”

Ja — het verkeer blijft **encrypted** met TLS. Maar **de authenticity/integrity/trust** van de cert wordt niet gewaarborgd: je weet niet of je echt met de verwachte server praat (man-in-the-middle risico hoger). Accepting the risk schakelt alleen de **validity checks** over; de cryptografische channel blijft actief.

“What is X.509?”

X.509 is de **standard** voor **public key certificates** (format for certificates, fields like Subject, Issuer, serial, validity, public key, extensions). X.509 certs worden gebruikt in TLS/SSL, S/MIME, etc.

“What is meant by CSR in this context?”

CSR (Certificate Signing Request) is een verzoekbestand dat een gebruiker/server genereert (bevat public key + subject informatie) en dat naar een CA gestuurd wordt om een cert te laten signen.

“What is SAN, Subject Alternative Name?”

SAN is een X.509 extension die meerdere DNS-names (en IPs, URIs) aangeeft waarvoor het cert geldig is (bv. example.com, www.example.com, api.example.com).

“What are Certificate chains and cross-certifications?”

- **Certificate chain**: pad van server cert → intermediate CA(s) → root CA. Trust is gebaseerd op root CA die in trust store staat.
- **Cross-certification**: twee CAs ondertekenen elkaars certs zodat vertrouwen kan worden overgedragen tussen PKI-hiërarchieën (brugtrust).

“How does a CA certificate renewal work using cross-certification?”

Kort: bij rollover wil je dat nieuwe root/CA vertrouwd wordt vóór de oude offline gaat. Cross-certification creëert een pad zodat klanten die de oude root vertrouwen een vertrouwenpad kunnen volgen naar de nieuwe CA via een cross-signed intermediate. Hierdoor is er overlap in trust tijdens de overgang. (Zie Wikipedia CA rollover diagram voor visueel pad: oude root signeert new CA intermediate en vice versa om overgang te verzekeren.)

“Difference between SSL and TLS? Current standard? Which version?”

- **SSL** = verouderde predecessor. Term wordt vaak nog informeel gebruikt.
- **TLS** = huidige secure protocol.
- Huidige standaard (2025): **TLS 1.3** is de aanbevolen versie; TLS 1.2 veel gebruikt maar 1.3 is modern standaard.

“Is MD5 still used in the most recent version?”

Nee. **MD5** is cryptografisch gebroken en wordt niet gebruikt voor signatures in modern TLS. Moderne ciphersuites gebruiken SHA-2 of SHA-3 families; TLS 1.3 gebruikt AEAD ciphers en moderne hashes.

“What is Let's Encrypt?”

Let’s Encrypt is een gratis CA die **automated** certificate issuance via het **ACME** protocol aanbiedt. Doel: TLS breed toegankelijk en eenvoudig maken. Certs zijn meestal korte levensduur (90 dagen) en ontworpen voor automatisatie.

“Isn't it bad that people (including hackers) can create webserver certificates for free?”

Niet per se: het voordeel is brede encryptie en lagere barrière. Redenen waarom dit acceptabel is:

- CA’s voeren **domain validation (DV)** uit, dus alleen de eigenaar van een domain kan een cert krijgen (of degene die DNS/http-challenge kan uitvoeren).
- Voor **authentication** zijn EV certs of additional vetting nodig; gratis DV-certificates bieden **encryption** maar geen sterke identity assurance.
- Misbruik is mogelijk (phishing met valid certs), maar overall encryptie verbetert security. Monitoring, CT logs, and revocation help beperken misbruik.

“How can you decrypt HTTPS traffic for TLS 1.2 configured webservers?”

Mogelijkheden (legitiem, enterprise-context):

- **Server private key**: als server gebruikte RSA key exchange (non-ephemeral) dan kun je met private key en captured traffic decrypten. Maar moderne configs gebruiken ephemeral DH so this often fails.
- **SSLKEYLOGFILE**: clients (browsers) kunnen session secrets loggen (pre-master secrets) en Wireshark kan die gebruiken om decryptie te doen.
- **TLS interception (proxy/MITM) with enterprise CA**: company installs its own CA on clients; proxy terminates TLS and re-encrypts to server (visible plaintext at proxy).
- **TLS session keys from endpoint** (endpoint forensics).

“Why isn't it possible to decrypt HTTPS traffic for TLS 1.3 the same way as for TLS 1.2?”

TLS 1.3 **mandates ephemeral key exchanges** (Ephemeral Diffie-Hellman) giving **Perfect Forward Secrecy (PFS)**. Server private key alone **is not sufficient** to reconstruct session keys for passive captures. So you cannot decrypt passive captures with only server private key.

“Is it possible for a company to still decrypt everything including TLS 1.3? If yes, how?”

Yes, but only via **active interception** under company control:

- **TLS interception proxy (man-in-the-middle)**: company issues a trusted enterprise CA certificate and installs it in clients’ trust stores. The proxy terminates client TLS and establishes its own TLS to remote server — proxy sees plaintext.
- **Endpoint key logging**: configure clients to export session secrets (SSLKEYLOGFILE) to central collector.
- **Agent on endpoints**: endpoint agents can capture decrypted traffic before encryption.

Opmerking: dit vereist beheer over endpoints (install CA or agents) en heeft privacy / legal implications.

“What is a SIEM? The acronym. Give some examples.”

SIEM = Security Information and Event Management. Functie: centraliseren, correleren en analyseren van logs/events voor detectie, alerting en forensics.

Voorbeelden: Splunk, ELK stack (Elasticsearch + Logstash + Kibana) + Beats, Graylog, IBM QRadar, Azure Sentinel, Wazuh (agent + ELK integratie).

“What is a SOC?”

SOC = Security Operations Center. Team/organisatie-eenheid die verantwoordelijk is voor monitoring, detectie en response op security incidents.

“What is meant by compliancy in terms of cybersecurity?”

Compliancy = voldoen aan wettelijke, industriële of organisatorische regels/standaarden (bv. GDPR, PCI-DSS, ISO27001). Het gaat om beleid, controles, rapportage en bewijsvoering.

“What features does Wazuh offer?”

Wazuh is een open-source security platform:

- Host-based monitoring & EDR-like features
- **FIM (File Integrity Monitoring)**
- Log collection & analysis
- Intrusion detection rules (via integrated rules)
- Vulnerability detection
- Configuration assessment / compliance checks
- Integration with ELK/Kibana

“What is FIM?”

FIM = File Integrity Monitoring. Detecteert wijzigingen aan belangrijke bestanden (hashes, permissions, owners). Wordt gebruikt voor detection and forensic evidence.

“What is Sysmon?”

Sysmon (System Monitor) is een Windows system service (Microsoft Sysinternals) die gedetailleerde events logt: process creation, network connections, driver loads, hashes. Handig voor endpoint telemetry and threat hunting.

“What is the difference between an IDS and an IPS?”

- **IDS (Intrusion Detection System)**: passief detecteert/analyseert en **alarmeert** (out-of-band).

- **IPS (Intrusion Prevention System)**: inline en **kan verkeer blokkeren** of aanpassen (in-band).

“Fundamental differences between a firewall and an IDS/IPS?”

- **Firewall**: policy-based filtering (IP, ports, basic L4/L3 rules) — main goal = access control.
- **IDS/IPS**: deep inspection, pattern/signature or anomaly detection, focus op attack detection and prevention. Conceptueel: firewall regelt “who/what can talk”, IDS/IPS inspects “what is being said / is malicious”.

“Create, understand and interpret some basic Suricata rules”

Detect specific protocol (e.g., HTTP method):

```
alert http any any -> any any (msg:"HTTP GET detected"; http.method;
content:"GET"; sid:1000001; rev:1;)
```

Detect traffic using a specific port (e.g., SSH port 22):

```
alert tcp any any -> any 22 (msg:"SSH connection detected"; sid:1000002; rev:1;)
```

(Uitleg: `alert` = actie; `proto srcIP srcPort -> dstIP dstPort`; `msg` = beschrijving; `sid` = signature id.)

“Placement of a physical IDS/IPS if separate from firewall?”

- **IDS (passive):** plaats het op een SPAN/mirror port of netwerk-tap om copy van verkeer te analyseren (geen impact op traffic).
- **IPS (inline):** plaats het tussen firewall en netwerksegment dat je wilt beschermen (inline) — dit introduceert latency en single point of failure (use high-availability).

Impact: IPS kan false positives veroorzaken (legitiem verkeer blokkeren). Placing requires planning for latency, redundancy (HA) and fail-open/closed policies.

“What is Security Onion? What is the goal?”

Security Onion is een open-source Linux distribution bundel voor network security monitoring (NSM). Het bevat tools zoals Suricata, Zeek (Bro), Wazuh, Elastic Stack, Sguil en meer. Doel: gemakkelijk deployment van full monitoring stack voor detection, analysis, and incident response.

LES 8 — IPsec (Theory)

“What is the fundamental goal of IPsec?”

Beschermen van IP-verkeer op netwerklaag: confdentiality, integrity en authenticity (VPN tunnel of host-to-host).

“How does IPsec work? What are SP's and SA's?”

- **IPsec** bouwt beveiligde tunnels via **Security Associations (SA)** die cryptografische parameters (keys, algorithms, lifetimes) definiëren.
- **SP = Security Policy**: regels die bepalen welk verkeer beveiligd moet zijn (policy database).
- SA’s worden opgeslagen in de **Security Association Database (SAD)**.

“What modes exist in IPsec? What is the difference?”

- **Transport mode**: alleen payload van IP-pakket wordt beschermd; originele IP header blijft intact. Gebruikt voor host-to-host.
- **Tunnel mode**: het volledige IP-pakket wordt ingekapseld en een nieuwe IP header wordt toegevoegd — gebruikt voor gateway-to-gateway VPNs.

“What is AH?”

AH (Authentication Header) biedt **integrity** en **authentication** van IP-pakketten maar **geen encryptie** (geen confidentiality) en kan problemen hebben met NAT omdat header velden gewijzigd worden.

“What is ESP?”

ESP (Encapsulating Security Payload) biedt **confidentiality (en optioneel integrity/authentication)**. Vaak gebruikt in tunnel mode voor VPNs.

“What is IKE?”

IKE (Internet Key Exchange) is het protocol dat SA’s onderhandelt en keys uitwisselt (IKEv1, IKEv2). IKE gebruikt Diffie-Hellman voor sleuteluitwisseling en authenticatie methodes (pre-shared keys, certificates).

LES 9 — VPNs, PKI & WireGuard

“Review some downsides of IPsec”

- Complex configuratie en interoperabiliteit issues.
- NAT traversal requires extra mechanisms (NAT-T).
- Performance overhead en MTU/fragmentatie issues.
- Troubleshooting is soms complex (policy vs SA mismatches).
- Management of certificates/PSKs can be heavy.

“Review what a CA is and how it works?”

CA = Certificate Authority: entiteit die public keys (CSR) verifieert en signed certs uitgeeft. Werkt in hiërarchie (root/intermediate), clients trust root CAs via trust stores.

“How is it possible to browse to https://chamilo.hogent.be without warnings? Explain.”

Omdat die site een geldig X.509 cert heeft waarvan de certificate chain leidt naar een trusted root CA in je browser/OS trust store. Browser valideert: hostname matches (CN/SAN), cert not expired/revoked en chain is trusted → geen warning.

“What is the goal of OpenVPN?”

OpenVPN doel: veilige, flexibele point-to-point of site-to-site VPN opzetten. Werkt op user-space en gebruikt SSL/TLS for key exchange.

“How does OpenVPN work? Crucial elements for setup?”

Belangrijke elementen:

- **PKI or pre-shared keys** voor authentication (server + client certs).
- **TUN/TAP device** voor creating virtual network interface.
- **TLS handshake** to authenticate and negotiate keys.
- **Routing / pushing routes** en firewall rules.
- Correct MTU and network config.

“What is PKI?”

PKI (Public Key Infrastructure) = framework for managing keys/certificates: CA’s, registration, issuance, revocation (CRL/OCSP), trust stores.

“Fundamental differences between IPsec and OpenVPN”

- **OSI layer:**
  - IPsec opereert op **Layer 3 (Network)** (kernel-level IP protection).
  - OpenVPN opereert op **Layer 2 or 3** via virtual network interfaces (user-space).
- **Goal:** Beide bieden confidentiality/integrity voor IP traffic, maar implementatie en use-cases verschillen.
- **Waarom kiezen?**
  - **IPsec**: native OS support, goed voor site-to-site, performanter in kernel.
  - **OpenVPN**: makkelijker NAT traversal, flexibeler config en vaak eenvoudiger client-setup (user-space).

“Is WireGuard more comparable to OpenVPN or to IPsec or to both? Explain.”

WireGuard is conceptueel **tussen beide in**:

- Net als **OpenVPN**: eenvoudiger config en moderne user-friendly setup.
- Net als **IPsec**: werkt op **Layer 3** en is zeer performante kernel-module (op Linux).
- WireGuard gebruikt moderne crypto primitives, kleinere codebase en simpler key model (no complex PKI built-in — uses static keys but can be integrated with PKI). Kort: WireGuard combineert performance en simplicity — dus het is vergelijkbaar met beide, maar technisch dichter bij IPsec qua laag en prestaties, en dichter bij OpenVPN qua eenvoud van deployment.