**Licensinformation**

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se licensvillkor nedan.

📄 Licens: Creative Commons BY-NC-SA 4.0

Du får dela och anpassa innehållet fritt, så länge du:

- Anger källan

- Inte använder det kommersiellt

- Delar vidare med samma licens

Mer info: https://creativecommons.org/licenses/by-nc-sa/4.0/

[1. Hårdvara 7](#__RefHeading___Toc11042_3533566184)

[1.1 Moderkortet 7](#__RefHeading___Toc11044_3533566184)

[1.2 Processorn (CPU) 11](#__RefHeading___Toc11046_3533566184)

[1.3 RAM (Arbetsminne) 19](#__RefHeading___Toc11048_3533566184)

[1.4 Långtidslagring 24](#__RefHeading___Toc11050_3533566184)

[1.5 PSU (Nätaggregat) 30](#__RefHeading___Toc11052_3533566184)

[1.6 GPU (Grafikkort) 33](#__RefHeading___Toc11054_3533566184)

[1.7 Portar och bussar 39](#__RefHeading___Toc11056_3533566184)

[2 Uppstart 48](#__RefHeading___Toc11058_3533566184)

[2.1 Introduktion 48](#__RefHeading___Toc11060_3533566184)

[2.2 Översikt av uppstartssekvensen 48](#__RefHeading___Toc11062_3533566184)

[2.3 Strömförsörjning och CPU-start 49](#__RefHeading___Toc11064_3533566184)

[2.4 BIOS 50](#__RefHeading___Toc11066_3533566184)

[2.5 Det tekniska bakom BIOS 50](#__RefHeading___Toc11068_3533566184)

[2.6 UEFI 51](#__RefHeading___Toc11070_3533566184)

[2.7 Secure Boot 52](#__RefHeading___Toc11072_3533566184)

[2.8 Bootloaders 52](#__RefHeading___Toc11074_3533566184)

[2.9 Uppstartssekvenser – exempel 53](#__RefHeading___Toc11076_3533566184)

[2.10 Vanliga problem 54](#__RefHeading___Toc11078_3533566184)

[2.11 Fördjupning 54](#__RefHeading___Toc11080_3533566184)

[2.12 Fördjupningslänkar 55](#__RefHeading___Toc11082_3533566184)

[2.13 Diskussions- och kontrollfrågor 55](#__RefHeading___Toc11084_3533566184)

[2.14 Egna anteckningar 56](#__RefHeading___Toc11086_3533566184)

[3 Virtualisering 57](#__RefHeading___Toc11088_3533566184)

[3.4 Hosted hypervisorer 58](#__RefHeading___Toc11098_3533566184)

[3.5 Koppling mot molnet 59](#__RefHeading___Toc11100_3533566184)

[3.6 Containrar (kort introduktion) 59](#__RefHeading___Toc11102_3533566184)

[3.7 Fördjupande länkar 60](#__RefHeading___Toc9583_1074010042)

[3.8 Kontrollfrågor 61](#__RefHeading___Toc9585_1074010042)

[3.9 Egna anteckningar 62](#__RefHeading___Toc9587_1074010042)

[4 Lagring 63](#__RefHeading___Toc11110_3533566184)

[4.1 Vad är lagring? 63](#__RefHeading___Toc11112_3533566184)

[4.2 Disk – den fysiska enheten 63](#__RefHeading___Toc11114_3533566184)

[4.3 Partition och partitionstabell 64](#__RefHeading___Toc11116_3533566184)

[4.4 Volym och filsystem 64](#__RefHeading___Toc11118_3533566184)

[4.5 Enhet i Windows – enhetsbokstav 66](#__RefHeading___Toc11120_3533566184)

[4.6 Sammanfattning 66](#__RefHeading___Toc11122_3533566184)

[4.7 Fördjupningslänkar 66](#__RefHeading___Toc11124_3533566184)

[4.8 Diskussions- och kontrollfrågor 67](#__RefHeading___Toc11126_3533566184)

[4.9 Egna anteckningar 67](#__RefHeading___Toc11128_3533566184)

[5 Windows 77](#__RefHeading___Toc11130_3533566184)

[5.1 Historia 77](#__RefHeading___Toc11132_3533566184)

[5.2 Uppbyggnad 81](#__RefHeading___Toc11134_3533566184)

[5.3 Fördelar och nackdelar med Windows som operativsystem 85](#__RefHeading___Toc11136_3533566184)

[5.4 Installation av Windows 11 i fysisk maskin/VM 90](#__RefHeading___Toc11138_3533566184)

[5.5 Konfiguration och hantering av Windows 94](#__RefHeading___Toc11140_3533566184)

[6 Windows Server 104](#__RefHeading___Toc11142_3533566184)

[6.1 Installation av Windows Server 104](#__RefHeading___Toc11144_3533566184)

[6.2 Workgroup, Domain och Active Directory 106](#__RefHeading___Toc11146_3533566184)

[6.3 Diskussions- och kontrollfrågor 110](#__RefHeading___Toc11148_3533566184)

[6.4 Fördjupningslänkar 111](#__RefHeading___Toc11150_3533566184)

[6.5 Egna anteckningar 112](#__RefHeading___Toc11152_3533566184)

[7. Linux – Historia, filosofi och ekosystem 113](#__RefHeading___Toc11154_3533566184)

[7.1 Vad är Linux? 113](#__RefHeading___Toc11310_3533566184)

[7.2 Installation av Ubuntu Desktop 118](#__RefHeading___Toc11156_3533566184)

[7.3 Konfiguration av Ubuntu Desktop 126](#__RefHeading___Toc11158_3533566184)

[7.3.5 Felsökning och underhåll 130](#__RefHeading___Toc11160_3533566184)

[7.4 Installation av Ubuntu Server och Webmin 136](#__RefHeading___Toc11162_3533566184)

[8 OSI-modellen 142](#__RefHeading___Toc11164_3533566184)

[8.1 Uppkomst – Bakgrunden till OSI 142](#__RefHeading___Toc11312_3533566184)

[8.2 Syfte – Varför finns OSI-modellen? 142](#__RefHeading___Toc11314_3533566184)

[8.3 Användningsområde – Hur används OSI-modellen? 143](#__RefHeading___Toc11316_3533566184)

[8.4 Diskussions- och kontrollfrågor 144](#__RefHeading___Toc11318_3533566184)

[8.5 Fördjupningslänkar 145](#__RefHeading___Toc11320_3533566184)

[8.6 Egna anteckningar 146](#__RefHeading___Toc11322_3533566184)

[9 Introduktion till IT-säkerhet 147](#__RefHeading___Toc11166_3533566184)

[9.1 Vad är IT-säkerhet? 147](#__RefHeading___Toc11168_3533566184)

[9.2 Skydd på olika nivåer 151](#__RefHeading___Toc11180_3533566184)

[9.3 Trådlös säkerhet 159](#__RefHeading___Toc11182_3533566184)

[9.4 Säkerhet i Windowsmiljö 162](#__RefHeading___Toc11184_3533566184)

[9.5 Säkerhet i Linuxmiljö 179](#__RefHeading___Toc11186_3533566184)

[9.7 Paketuppdatering och patchhantering 190](#__RefHeading___Toc11194_3533566184)

[9.8 Systemövervakning och loggning (journald, rsyslog, AIDE) 196](#__RefHeading___Toc9162_895980343)

[9.9 Root, sudo och användarbehörighet 200](#__RefHeading___Toc11196_3533566184)

[9.10 Loggning och övervakning i Linux 206](#__RefHeading___Toc11198_3533566184)

[9.11 Härdning av tjänster (t.ex. SSH, brandvägg, TLS/SSL) 212](#__RefHeading___Toc11200_3533566184)

[9.12 Backup och återställning 219](#__RefHeading___Toc11208_3533566184)

[9.13 Automatisering av säkerhetsuppgifter (cron, script, loggning) 225](#__RefHeading___Toc11216_3533566184)

[9.14 Säkerhetsverktyg i Linux 232](#__RefHeading___Toc11324_3533566184)

[9.15 IT-säkerhet ur ett verksamhetsperspektiv 239](#__RefHeading___Toc11224_3533566184)

[9.16 Moderna hot och skydd 265](#__RefHeading___Toc11232_3533566184)

[9.17 Verktyg, övervakning och loggning 273](#__RefHeading___Toc11240_3533566184)

[9.18 Säkerhet i skarpa miljöer och incidenthantering 280](#__RefHeading___Toc11248_3533566184)

[9.19 Social engineering, phishing och mänskliga misstag 286](#__RefHeading___Toc11256_3533566184)

[9.20 Loggning, övervakning och revision 293](#__RefHeading___Toc11264_3533566184)

[9.21 Backup och återställning 300](#__RefHeading___Toc11272_3533566184)

[9.22 Säkerhetskopiering i molnet 307](#__RefHeading___Toc11280_3533566184)

[9.23 Etisk hacking och hackingkultur 315](#__RefHeading___Toc11282_3533566184)

[9.24 Grundläggande säkerhetsprinciper 322](#__RefHeading___Toc11284_3533566184)

[9.25 IT-säkerhet på arbetsplatsen – framtid och ansvar 328](#__RefHeading___Toc11286_3533566184)

# **1. Hårdvara**

## **1.1 Moderkortet**

### **1.1.1 Vad är ett moderkort?**

Moderkortet är datorns centrala kretskort. Det fungerar som en plattform där alla andra komponenter ansluts och kommunicerar med varandra. Du kan tänka på det som datorns nervsystem – utan moderkortet kan inte processorn, minnet, lagringen eller grafikkortet samarbeta.

Moderkort tillverkas av olika företag, till exempel ASUS, MSI, Gigabyte och ASRock. Dessa tillverkare designar moderkort för olika behov, som gaming, kontorsdatorer eller servrar.

### **1.1.2 Funktion**

Moderkortets huvuduppgift är att koppla ihop alla komponenter i datorn och se till att de kan kommunicera med varandra. Det innehåller ledningsbanor och styrkretsar (chipset) som kontrollerar datatrafiken mellan processorn (CPU), arbetsminnet (RAM), grafikkortet (GPU), lagringsenheter (SSD/HDD) och andra anslutningar.

Moderkortet fördelar också ström från nätaggregatet till olika komponenter.

### **1.1.3 Viktiga delar på ett moderkort**

* **CPU-sockel:** Platsen där processorn installeras. Olika socklar passar olika CPU-modeller och tillverkare (t.ex. Intel eller AMD).
* **RAM-platser:** Platser för att installera arbetsminne (RAM-moduler). Antalet och typen varierar mellan moderkort.
* **Chipset:** En styrkrets som bestämmer hur snabbt och på vilket sätt olika delar kommunicerar. Chipsetet påverkar vilka funktioner moderkortet har.
* **PCIe-platser:** Platser för expansionskort som grafikkort, ljudkort eller nätverkskort.
* **M.2/lagringsplatser:** Snabba anslutningar för SSD-diskar.
* **Strömanslutningar:** Kontakter där nätaggregatet kopplas in för att förse moderkortet och processorn med ström.
* **SATA-portar:** Anslutningar för traditionella hårddiskar och SSD:er.
* **CMOS-batteri:** Ett litet batteri som håller inställningarna sparade även när datorn är avstängd.

### **1.1.4 Viktiga anslutningar på moderkortet**

* **Främre panel-anslutningar:** Kontakter för strömknapp, reset-knapp och lysdioder på chassit.
* **USB-header:** Interna kontakter för att koppla USB-portar på chassit.
* **Fläktkontakter:** För att ansluta och styra chassifläktar och CPU-kylare.
* **24-pin ATX-ström:** Huvudströmförsörjning från nätaggregatet till moderkortet.
* **8-pin CPU-ström:** Extra strömförsörjning till processorn.

### **1.1.5 Formfaktorer**

Moderkort finns i olika storlekar som kallas formfaktorer. Vanliga är:

* **ATX:** Standardstorlek med flest anslutningar och expansionsplatser.
* **microATX:** Lite mindre, färre platser men billigare och kompaktare.
* **Mini-ITX:** Mycket liten formfaktor för små datorbyggen, men har oftast färre anslutningar och platser.

### **1.1.6 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan ATX och Mini-ITX?
2. Vad gör ett chipset?
3. Vad händer om CMOS-batteriet tar slut?
4. Varför är CPU-sockeln viktig att välja rätt?

### **1.1.7 Fördjupande länkar**

* Intel – Vad är ett moderkort?  
   <https://www.intel.se/content/www/se/sv/gaming/resources/what-is-a-motherboard.html>
* MSI – Basic Motherboard Guide  
   <https://www.msi.com/blog/motherboard-basics-beginners-guide>
* ASUS – How to Choose a Motherboard  
   <https://www.asus.com/us/support/FAQ/1044664/>
* Wikipedia – Moderkort (svenska)  
   <https://sv.wikipedia.org/wiki/Moderkort>

### **1.1.8 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om moderkort, frågor eller reflektioner)*

## **1.2 Processorn (CPU)**

### **1.2.0 Inledning och historia**

En CPU (Central Processing Unit) är den del av datorn som utför instruktionerna i ett program. Det är viktigt att förstå att *program skrivs för en processorarkitektur*, inte direkt för ett operativsystem. Operativsystemet är istället ett lager mellan programmet och hårdvaran.

På 1970-talet och tidigt 80-tal fanns många olika typer av processorer med egna instruktionsuppsättningar. Intels 8086-processor (lanserad 1978) blev revolutionerande eftersom den introducerade x86-arkitekturen. Den hade en 16-bitars instruktionsuppsättning och var bakåtkompatibel med tidigare modeller (8080 och 8085), vilket gjorde det enklare för utvecklare att porta eller skriva program. x86-arkitekturen blev en standard som än idag används (numera som 64-bitarsversionen x86-64 eller AMD64).

**AMD:s roll i historien är också viktig.** AMD började som en tillverkare som licenstillverkade Intels x86-processorer på 80-talet, så att Intel inte hade monopol. Det innebar att både Intel och AMD tillverkade kompatibla processorer som kunde köra samma program. Under årens lopp utvecklade AMD egna unika processorer och innovationer, som introduktionen av 64-bitars x86-arkitekturen (AMD64) och flertrådningstekniker i konsumentprocessorer.

Under 80- och 90-talet fanns även andra tillverkare av x86-kompatibla CPU:er, till exempel Cyrix, VIA och NexGen, men hård konkurrens gjorde att de flesta försvann eller köptes upp. Idag är Intel och AMD de två dominerande tillverkarna för stationära och bärbara datorer.

AMD har också satsat mycket på *APU*-konceptet (Accelerated Processing Unit), där man integrerar CPU och GPU på samma chip. Det gör att enklare system (t.ex. laptops eller budgetdatorer) kan klara grafik utan ett separat grafikkort, samtidigt som det sparar ström och utrymme.

Bärbara spelkonsoler, som Steam Deck, använder ofta AMD-APU:er av denna anledning – du kan få mycket högre grafikprestanda på mindre yta och med lägre strömförbrukning.

Intel har liknande teknik med sina "integrated graphics", men AMD:s APU:er är ofta kända för något bättre grafikprestanda i instegssegmentet.

**Nomenklatur – hur läser man modellnamnen?**  
 Intel använder ofta namn som *Core i3*, *i5*, *i7* och *i9* för att visa prestandanivåer. Efter namnet kommer en sifferkombination som visar generation och modell, till exempel *i5-11500* betyder generation 11, modell 500. Bokstäver i slutet visar särskilda egenskaper:

* **K:** upplåst för överklockning
* **F:** saknar integrerad grafik
* **T:** låg effekt (lägre strömförbrukning)
* **HK/H:** hög prestanda i laptops (HK ofta överklockningsbar)

AMD:s Ryzen-processorer är indelade i serier som Ryzen 3, 5, 7 och 9. Siffernamnen liknar Intels modellnamn – till exempel *Ryzen 5 5600X* betyder Ryzen 5 (mellansegment), generation 5000 (Zen 3-arkitektur) och modell 600. Bokstäver kan visa egenskaper:

* **X:** högre klockfrekvens eller bättre binning
* **G:** integrerad grafik (APU)
* **U:** låg strömförbrukning för laptops
* **H/HS/HX:** hög prestanda i laptops (H-serien)

Genom att förstå dessa namn kan man lättare jämföra och välja rätt processor för olika behov – och se skillnaden på till exempel en högpresterande desktop-CPU, en energisnål laptop-CPU eller en APU med inbyggd grafik.

### **1.2.1 Vad är en CPU?**

En CPU är datorns *hjärna och arbetsledare*. Den utför instruktionerna i de program som körs, men ansvarar också för att styra och samordna resurserna i systemet. CPU:n tar emot instruktioner från minnet, avkodar dem och exekverar dem. Det kan handla om allt från att göra matematiska beräkningar till att styra annan hårdvara eller fördela arbetsuppgifter mellan olika delar av systemet.

En stor del av CPU:ns arbete handlar faktiskt om att göra *väldigt snabba och smarta gissningar* om vilka instruktioner som behövs härnäst (branch prediction), och om att hålla data redo för att undvika väntetider. På så sätt fungerar CPU:n inte bara som en räknemaskin utan också som en avancerad arbetsledare som organiserar och optimerar datorns arbete.

### **1.2.2 Funktion**

CPU:ns funktion kan delas in i tre huvudsakliga steg:

* **Fetch:** Processorn hämtar instruktioner från minnet (RAM).
* **Decode:** Den avkodar instruktionen för att förstå vad som ska göras.
* **Execute:** Den utför instruktionen (t.ex. en beräkning eller att flytta data).

Hur snabbt en CPU kan göra dessa steg bestäms av dess **klockfrekvens**. Klockfrekvensen mäts i gigahertz (GHz) och representerar antalet svängningar per sekund som styr processorcyklerna. Varje klockcykel är som ett "tick" som synkroniserar alla delar av CPU:n. En högre klockfrekvens betyder i princip att fler instruktioner kan utföras per sekund.

Vanliga basfrekvenser idag ligger mellan ca 2,5 och 4,0 GHz för stationära processorer. Moderna CPU:er har också **Turbo Boost** eller liknande teknik som höjer klockfrekvensen tillfälligt (t.ex. 4,5–5,5 GHz) när extra prestanda behövs, så länge temperaturen och strömförbrukningen tillåter det.

Men högre klockfrekvens innebär också högre **strömförbrukning och värmeutveckling**. Det finns en gräns för hur mycket man kan öka frekvensen innan processorn blir för varm eller ineffektiv. För att öka prestanda utan att spränga dessa gränser har man istället börjat använda **fler kärnor**, så att fler instruktioner kan köras parallellt.

### **1.2.3 Viktiga delar och egenskaper**

* **Cores / Threads:**  
   En modern CPU har flera kärnor (cores) som kan utföra beräkningar parallellt. Threads (trådar) är en teknik för att simulera flera parallella processer på en kärna (t.ex. Intels Hyper-Threading eller AMDs SMT).

Eftersom klockfrekvensen har fysikaliska begränsningar (värme, ström), har CPU-tillverkare istället ökat antalet kärnor för att höja den totala prestandan. Med fler kärnor kan flera program eller trådar arbeta samtidigt utan att en enskild kärna behöver köras extremt snabbt.

* **Cache:**  
   Ett litet men extremt snabbt minne inuti processorn. Cacheminnet lagrar data och instruktioner som används ofta så att CPU:n slipper hämta dem från det långsammare RAM-minnet. Det finns ofta flera nivåer av cache (L1, L2, L3) med olika hastighet och storlek.
* **Die / Dye och tillverkningsteknik:**  
   Processorns hjärta är en liten bit kisel som kallas *die*. På denna yta sitter miljarder transistorer som fungerar som strömbrytare i elektroniska kretsar. Moderna tillverkningsprocesser använder extremt små strukturer, ofta 3–7 nanometer (nm), vilket betyder att avståndet mellan transistorernas leder bara är några miljarddelar av en meter. Mindre tillverkningsprocesser innebär ofta högre prestanda, lägre strömförbrukning och möjlighet att få in fler kärnor på samma yta.

Hela *die* monteras på ett kretskort med kontakter och ett skyddande hölje som kallas *package*. I videor från t.ex. Linus Tech Tips ser man ofta när någon tar bort den övre metallkapseln och blottar själva *die*. OBS! Många kallar det felaktigt för *dye* på nätet – korrekt stavning är *die*.

* **Sockelkompatibilitet:**  
   CPU:n måste passa i moderkortets sockel. Intel och AMD har olika socklar, och även inom samma tillverkare varierar socklar mellan generationer. Att välja rätt sockel är avgörande vid datorbygge.
* **Arkitektur:**  
   En CPU:s design och instruktionsuppsättning avgör hur den fungerar. Exempel är x86-64 (vanlig i stationära datorer) och ARM (vanlig i mobiltelefoner och surfplattor). Arkitekturen bestämmer hur program måste skrivas för att kunna köras på processorn.

### **1.2.4 Vanliga tillverkare och modeller**

De största tillverkarna av CPU:er för persondatorer är Intel och AMD. Intel har serier som Core i3, i5, i7 och i9 som visar olika prestandanivåer. Till exempel är i3 enklare och billigare, medan i9 är avsedd för de mest krävande användarna. AMD använder Ryzen 3, 5, 7 och 9 på liknande sätt.

Efter modellnamnet kommer en sifferkombination som visar generation och modell. Exempelvis betyder *i5-11500* generation 11, modell 500. *Ryzen 5 5600X* betyder Ryzen 5-serien, generation 5000, modell 600 och bokstaven *X* för högre prestanda.

Intel och AMD tillverkar även serverprocessorer (Intel Xeon, AMD EPYC) och strömsnåla varianter för laptops med andra bokstavsändelser (t.ex. Intel U, H eller HK). ARM-baserade processorer (t.ex. Apple M-serien, Qualcomm Snapdragon) används i mobiltelefoner, surfplattor och ibland i laptops.

### **1.2.5 Installation och kylning**

En CPU installeras i moderkortets sockel. Det är avgörande att välja en processor som passar moderkortets sockeltyp. När processorn placeras i sockeln används kylpasta mellan CPU:n och kylaren för att förbättra värmeöverföringen.

Kylaren kan vara luftbaserad (med fläkt och kylfläns) eller vattenkyld. Effektiv kylning är nödvändig eftersom processorer genererar mycket värme under drift, särskilt vid hög klockfrekvens eller överklockning. För högpresterande system används därför ofta avancerade kylsystem.

### **1.2.6 Diskussions- och kontrollfrågor**

1. Vad menas med fetch, decode och execute?
2. Vad är skillnaden mellan en core och en thread?
3. Varför är cacheminnet viktigt?
4. Vad betyder 3 nm tillverkningsteknik?
5. Ge exempel på olika CPU-tillverkare och serier.
6. Varför behöver man kylpasta?
7. Vad kan hända om en CPU blir för varm?

### **1.2.7 Fördjupande länkar**

* Intel – How CPUs Work  
   <https://www.intel.com/content/www/us/en/gaming/resources/how-processors-work.html>
* AMD – Ryzen Processors Overview  
   <https://www.amd.com/en/products/ryzen-processors>
* Linus Tech Tips – What is a CPU? (YouTube)  
   <https://www.youtube.com/watch?v=cNN_tTXABUA>
* Wikipedia – Central Processing Unit  
   <https://en.wikipedia.org/wiki/Central_processing_unit>

### **1.2.8 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om CPU, frågor eller reflektioner)*

## **1.3 RAM (Arbetsminne)**

### **1.3.0 Inledning**

RAM står för *Random Access Memory* och kallas på svenska oftast arbetsminne. Det är en typ av snabbt, tillfälligt lagringsutrymme som datorn använder för att hålla program och data som används just nu. Utan tillräckligt med RAM blir datorn långsam eftersom den måste använda mycket långsammare lagringsutrymme på hårddisken eller SSD:n som "virtuellt minne".

RAM är **volatilt minne**, vilket betyder att allt innehåll raderas när datorn stängs av. Det är en viktig skillnad mot hårddiskar eller SSD som lagrar data permanent.

### **1.3.1 DDR-standarder**

Moderna datorer använder **DDR (Double Data Rate)** RAM. DDR innebär att minnet kan överföra data två gånger per klockcykel, vilket ökar hastigheten.

Det finns flera generationer:

* **DDR1:** Tidiga 2000-talet, långsam och föråldrad.
* **DDR2/DDR3:** Snabbare, används ibland fortfarande i äldre system.
* **DDR4:** Standard i många datorer idag. Erbjuder höga hastigheter och bra energieffektivitet.
* **DDR5:** Nyare standard med ännu högre hastigheter och förbättrad energieffektivitet. Börjar bli vanligare i moderna system.

Äldre och nyare DDR-standarder är inte kompatibla med varandra. Moderkortet stödjer alltid bara en viss DDR-generation.

### **1.3.2 Modultyper**

RAM finns i olika formfaktorer beroende på användningsområde:

* **DIMM:** Fullstorlek för stationära datorer.
* **SO-DIMM:** Mindre, används i laptops och små formfaktorsystem.
* **Serverminne:** Kan vara ECC (Error Correcting Code) för att automatiskt rätta fel, vilket är viktigt i servrar där stabilitet är avgörande.

SO-DIMM är kortare för att passa i trängre utrymmen, men fungerar på samma sätt.

### **1.3.3 Frekvens och latens**

RAM:s hastighet mäts oftast i MHz (megahertz), vilket anger hur många cykler per sekund minnet kan kommunicera på. Högre frekvens innebär mer data kan flyttas per sekund.

Men prestanda handlar inte bara om frekvens – även **latens (CAS-latency, CL)** är viktig. Latens är antalet klockcykler det tar att svara på en begäran. Låg latens ger snabbare svar, även om frekvensen är densamma.

Vanliga hastigheter idag är:

* DDR4: 2133–3600+ MHz
* DDR5: 4800–6000+ MHz

### **1.3.4 Dual Channel och flerkanaligt minne**

Många moderkort har stöd för **dual channel**, **triple channel** eller **quad channel**. Det betyder att de kan använda två eller fler RAM-moduler parallellt för att öka minnesbandbredden.

För att få fördelarna måste man installera minnena i rätt platser (ofta färgmarkerade på moderkortet).

✅ Exempel: Två identiska 8 GB-moduler i dual channel ger effektivare dataöverföring än en ensam 16 GB-modul.

✅ Fördelar: Bättre prestanda i spel, video- och bildredigering, och andra minnesintensiva uppgifter.

### **1.3.5 Placering nära CPU**

RAM sitter nära CPU:n på moderkortet för att hålla signalförluster och fördröjningar så låga som möjligt.

Ju kortare avstånd mellan CPU och RAM, desto snabbare och mer pålitlig kan datakommunikationen bli. Detta är en av anledningarna till varför moderna CPU-arkitekturer ofta har minneskontrollern inbyggd i själva processorn, för att minska latens ytterligare.

### **1.3.6 Kapacitet och energiförbrukning**

✅ **Kapacitet:**

* 8 GB: Minimum för enklare användning (kontor, surf).
* 16 GB: Standard för de flesta moderna system (spel, kreativt arbete).
* 32 GB eller mer: För tyngre uppgifter som videoredigering, CAD, virtuella maskiner.

✅ **Energiförbrukning:**

* Laptops använder ofta **Low Voltage DDR (LPDDR/DDR-L)** för att spara ström.
* Serverminne kan ha **ECC** som automatiskt korrigerar fel – lite dyrare och långsammare, men säkrare.

### **1.3.7 Skillnaden mellan RAM och lagring**

RAM är **volatilt minne** – innehållet försvinner när strömmen bryts. Det är som datorns korttidsminne: snabbt och tillgängligt för aktuell bearbetning.

Lagring (SSD/HDD) är däremot **permanent minne** som behåller data även när datorn stängs av. RAM är mycket snabbare men kan inte användas för långtidslagring.

### **1.3.8 Diskussions- och kontrollfrågor**

1. Vad betyder DDR?
2. Vad är skillnaden mellan DIMM och SO-DIMM?
3. Vad är fördelen med dual channel?
4. Vad händer om du blandar olika DDR-standarder?
5. Vad är CAS-latens?
6. Varför behöver RAM sitta nära CPU:n?
7. Vad är skillnaden mellan RAM och lagring?

### **1.3.9 Fördjupande länkar**

* Crucial – What is RAM?  
   <https://www.crucial.com/articles/about-memory/what-is-computer-memory>
* Kingston – Memory Types Explained  
   <https://www.kingston.com/en/memory/desktop-laptop-memory>
* Wikipedia – DDR SDRAM  
   <https://en.wikipedia.org/wiki/DDR_SDRAM>
* Techquickie – RAM Explained (YouTube)  
   <https://www.youtube.com/watch?v=9EaD8BfG2Hg>

### **1.3.10 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om RAM, frågor eller reflektioner)*

## **1.4 Långtidslagring**

### **1.4.0 Inledning**

Långtidslagring är det som gör att vi kan spara data permanent på en dator – till skillnad från RAM som är volatilt och försvinner när strömmen bryts. Operativsystem, program, bilder, dokument och spel lagras på enheter som är gjorda för att hålla data även när datorn stängs av.

De vanligaste typerna av långtidslagring idag är HDD (hårddiskar), SATA SSD och NVMe SSD. NVMe-enheter i M.2-formfaktor har blivit standard i nya datorer de senaste åren, tack vare mycket högre hastigheter. 2,5" SATA-SSD används ofta som billigare eller enklare komplement, medan HDD främst används för billig långtidslagring av stora datamängder – till exempel i servrar eller NAS-system.

### **1.4.1 HDD – Hårddiskar**

✅ **Uppbyggnad:**  
 En hårddisk är en mekanisk lagringsenhet med roterande magnetiska skivor (platter) och ett läs/skrivhuvud. Den är billig per gigabyte och lämpar sig därför mycket bra för **arkivering, backup och serverbruk** där stora datamängder behöver sparas till låg kostnad.

Idag används HDD allt mindre som primär lagring i konsumentdatorer, men den är fortfarande oumbärlig för långtidslagring av stora volymer data.

✅ **Teknik:**

* Skivorna roterar i hög hastighet (vanligtvis 5400 eller 7200 varv per minut – RPM).
* Data organiseras i spår och sektorer.
* Läs/skrivarmen flyttar sig för att nå rätt plats på skivan.

✅ **Kluster och defragmentering:**  
 Data sparas i små enheter som kallas *kluster*. När filer skrivs över tid kan de bli splittrade (fragmenterade) över skivan. Detta gör att läsarmen måste hoppa fram och tillbaka, vilket gör hårddisken långsammare.

**Defragmentering** är en process som flyttar och sammanfogar kluster så att filer ligger i ordning – vilket minskar söktiden och gör hårddisken snabbare. På mekaniska diskar är detta viktigt underhåll.

✅ **Hastigheter:**

* 5400 RPM: ca 80–100 MB/s sekventiell läs/skriv.
* 7200 RPM: ca 120–160 MB/s.
* Enterprise-diskar kan nå ännu högre.

✅ **Anslutningar:**  
 Nästan alla moderna HDD använder **SATA**-anslutning, vilket är standard för interna diskar.

### **1.4.2 SSD – Solid State Drives**

✅ **Uppbyggnad:**  
 SSD saknar rörliga delar och använder **flashminne** (NAND) för att lagra data. De finns både som **2,5" SATA-SSD** och som **NVMe-enheter i M.2-formfaktor**.

2,5" SATA-SSD används idag ofta som en **billigare eller enklare uppgradering** i äldre system eller som extra lagring, men har lägre hastigheter än NVMe. NVMe-SSD i M.2 är däremot den **vanligaste lösningen i nya datorbyggen** tack vare sina mycket högre överföringshastigheter.

✅ **Teknik:**

* NAND-flash lagrar data som elektriska laddningar.
* Finns olika typer av NAND beroende på hur många bitar som lagras per cell:
  + **SLC** (Single Level Cell) – 1 bit/cell, dyrt men snabbt och hållbart.
  + **MLC** (Multi Level Cell) – 2 bitar/cell.
  + **TLC** (Triple Level Cell) – 3 bitar/cell, vanligast för konsument.
  + **QLC** (Quad Level Cell) – 4 bitar/cell, billigare men långsammare och mindre hållbart.

✅ **Kluster, trim och varför man aldrig defragmenterar SSD:**  
 SSD har också filsystemkluster men fungerar annorlunda än HDD. Istället för att läsa sekventiellt med en arm kan SSD:n direkt nå alla minnesceller.

Fragmentering påverkar därför inte prestanda på samma sätt. Att defragmentera en SSD är faktiskt **dåligt** eftersom det orsakar onödigt skrivslitage.

Istället använder man **fstrim (eller TRIM-kommandot i Windows)** som berättar för SSD:n vilka block som inte längre används, så att den kan optimera platsen och förbereda block för att skriva effektivt. TRIM är viktigt för att hålla prestandan hög över tid.

✅ **Hastigheter:**

* SATA SSD: ca 400–550 MB/s (begränsas av SATA-bussen).
* NVMe SSD (se nästa avsnitt): flera GB/s.

✅ **Anslutningar:**

* SATA (2,5" formfaktor).
* M.2-kort som använder SATA eller NVMe.

### **1.4.3 NVMe – M.2 och PCIe-lagring**

✅ **Vad är NVMe?**  
 NVMe (Non-Volatile Memory Express) är ett modernt protokoll för SSD:er som använder PCIe-bussen istället för SATA. Det är idag **standardvalet för primär lagring i nya datorer**, tack vare mycket högre hastigheter och lägre latens.

NVMe-enheter kommer oftast i **M.2-formfaktor**, ett litet kort som sätts direkt på moderkortet. De har blivit den vanligaste formen av systemdisk i stationära och bärbara datorer de senaste åren.

✅ **Uppbyggnad:**

* Består av NAND-flash och en kontroller precis som andra SSD:er.
* Fysiskt ofta i M.2-formfaktor som är ett litet kort direkt på moderkortet.
* Kan också finnas som PCIe-kort eller i U.2/enterprise-form.

✅ **Teknikfördelar:**

* Mycket fler parallella kommandon än SATA.
* Lägre latens tack vare direktkoppling till CPU via PCIe.
* Mindre overhead.

✅ **Hastigheter:**

* NVMe SSD kan ofta nå 3000–7000+ MB/s.
* Nya PCIe 4.0/5.0-enheter kan vara ännu snabbare.

✅ **Anslutningar:**

* **M.2-slot** på moderkortet (vanligast).
* **U.2** för enterprise/server.
* **PCIe-slot** direkt på moderkortet (adapterkort).

### **1.4.4 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan HDD och SSD?
2. Varför behöver man defragmentera en HDD?
3. Varför ska man inte defragmentera en SSD?
4. Vad gör TRIM på en SSD?
5. Vad är NVMe och varför är det snabbare än SATA?
6. Vilka anslutningar används för lagring i en dator?

### **1.4.5 Fördjupande länkar**

* Kingston – SSD vs HDD  
   <https://www.kingston.com/en/blog/pc-performance/ssd-vs-hdd>
* Crucial – What is NVMe?  
   <https://www.crucial.com/articles/about-ssd/what-is-nvme-ssd>
* Wikipedia – Hard Disk Drive  
   <https://en.wikipedia.org/wiki/Hard_disk_drive>
* Wikipedia – Solid-state drive  
   <https://en.wikipedia.org/wiki/Solid-state_drive>

### **1.4.6 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om lagring, frågor eller reflektioner)*

## **1.5 PSU (Nätaggregat)**

### **1.5.0 Inledning**

PSU står för *Power Supply Unit* och är datorns nätaggregat. Det är komponenten som förser alla delar i datorn med ström. Utan ett PSU kan inget annat fungera – det är därför en absolut nödvändig del av varje datorbygge.

### **1.5.1 Växelström till likström**

PSU:ns främsta uppgift är att omvandla den **växelström (AC)** som kommer från eluttaget (vanligtvis 230 V i Europa) till **likström (DC)** som datorns komponenter kan använda.

I en dator används flera olika spänningar, till exempel:

* **12 V**: Till grafikkort, CPU, fläktar.
* **5 V**: Till USB-portar och vissa komponenter.
* **3.3 V**: Till vissa moderkortskretsar och minnen.

PSU:n innehåller olika spänningslinor för att leverera rätt spänning till olika komponenter på ett säkert och stabilt sätt.

### **1.5.2 Effekt och dimensionering**

PSU:ns effekt mäts i **watt (W)** och anger hur mycket ström den kan leverera totalt. När man väljer nätaggregat är det viktigt att räkna på den totala förbrukningen för alla komponenter – CPU, GPU, moderkort, lagring, fläktar – och sedan **ta höjd**.

✅ Rekommendation:

* Undvik att köpa ett PSU som ligger precis på gränsen.
* Satsa på minst 20–30 % högre watt-tal än beräknat behov för att ha marginal för uppgraderingar och för att undvika att köra nätaggregatet på max hela tiden.

✅ Exempel:

* Enkel kontorsdator: 300–400 W räcker ofta.
* Speldator med kraftfullt grafikkort: 600–850 W beroende på GPU.
* Högpresterande system eller fler GPU:er: 1000 W eller mer.

### **1.5.3 Effektivitetsklassning**

För att hjälpa kunder att förstå hur effektivt ett nätaggregat är finns **80 PLUS-certifieringar** som garanterar en viss verkningsgrad vid olika belastningar. Ju högre klassning, desto mindre ström går förlorad som värme.

✅ Vanliga nivåer:

* **80 PLUS Bronze**: Minst 82–85 % effektivitet.
* **Silver**: Lite bättre än Bronze.
* **Gold**: 87–90 % effektivitet – ett bra val för många byggare.
* **Platinum**: 90–94 % – mycket effektivt, ofta för entusiastbyggen eller servrar.
* **Titanium**: 94–96 % – toppklass, dyrt men extremt effektivt.

En bättre certifiering innebär mindre elförbrukning och lägre värmeutveckling.

### **1.5.4 Modulärt eller icke-modulärt**

✅ **Icke-modulärt PSU:**

* Alla kablar sitter fast.
* Billigare, men kan bli stökigt i chassit.

✅ **Modulärt PSU:**

* Löstagbara kablar.
* Du använder bara de kablar du behöver.
* Bättre kabeldragning och luftflöde.

✅ **Semi-modulärt:**

* Vissa kablar (t.ex. 24-pin och CPU) sitter fast, resten är löstagbara.

### **1.5.5 Diskussions- och kontrollfrågor**

1. Vad gör ett PSU i en dator?
2. Vad är skillnaden mellan växelström och likström?
3. Varför är det viktigt att ta höjd i watt-tal när du väljer PSU?
4. Vad betyder 80 PLUS Gold?
5. Vad är skillnaden mellan modulärt och icke-modulärt nätaggregat?
6. Vilka spänningar levererar ett PSU i en dator?

### **1.5.6 Fördjupande länkar**

* Corsair – PSU Buying Guide  
   <https://www.corsair.com/us/en/blog/psu-buying-guide>
* Seasonic – Efficiency Ratings Explained  
   <https://seasonic.com/80-plus>
* Wikipedia – Power Supply Unit (Computer)  
   <https://en.wikipedia.org/wiki/Power_supply_unit_(computer)>

### **1.5.7 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om PSU, frågor eller reflektioner)*

## **1.6 GPU (Grafikkort)**

### **1.6.0 Inledning**

GPU står för *Graphics Processing Unit* och är den del av datorn som är specialiserad på att bearbeta och rendera grafik. I en dator är GPU:n oftast ett separat kort (grafikkort) eller en del av CPU:n som integrerad grafik.

Till skillnad från CPU:n, som är bra på allmän beräkning och styrning, är GPU:n designad för att hantera många enkla beräkningar parallellt – perfekt för att rita bilder, 3D-modeller och video.

### **1.6.1 Funktion och användning**

GPU:ns huvuduppgift är att omvandla instruktioner från spel och program till bilder som visas på skärmen. Den är optimerad för **massivt parallell bearbetning**, vilket gör den snabb på uppgifter som kräver många liknande beräkningar.

GPU används inte bara till spel utan också för:

* 3D-rendering (film, arkitektur)
* Vetenskapliga beräkningar (GPU-accelererat arbete)
* AI och maskininlärning
* Videoacceleration och uppspelning

### **1.6.2 Minnet – GDDR**

GPU har eget specialiserat minne kallat **GDDR (Graphics Double Data Rate)**. Det är likt RAM men optimerat för hög bandbredd.

✅ Vad gör GDDR?

* Lagrar texturer, 3D-modeller, framebuffer.
* Ger GPU snabb åtkomst till data under rendering.

✅ Skillnad mot system-RAM:

* Snabbare men med högre latens.
* Byggt för bandbredd snarare än låg latens.

### **1.6.3 Buss-storlek**

**Minnesbussen** anger hur bred anslutningen mellan GPU och dess minne är (i bitar – t.ex. 128-bit, 256-bit).

En bredare buss kan överföra mer data per klockcykel:  
 ✅ Exempel:

* 128-bit buss = bra för enklare kort.
* 256-bit eller mer = för högpresterande kort.

✅ Bussbredd + GDDR-hastighet = minnesbandbredd.  
 Högre bandbredd är viktigt för högupplöst grafik och stora texturer.

### **1.6.4 RTX och strålspårning**

**RTX** är Nvidias varumärke för hårdvaruaccelererad ray tracing – en teknik som simulerar ljusets vägar realistiskt i 3D-scener.

✅ Vad är ray tracing?

* Exakt beräkning av skuggor, reflektioner, ljusbrytning.
* Ger mer realistisk grafik men är beräkningsintensivt.

✅ RTX-kort har dedikerade kärnor för ray tracing.  
 ✅ AMD har liknande teknik i sina RDNA 2/3-kort (Ray Accelerators).

### **1.6.5 SLI och CrossFire**

För att öka prestanda kunde man tidigare koppla ihop flera grafikkort:

* **SLI** (Scalable Link Interface) – Nvidia.
* **CrossFire** – AMD.

✅ Hur fungerade det?

* Delade renderingsarbetet mellan flera kort.
* Krävde stöd från spelutvecklare.

✅ Begränsningar:

* Stort strömbehov.
* Ojämn skalning.
* Drivrutinsstöd har minskat – idag används det nästan inte längre.

### **1.6.6 Vanliga tillverkare och serier**

✅ **Nvidia:**

* GeForce GTX/RTX-serier.
* RTX 20xx och senare har hårdvaru-ray tracing och DLSS.

✅ **AMD:**

* Radeon RX-serier.
* RDNA-arkitektur med ray tracing-stöd och FSR.

✅ **Intel:**

* Nya på marknaden med Intel Arc-serien.
* Fokus på prisvärd prestanda, stöd för ray tracing och XeSS.

### **1.6.7 Grafikinställningar och tekniker i spel**

✅ **Anti-aliasing (AA):**

* Minskar trappstegseffekter i kanter.
* Typer: MSAA, FXAA, TAA.

✅ **DLSS (Deep Learning Super Sampling):**

* Nvidias AI-baserade upscaling.
* Renderar i lägre upplösning men skalar upp med hög kvalitet.
* Ger högre FPS med bibehållen bildskärpa.

✅ **FSR (FidelityFX Super Resolution):**

* AMDs motsvarighet.
* Öppen standard, fungerar på fler kort.

✅ **Upscaling och sharpening:**

* Tekniker för att förbättra bild utan att rendera allt i full upplösning.

✅ **Varför är detta viktigt?**

* Balans mellan grafik och prestanda.
* Låter spelare justera hur mycket kraft som ska läggas på kvalitet kontra FPS.

### **1.6.8 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan GPU och CPU?
2. Vad gör GDDR-minne?
3. Vad betyder buss-storlek på ett grafikkort?
4. Vad är ray tracing?
5. Varför används SLI/CrossFire allt mindre idag?
6. Vad är DLSS och FSR?
7. Vad är fördelen med uppscaling-tekniker i spel?

### **1.6.9 Fördjupande länkar**

* Nvidia – What is Ray Tracing?  
   <https://www.nvidia.com/en-us/geforce/technologies/ray-tracing/>
* AMD – Radeon Graphics Overview  
   <https://www.amd.com/en/graphics/radeon-graphics>
* Intel – Intel Arc Graphics  
   <https://www.intel.com/content/www/us/en/products/docs/arc-discrete-graphics.html>
* Techquickie – How Graphics Cards Work (YouTube)  
   <https://www.youtube.com/watch?v=sKkKkJ4eJ3M>

### **1.6.10 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om GPU, frågor eller reflektioner)*

## **1.7 Portar och bussar**

### **1.7.0 Inledning**

En port eller buss är en fysisk eller elektrisk anslutning som låter olika komponenter i datorn kommunicera med varandra. Vissa portar sitter på moderkortet invändigt för att ansluta interna delar, medan andra är externa kontakter som används för att koppla in tillbehör som skärmar, USB-enheter och nätverk.

Att förstå hur dessa portar fungerar hjälper dig att välja rätt komponenter, bygga en dator och felsöka anslutningsproblem.

### **1.7.1 Interna anslutningar**

✅ **Vad är interna kontakter?**  
 Interna anslutningar på moderkortet är små stift eller headers som används för att koppla in funktioner på chassit, som knappar och lysdioder.

✅ **Jumpers för ström, reset, power LED:**

* **Power Switch (PWR SW):** Startar datorn när du trycker på knappen.
* **Reset Switch:** Startar om datorn.
* **Power LED:** Visar om datorn är på.
* **HDD LED:** Blinkar vid aktivitet på hårddisken.

Dessa kopplas via kablar från chassit till små pins på moderkortet – ibland kallat **front panel header**.

### **1.7.2 Interna portar och bussar**

#### **PCI**

✅ En äldre anslutningsstandard för expansionskort.

* Användes för ljudkort, nätverkskort m.m.
* Långsam jämfört med moderna alternativ.
* Idag nästan helt ersatt av PCI Express.

#### **PCI Express (PCIe)**

✅ Standardanslutningen för moderna expansionskort.

* Används för grafikkort, nätverkskort, lagringskort m.m.
* Bygger på **länkar/lanes** (x1, x4, x8, x16) som bestämmer hur mycket data som kan överföras parallellt.
* Snabb och flexibel – nya generationer (PCIe 4.0, 5.0) ger ännu högre hastigheter.

#### **SATA**

✅ Serial ATA – används för att ansluta lagringsenheter.

* Standard för 2,5" SSD och traditionella hårddiskar.
* SATA III har en teoretisk hastighet på **6 Gb/s** (gigabit per sekund) ≈ 600 MB/s.
* Men en del av hastigheten försvinner i overhead (felkorrigering, protokoll).
* Faktiska läs/skriv-hastigheter för SSD ligger ofta runt **400–550 MB/s**.
* Har ofta flera portar på moderkortet för att ansluta flera enheter.

#### **M.2**

✅ En kompakt intern portstandard som används för olika typer av kort.

✅ **Vanligast som lagring:**

* M.2 NVMe SSD – mycket snabb tack vare PCIe-bussen.
* M.2 SATA SSD – samma hastighet som vanlig SATA men i mindre formfaktor.

✅ **Men M.2 är inte bara för lagring:**

* WiFi- och Bluetooth-kort.
* 4G/5G-modem (framför allt i laptops).

M.2 är alltså en flexibel plats på moderkortet för olika typer av moduler – inte bara för att spara data.

### **1.7.3 Externa portar**

#### **USB (Universal Serial Bus)**

✅ **Historia och uppkomst:**  
 USB utvecklades i mitten av 1990-talet som en *universell standard* för att ansluta tillbehör till datorer. Innan USB behövde man olika portar (seriell, parallell, PS/2) och ofta starta om datorn eller installera drivrutiner för att byta mus eller tangentbord.

USB:s stora revolution var att skapa en standard där **alla enheter kunde kopplas in med samma kontakt**, med **plug-and-play** utan omstart, och där strömförsörjning kunde ges direkt via kabeln.

✅ **Hur fungerar det?**

* Plug-and-play: enheten identifieras automatiskt och ofta installeras drivrutiner automatiskt.
* Hot swapping: du kan koppla in och ur utan att starta om datorn.
* Strömförsörjning: USB-porten levererar ström, vilket gör att många små enheter inte behöver separat nätadapter.

✅ **USB 1.x – En början, men långsam:**

* Introducerades 1996.
* USB 1.1 (1998) var den första breda standarden.
* Hastighet: upp till **12 Mb/s**.
* Perfekt för tangentbord och möss men för långsam för lagring.

✅ **USB 2.0 – Stor förbättring och fortfarande vanlig:**

* Introducerades 2000.
* Hastighet: upp till **480 Mb/s** (~60 MB/s).
* Blev standard för externa hårddiskar, skrivare och USB-minnen.
* Bakåtkompatibel med USB 1.

✅ **USB 3.x – Hög hastighet i flera generationer:**

* **USB 3.0** (2010): 5 Gb/s (~500 MB/s).
* **USB 3.1 Gen 2**: 10 Gb/s (~1 GB/s).
* **USB 3.2 Gen 2x2**: 20 Gb/s (~2 GB/s).
* Används för snabba externa SSD, videoöverföring, dockingstationer.
* Bakåtkompatibel men behöver rätt kablar och portar för maxhastighet.
* USB 3-kontakter har ofta **blå färg** eller är märkta med "SS" (SuperSpeed).

✅ **USB4 och Thunderbolt-kompatibilitet:**

* USB4 bygger på Thunderbolt 3-specen.
* Upp till **40 Gb/s**.
* Stödjer video (DisplayPort) och ström via USB-C.
* Kan ansluta externa GPU:er, skärmar och snabba lagringsenheter.

✅ **USB PD (Power Delivery):**

* En modern standard för att leverera mer ström via USB.
* Stödjer upp till **100 W eller mer** via USB-C.
* Laddar laptops, surfplattor och mobiler via samma kabel.
* Dynamisk förhandling av spänning och ström (t.ex. 5 V, 9 V, 20 V).

✅ **Begränsningar:**

* Kabellängd påverkar hastighet och ström.
  + USB 2.0: ~5 m max för full hastighet.
  + USB 3.x: kortare kablar (typiskt ~3 m) för att bibehålla SuperSpeed.
* För längre sträckor används hubbar eller aktiva kablar.
* Många versioner och kontakter kan skapa förvirring.

✅ **Sammanfattning:**  
 USB är en extremt mångsidig standard som ersatt äldre portar och gjort datoranvändning enklare. Från långsamma möss på 12 Mb/s till moderna USB4-lösningar som klarar 40 Gb/s och laddar din laptop – USB är en hörnsten i modern IT.

#### **HDMI**

✅ Digital video- och ljudanslutning för skärmar och TV-apparater.

* Vanligaste porten på grafikkort och laptops.
* Stödjer både ljud och högupplöst video.

#### **RJ45 (Ethernet)**

✅ Används för trådbunden nätverksanslutning.

* Vanligt i datorer, servrar och routrar.
* Stödjer olika hastigheter (100 Mbit/s, 1 Gbit/s, 10 Gbit/s).

#### **Thunderbolt**

✅ En snabb anslutning utvecklad av Intel och Apple.

* Kombinerar PCIe och DisplayPort över en USB-C-kontakt.
* Stödjer höga överföringshastigheter och anslutning av skärmar, externa GPU:er och lagring.
* Thunderbolt 3 och 4 är vanligast idag.

#### **FireWire (IEEE 1394)**

✅ Äldre standard för höghastighetsanslutning av externa enheter.

* Utvecklades med stort stöd från Apple, som använde den flitigt i sina datorer för video- och ljudredigering.
* Användes för att ansluta DV-videokameror, ljudinterface och hårddiskar, med höga överföringshastigheter för sin tid.
* Apple satsade på FireWire istället för USB 1.1 som de ansåg var för långsamt.
* Har i princip ersatts av USB 2.0/3.x och Thunderbolt i modern utrustning.

#### **DisplayPort**

✅ Digital anslutning för skärmar.

* Vanlig på grafikkort.
* Stödjer höga upplösningar, hög uppdateringsfrekvens och flera skärmar via daisy chaining.

### **1.7.4 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan interna och externa portar?
2. Vad används en jumper till på moderkortet?
3. Vad är skillnaden mellan PCI och PCIe?
4. Vad används SATA till och vad är dess faktiska hastighet?
5. Varför är M.2 inte bara för lagring?
6. Vad gjorde USB till en stor förbättring mot äldre anslutningar?
7. Vad är USB PD och varför är det relevant idag?

### **1.7.5 Fördjupande länkar**

* Intel – What is PCI Express?  
   <https://www.intel.com/content/www/us/en/io/pci-express/pci-express-technology.html>
* SATA-IO – About SATA  
   <https://sata-io.org/sata-overview>
* USB.org – USB Explained  
   <https://www.usb.org/>
* Wikipedia – Computer Bus  
   <https://en.wikipedia.org/wiki/Computer_bus>

### **1.7.6 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om portar och bussar, frågor eller reflektioner)*

# **2 Uppstart**

## **2.1 Introduktion**

När vi slår på en dator startar en noga planerad kedja av steg som till slut laddar operativsystemet i minnet. Den här processen kallas **uppstart** eller **boot** och är avgörande för att en dator överhuvudtaget ska fungera.

I det här kapitlet går vi igenom vad som händer tekniskt när datorn startar. Vi tittar på traditionell **BIOS** och den modernare **UEFI**, vad de gör, varför de finns och hur bootloaders som GRUB och Windows Boot Manager fungerar.

Vi avslutar med en fördjupning som tar upp historik, detaljerad teknik och Secure Boot.

## **2.2 Översikt av uppstartssekvensen**

När du trycker på strömknappen händer följande i korthet:

1. **Strömförsörjning (PSU) aktiveras** och skickar stabil spänning till moderkortet.
2. **CPU:n startar** vid en definierad adress och söker efter startkod i firmware.
3. **POST** (Power-On Self Test) körs för att testa grundläggande hårdvara.
4. **Firmware (BIOS/UEFI)** letar efter en boot-enhet enligt boot-ordningen.
5. **Bootloader** från boot-enheten laddas in i minnet.
6. **Bootloadern** hittar och laddar operativsystemets kärna.
7. **Operativsystemet** tar över och fortsätter startsekvensen.

## **2.3 Strömförsörjning och CPU-start**

### **2.3.1 Power Supply Unit (PSU)**

När strömknappen trycks ned skickar PSU ström till moderkortet. Den skickar en **Power Good-signal** när spänningen är stabil. Detta signalerar att moderkortet kan starta CPU:n.

### **2.3.2 CPU:s första steg**

En x86-baserad CPU börjar alltid på en välkänd adress. Vid kallstart rensas registren och CPU:n ställs in på:

* **CS (Code Segment) Selector:** 0xF000
* **IP (Instruction Pointer):** 0xFFF0

Adderat som fysisk adress:

CS base: 0xFFFF0000  
IP: 0x0000FFF0  
---------------------  
= 0xFFFFFFF0

Detta är den sista delen av adressrymden – här ligger en **jump-instruktion** till BIOS/UEFI. CPU:n börjar alltså med att hoppa in i firmware.

## **2.4 BIOS**

### **2.4.1 Vad är BIOS?**

BIOS står för **Basic Input/Output System** och är firmware lagrad på moderkortets ROM-chip. Den är låg-nivåkod som lever kvar oavsett om hårddisken är tom eller trasig.

### **2.4.2 Varför finns BIOS?**

Syftet är att ge en minimal uppstartsmiljö som kan:

* Testa hårdvaran (POST)
* Initiera grundläggande I/O (tangentbord, skärm)
* Leta upp en startbar enhet (HDD, SSD, USB, CD)
* Ladda och överlåta kontrollen till en bootloader

### **2.4.3 POST – Power-On Self Test**

POST är det första som BIOS gör. Det är en serie tester som säkerställer att:

* RAM fungerar
* CPU fungerar
* Grafikkort är åtkomligt
* Tangentbord svarar

Eventuella fel signaleras med pipkoder eller blinkande lysdioder. Vid godkänd POST går systemet vidare till nästa steg: att hitta bootenheten.

## **2.5 Det tekniska bakom BIOS**

* BIOS är traditionellt lagrat i ROM eller Flash på moderkortet.
* Den innehåller bootstrapkod som alltid är tillgänglig.
* Den har en **boot order** som kan konfigureras av användaren för att prioritera exempelvis HDD, SSD, USB eller nätverk.
* BIOS använder MBR (Master Boot Record) som partitionstabell på diskar upp till 2 TB.
* MBR har en 512-byte bootsektor där en liten del av bootloadern ligger.

## **2.6 UEFI**

### **2.6.1 Vad är UEFI?**

**Unified Extensible Firmware Interface (UEFI)** är en modernare ersättare för BIOS.

* Lagrad i flashminne på moderkortet.
* Har grafiskt gränssnitt, musstöd.
* Kan läsa större diskar (>2 TB) via GPT (GUID Partition Table).
* Har nätverksstöd i firmware.
* Kan köra drivrutiner direkt.

### **2.6.2 Historia och utveckling**

UEFI utvecklades från Intel EFI och blev en industristandard i början av 2010-talet. Syftet var att överkomma BIOS-begränsningar:

* Större diskar via GPT
* Moderna drivrutiner
* Modulär design
* Snabbare uppstart

### **2.6.3 Varför är UEFI bättre?**

* Stöd för >2 TB-diskar
* Säkerhetsfunktioner som Secure Boot
* Bättre konfigurationsmöjligheter
* Stöd för mus och GUI
* Snabbare POST och uppstart

## **2.7 Secure Boot**

### **2.7.1 Vad är Secure Boot?**

Secure Boot är en UEFI-funktion som kontrollerar att operativsystemets bootloader är kryptografiskt signerad. Den hindrar obehörig kod från att köras vid uppstart.

### **2.7.2 Fördelar**

* Skydd mot rootkits och bootkits
* Säker kedja från firmware till OS
* Förhindrar obehörig manipulation av bootloadern

### **2.7.3 Nackdelar**

* Kan hindra installation av osignerade operativsystem
* Kräver certifikatunderhåll (tillverkare kan dra tillbaka certifikat)
* Kan behöva inaktiveras för vissa Linux-distributioner eller specialsystem

## **2.8 Bootloaders**

### **2.8.1 Vad är en bootloader?**

En bootloader är ett litet program som firmware laddar från disken. Dess uppgift är att:

* Initiera hårddiskläsning
* Ladda operativsystemets kärna i RAM
* Ge kontrollen till kärnan

### **2.8.2 Exempel på bootloaders**

* **Windows Boot Manager** (BCD-store)
* **GRUB** (Grand Unified Bootloader) för Linux
* **boot.efi** för macOS

### **2.8.3 Bootloaderns plats**

* BIOS: MBR (512 byte bootstrap + ~31 KB VBR) pekar vidare
* UEFI: EFI-systempartition (ESP) innehåller .efi-filer

## **2.9 Uppstartssekvenser – exempel**

### **2.9.1 Windows (UEFI)**

1. Ström på → UEFI POST
2. UEFI läser EFI-partitionen
3. Laddar **bootmgfw.efi**
4. Läser BCD-databas
5. Laddar **winload.efi** → Kernel

### **2.9.2 macOS**

1. UEFI POST
2. Laddar **boot.efi**
3. Laddar XNU-kärnan

### **2.9.3 Linux (GRUB)**

1. UEFI POST
2. Laddar **grubx64.efi** från ESP
3. GRUB visar meny
4. Användaren väljer kernel
5. GRUB laddar vmlinuz och initramfs
6. Kernel startar

## **2.10 Vanliga problem**

* "No bootable device found" – fel boot order
* Korrupt bootloader – behöver repareras
* Secure Boot-blockering av osignerade OS
* Felaktiga partitionstabeller (MBR/GPT-konflikter)

## **2.11 Fördjupning**

### **2.11.1 BIOS och MBR – tekniskt**

* MBR är alltid första 512 bytes på disken
* Innehåller bootstrap-kod (440 bytes), partitionsschema (64 bytes)
* Kan bara ha fyra primära partitioner
* Använder CHS/LBA-adressering: CHS (Cylinder-Head-Sector) beskriver disken med dess fysiska geometri, medan LBA (Logical Block Addressing) använder sekventiella blocknummer för att enklare adressera stora diskar.

### **2.11.2 UEFI och GPT – tekniskt**

* GPT använder GUID-identifierare
* Stöd för nästan obegränsat antal partitioner
* ESP (EFI System Partition) lagrar .efi-bootloaders
* Variabel lagring i NVRAM

### **2.11.3 Secure Boot**

* Microsofts nycklar är förinstallerade på många system
* Linux-distributioner signerar bootloaders med dessa nycklar
* Kan stängas av i UEFI-inställningar

## **2.12 Fördjupningslänkar**

1. <https://en.wikipedia.org/wiki/Power-on_self-test>
2. <https://en.wikipedia.org/wiki/BIOS>
3. <https://uefi.org/specifications>
4. <https://wiki.archlinux.org/title/GRUB>
5. <https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot>

## **2.13 Diskussions- och kontrollfrågor**

* Vad är syftet med POST?
* Vad gör BIOS/UEFI för att hitta ett operativsystem?
* Vilka tekniska begränsningar har MBR jämfört med GPT?
* Vad är skillnaden mellan BIOS och UEFI?
* Vad är Secure Boot och vilka för- respektive nackdelar har det?
* Hur skiljer sig en bootloader i UEFI från den i ett MBR-system?
* Hur fungerar boot-sekvensen i Linux med GRUB?

## **2.14 Egna anteckningar**

*(Här kan du som elev skriva egna anteckningar om uppstartsprocessen, BIOS/UEFI, bootloaders eller reflektioner.)*

## **3 Virtualisering**

### **3.1 Vad innebär virtualisering**

Virtualisering betyder att man skapar en virtuell version av något som annars är fysiskt. Oftast handlar det om att köra flera virtuella datorer (VM – Virtual Machines) på en och samma fysiska server. Varje VM fungerar som en egen dator med operativsystem och program, men delar resurser som CPU, minne och disk med andra VM på samma hårdvara.

Virtualisering styrs av ett lager som kallas **hypervisor**. Hypervisorn gör det möjligt att starta, stoppa och hantera virtuella maskiner på ett kontrollerat sätt.

### **3.2 Varför använder man virtualisering**

Virtualisering är en central teknik inom IT-drift och utveckling. Några vanliga anledningar till att använda virtualisering är:

* **Bättre resursutnyttjande:** Flera VM på en fysisk maskin gör att hårdvaran används mer effektivt.
* **Kostnadsbesparing:** Färre fysiska servrar ger lägre kostnader för inköp, el och kyla.
* **Flexibilitet:** Lätt att skapa, ändra eller ta bort VM efter behov.
* **Isolering:** Problem i en VM påverkar inte andra VM.
* **Test och utveckling:** Möjlighet att snabbt skapa testmiljöer.

### **3.3 Bare-metal hypervisorer**

En bare-metal hypervisor (även kallad typ 1-hypervisor) körs direkt på serverns hårdvara, utan att först installera ett värdoperativsystem. Den blir i princip serverns "operativsystem".

### **3.3.1 Fördelar och nackdelar**

**Fördelar:**

* Mycket bra prestanda eftersom hypervisorn har direkt åtkomst till hårdvaran.
* Bättre säkerhet och isolering.
* Vanligt val i datacenter.

**Nackdelar:**

* Kan vara mer komplex att installera och administrera.
* Mindre bra för småskaligt bruk eller på klientdatorer.

### **3.3.2 Exempel på plattformar**

* VMware ESXi
* Microsoft Hyper-V (Server-installation)
* KVM (Linux-baserad)
* Proxmox (som vi kommer att arbeta med i kursen)

## **3.4 Hosted hypervisorer**

En hosted hypervisor (typ 2-hypervisor) installeras som ett program på ett vanligt operativsystem (som Windows, macOS eller Linux).

### **3.4.1 Fördelar och nackdelar**

**Fördelar:**

* Enkel att installera och använda.
* Bra för labb, utveckling och utbildning.
* Kan köras på en vanlig dator.

**Nackdelar:**

* Lite sämre prestanda än bare-metal.
* Dubbla lager (värd-OS + hypervisor) kan ge fler attackytor.

### **3.4.2 Exempel på plattformar**

* Oracle VirtualBox
* VMware Workstation / Fusion
* Parallels (för macOS)

## **3.5 Koppling mot molnet**

Många molntjänster bygger på virtualisering i grunden. Stora molnleverantörer som Microsoft Azure, Amazon AWS och Google Cloud Platform använder virtualisering för att kunna erbjuda:

* Virtuella servrar (t.ex. Azure Virtual Machines)
* Containerplattformar (t.ex. Google Kubernetes Engine)
* Serverless-tjänster (t.ex. AWS Lambda)

För kunden betyder det att man inte behöver köpa egen hårdvara utan "hyr" resurser som är virtualiserade i leverantörens datacenter.

## **3.6 Containrar (kort introduktion)**

Containrar är ett annat sätt att paketera och isolera applikationer. Till skillnad från virtuella maskiner delar containrar operativsystemets kärna, vilket gör dem mycket lättare och snabbare att starta.

Exempel på containerplattformar:

* Docker
* Podman
* Kubernetes (för att hantera många containrar)

Containrar används ofta tillsammans med virtualisering eller i molnmiljöer. Det är bra att känna till skillnaden:

* **Virtuella maskiner** emulerar hela hårdvaran och kör ett eget OS.
* **Containrar** delar värd-OS men isolerar applikationen.

## 3.7 Fördjupande länkar

Här är några länkar för dig som vill läsa mer om virtualisering:

* VMware – Vad är virtualisering?  
   <https://www.vmware.com/se/topics/glossary/content/virtualization.html>
* Microsoft Learn – Hyper-V översikt  
   <https://learn.microsoft.com/sv-se/virtualization/hyper-v-on-windows/>
* Proxmox VE – Officiell dokumentation  
   <https://pve.proxmox.com/wiki/Main_Page>
* Oracle VirtualBox – User Manual  
   <https://www.virtualbox.org/manual/>
* Docker – Vad är en container?  
   <https://www.docker.com/resources/what-container/>

## 3.8 Kontrollfrågor

1. Vad menas med virtualisering inom IT?
2. Vad är en hypervisor?
3. Nämn två fördelar med att använda virtualisering.
4. Vad är skillnaden mellan en bare-metal hypervisor och en hosted hypervisor?
5. Ge ett exempel på en bare-metal hypervisor.
6. Ge ett exempel på en hosted hypervisor.
7. Vilka är några fördelar med att använda virtualisering i molnet?
8. Vad är en container, och hur skiljer den sig från en virtuell maskin?
9. Varför är isolering viktigt när man kör flera system på samma hårdvara?
10. Nämn en plattform eller tjänst som använder virtualisering i molnet.

## 3.9 Egna anteckningar

*(Lämna plats här i kompendiet för eleverna att själva skriva)*

# **4 Lagring**

## **4.1 Vad är lagring?**

Lagring är grunden för hur vi sparar och organiserar data på datorer och andra enheter. All information – operativsystem, appar, dokument och bilder – behöver någon form av lagringsmedia. Det kan vara allt från hårddiskar och SSD:er till USB-minnen och SD-kort.

I det här kapitlet går vi igenom några viktiga begrepp:

* **Disk (fysisk enhet)**
* **Partition och partitionstabell (MBR/GPT)**
* **Volym och filsystem (FAT32, exFAT, ext4, NTFS, APFS)**
* **Enhet i Windows (enhetsbokstäver)**

## **4.2 Disk – den fysiska enheten**

En disk är den faktiska hårdvaran som lagrar data. Det kan vara:

* Mekanisk hårddisk (HDD)
* Solid State Drive (SSD)
* USB-minne
* SD-kort

### **🗂️ Ikea-liknelsen**

**Disken är som själva garderoben (PAX-stommen).** Det är det fysiska skalet som allt annat sitter i.

## **4.3 Partition och partitionstabell**

En disk kan delas upp i **partitioner**. Partitionen är en avgränsad del av disken som kan användas som en separat enhet. För att datorn ska veta hur partitionerna är upplagda används en **partitionstabell**.

De två vanligaste partitionstabellerna är:

* **MBR (Master Boot Record)**
  + Äldre standard
  + Max 4 primära partitioner
  + Max 2 TB per disk
  + Sårbar: all information om diskens struktur lagras i en liten MBR-del – blir den förstörd kan allt gå förlorat
* **GPT (GUID Partition Table)**
  + Nyare standard
  + Nästan obegränsat antal partitioner
  + Stöd för mycket större diskar
  + Inbyggd redundans (lagrar flera kopior av partitionstabellen)

### **📚 Biblioteksliknelsen**

* **MBR är som en ensam bibliotekarie som vet var alla böcker finns.** Om hon blir sjuk (eller får "virus") blir det kaos – ingen annan har koll.
* **GPT är som ett bibliotek med många bibliotekarier.** Även om några faller bort finns alltid fler som vet var allt finns.

## **4.4 Volym och filsystem**

En **volym** är det som operativsystemet ser som en "disk". Volymen skapas ovanpå en partition och har ett **filsystem** som strukturerar hur filer och mappar sparas.

Vanliga filsystem:

* **FAT32**
  + Stöd på nästan alla enheter
  + Max filstorlek: 4 GB
  + Bra för kompatibilitet, men begränsningar för stora filer
* **exFAT**
  + Nyare variant
  + Inga praktiska filstorleksgränser för vanliga behov
  + Bra för USB-stickor mellan Windows och Mac
* **NTFS**
  + Windows-standard
  + Stöd för stora filer
  + Rättigheter och journalföring
* **ext4**
  + Vanligt i Linux
  + Stabilt, snabbt, bra för servrar
* **APFS**
  + Apples filsystem för macOS
  + Modern funktionalitet: snapshot, kryptering

### **🗂️ Ikea-liknelsen – lådor och fack**

* **Partitionen är som lådorna eller stången i garderoben.** Du bestämmer hur du vill dela in din garderob.
* **Volymens filsystem är som insatser i lådan.** Om lådan är uppdelad i små fack (FAT32) kan du inte lägga ner en enorm filt på 6 GB – den får inte plats. Men med större fack (exFAT, NTFS) går det bra.

## **4.5 Enhet i Windows – enhetsbokstav**

I Windows får varje volym en **enhetsbokstav** (C:, D:, E: osv.). Det är Windows sätt att identifiera enheter. För att kunna användas måste enheten ha en bokstav tilldelad.

På Linux/macOS används istället **monteringspunkter** i filsystemsträdet, t.ex. /mnt/usb.

## **4.6 Sammanfattning**

* **Disk**: den fysiska lagringsenheten
* **Partition**: en del av disken, definierad i partitionstabellen (MBR/GPT)
* **Volym**: en partition med filsystem
* **Filsystem**: bestämmer hur filer organiseras
* **Enhet (Windows)**: tilldelas enhetsbokstav

## **4.7 Fördjupningslänkar**

1. Partitioning explained (How-To Geek):  
    <https://www.howtogeek.com/192772>
2. MBR vs GPT (Microsoft Docs):  
    <https://learn.microsoft.com/en-us/windows/win32/fileio/mbr-vs-gpt>
3. Filesystems explained (DigitalOcean):  
    <https://www.digitalocean.com/community/tutorials/an-introduction-to-filesystems>
4. NTFS vs FAT32 vs exFAT (Lifewire):  
    <https://www.lifewire.com/ntfs-vs-fat32-vs-exfat-4160714>
5. Apple's APFS Overview:  
    <https://support.apple.com/en-us/HT208018>

## **4.8 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan en disk och en partition?
2. Vad används en partitionstabell till?
3. Förklara med egna ord MBR:s begränsningar.
4. Vad är en fördel med GPT jämfört med MBR?
5. Ge ett exempel på när FAT32 kan vara ett dåligt val.
6. Vad är ett filsystem och varför behövs det?
7. Nämn två skillnader mellan exFAT och FAT32.
8. Vilket filsystem är standard på Windows-datorer?
9. Vad är enhetsbokstäver i Windows och varför behövs de?
10. Hur skiljer sig monteringspunkter i Linux/macOS från enhetsbokstäver i Windows?

## **4.9 Egna anteckningar**

*(Här kan du som elev skriva egna stödanteckningar, exempel eller frågor.)*

### **4.10 RAID och LVM – Redundans och flexibel lagring**

Lagring handlar inte bara om hur mycket utrymme man har – utan hur man fördelar det, säkrar det och anpassar det för framtiden. I detta kapitel går vi igenom två tekniker som hjälper till med just det: **RAID** och **LVM**.

#### **4.10.1 Lagringsförluster och effektiv användning**

För att förstå RAID och LVM är det viktigt att först greppa att det alltid finns en skillnad mellan fysisk kapacitet (den storlek som anges på en hårddisk, t.ex. 2TB) och den tillgängliga kapaciteten som faktiskt kan användas i operativsystemet. En del av utrymmet går förlorat på grund av metadata, filsystemets struktur och olika typer av overhead. I RAID-konfigurationer används dessutom delar av diskarna för redundans och felskorrigering, vilket ytterligare minskar det användbara utrymmet. Operativsystem och filsystem tar plats, och i redundanssystem som RAID förloras också diskar för att skapa säkerhet.

Ett tumregeln är att **ca 15% försvinner** till overhead. Den effektiva lagringen kan då beräknas med formeln:

**Effektiv lagring (Ze):**

X \* Y \* 0,85

Där:

* **X** = storlek på varje disk
* **Y** = antal diskar som används för lagring (ej paritet/spegling)

Om du vill veta hur mycket du **behöver köpa** för att få ett visst utrymme, använd: Ze / 0,85 = X \* Y

#### **4.10.2 Vad är RAID?**

**RAID** står för *Redundant Array of Independent Disks*. Det innebär att man kopplar samman flera hårddiskar för att de ska samarbeta och få antingen högre prestanda, högre säkerhet – eller båda.

##### **RAID med mjukvara eller hårdvara**

* **Hårdvaru-RAID** sker i en RAID-kontroller som sitter på moderkortet eller som separat kort. Operativsystemet ser hela RAID-arrayen som en vanlig disk.
* **Mjukvaru-RAID** sker i operativsystemet (t.ex. i Linux via mdadm). Det är billigare, mer flexibelt, men kan belasta CPU:n mer.

#### **4.10.3 RAID-nivåer**

**RAID 0**

*Striping* utan redundans. Alla diskar skrivs till samtidigt, vilket ger hög prestanda, men om en disk går sönder förlorar du allt.

* Effektiv lagring: n \* X \* 0,85
* Fördelar: Snabbt
* Nackdelar: Ingen säkerhet

**RAID 1**

*Spegling*. Varje disk har en exakt kopia. Om en disk går sönder tar den andra över.

* Effektiv lagring: (n / 2) \* X \* 0,85
* Fördelar: Säkerhet
* Nackdelar: Hälften av utrymmet försvinner

**RAID 5**

*Striping med paritet*. En disk används för felskorrigeringsdata. Klarar att en disk kraschar.

* Effektiv lagring: (n - 1) \* X \* 0,85
* Fördelar: Bra balans mellan prestanda och säkerhet
* Nackdelar: Kan vara segt vid återuppbyggnad

**RAID 6**

Som RAID 5, men med två paritetsdiskar. Klarar att två diskar kraschar samtidigt.

* Effektiv lagring: (n - 2) \* X \* 0,85
* Fördelar: Mer säkerhet
* Nackdelar: Mindre lagringsyta, lägre skrivprestanda

**RAID 10**

Kombination av RAID 1 och 0 (först spegling, sedan striping). Hög prestanda och säkerhet, men kräver minst 4 diskar.

* Effektiv lagring: (n / 2) \* X \* 0,85
* Fördelar: Snabbt och säkert
* Nackdelar: Dyrt

##### **RAID-kombinationer (t.ex. RAID 50, RAID 55)**

I vissa fall bygger man RAID "i flera lager". Exempelvis är RAID 50 en kombination av RAID 5 och RAID 0, vilket kan ge högre prestanda men fortfarande ha viss redundans. En typisk konfiguration kan vara två RAID 5-grupper med vardera fyra diskar (8 totalt), som sedan sätts ihop i RAID 0. Då används totalt två diskar till paritet, och sex diskar är tillgängliga för lagring. RAID 55 är två RAID 5-arrayer kopplade i spegling (ovanligt, men möjligt) vilket ger hög redundans men kräver många diskar. I vissa fall bygger man RAID "i flera lager". Exempelvis är RAID 50 en kombination av RAID 5 och RAID 0, vilket kan ge högre prestanda men fortfarande ha viss redundans. RAID 55 är två RAID 5-array kopplade i spegling (ovanligt, men möjligt).

🔹 **Tips!** RAID är inte en backup. Det skyddar mot diskkrasch, inte mot radering, virus eller brand.

#### **4.10.4 Vad är ZFS?**

**ZFS** är ett avancerat filsystem som också hanterar volymer och redundans, likt RAID. Det utvecklades av Sun Microsystems och används ofta i servrar och NAS:ar. En av de största fördelarna med ZFS är att det kan ersätta både traditionell RAID och LVM i ett och samma system.

ZFS använder "pools" och självreparerande block. Det har inbyggd checksumming, snapshot-funktioner och automatisk korrigering av bitfel.

Vanliga konfigurationer är:

* **ZFS RAID-Z1** = likt RAID 5
* **ZFS RAID-Z2** = likt RAID 6
* **ZFS RAID-Z3** = klarar tre diskkrascher

Fördelar:

* Självreparerande
* Snapshot-möjligheter
* Skalbart och flexibelt
* Kan ersätta både RAID och LVM

Nackdelar:

* Resurskrävande (RAM!)
* Komplicerat att sätta upp

#### **4.10.5 Vad är LVM?**

**LVM** står för *Logical Volume Manager*. Istället för att direkt skapa partitioner på en fysisk disk, lägger LVM till ett extra lager.

LVM består av:

* **Fysiska volymer (PV)** = t.ex. /dev/sda1
* **Volymgrupper (VG)** = flera PV bildar en VG
* **Logiska volymer (LV)** = själva partitionerna man monterar

🔹 **Metafor:** Tänk dig ett LEGO-set där du kan bygga om huset (disken) utan att riva allt. Med LVM kan du ändra storlek, lägga till mer lagring och flytta runt utan att ominstallera.

LVM används ofta tillsammans med RAID eller ZFS för att ge ännu större flexibilitet och kontroll över lagringen.

Fördelar:

* Flexibilitet
* Kan ändra storlek på diskar live
* Snapshot-funktioner

Nackdelar:

* Lite mer komplext

#### **4.10.6 Exempel från verkligheten**

* **Ubuntu Server**: Vid installation kan man välja LVM direkt. Kombinera med RAID via mdadm eller ZFS.
* **Proxmox**: Använder ofta ZFS som grund, för snapshots, replikering och säker lagring.
* **NAS-lösningar** (t.ex. TrueNAS): Använder ZFS som standard.

#### **4.10.7 Kontrollfrågor**

1. Vad står RAID för och vad är syftet?
2. Hur skiljer sig mjukvaru-RAID från hårdvaru-RAID?
3. Vad är för- och nackdelar med RAID 0?
4. Hur fungerar RAID 5 och vad händer om en disk går sönder?
5. Vad menas med att RAID inte är en backup?
6. Hur skiljer sig ZFS från traditionell RAID?
7. Vad är RAID-Z2?
8. Vad är LVM och vad är fördelen med att använda det?
9. Vad är en volymgrupp inom LVM?
10. Ge exempel på någon verklig situation där man använder ZFS.

#### **4.10.8 Fördjupningslänkar**

1. <https://raid.wiki.kernel.org/>
2. <https://help.ubuntu.com/community/Installation/SoftwareRAID>
3. <https://docs.proxmox.com/>
4. <https://openzfs.org/wiki/Main_Page>
5. <https://wiki.archlinux.org/title/LVM>

#### **4.10.9 Egna anteckningar**

...

# **5 Windows**

## **5.1 Historia**

Windows är idag världens mest spridda operativsystem för persondatorer. Men resan dit har varit lång och innehållsrik. För att förstå dagens Windows 11 (och även Windows Server) är det bra att ha en grundläggande bild av hur systemet har utvecklats.

### **5.1.1 Utveckling**

✅ **MS-DOS och tidiga Windows-versioner (1980-tal – början av 90-talet)**

* Windows började som ett grafiskt skal ovanpå MS-DOS.
* Windows 1.0, 2.0 och 3.x var inte egna operativsystem i egentlig mening, utan byggde på MS-DOS som kärna.
* Gränssnittet gjorde det enklare att starta program, hantera filer och jobba med fler fönster.

✅ **Windows 95 och Windows 98 (mitten/slutet av 90-talet)**

* Stort steg framåt: integrerade DOS och Windows i ett mer sammanhållet system.
* Introducerade Start-menyn och aktivitetsfältet – kännetecken som finns kvar än idag.
* Bättre stöd för 32-bitarsprogram, Plug and Play och nätverk.

✅ **Windows NT-linjen**

* Parallellt utvecklade Microsoft Windows NT (New Technology) som en stabil och säker kärna för företag.
* Windows NT 3.5, 4.0 och senare Windows 2000 byggde på en modern mikrokärnearkitektur.
* NT-linjen introducerade bättre användarhantering och filsystemet NTFS med avancerade rättigheter.

✅ **Windows XP (2001)**

* Enade konsument- och företagslinjen under NT-kärnan.
* Stabilare än 9x-serien och mycket populärt.
* Stöd för flera användare, fjärrskrivbord och bättre drivrutinshantering.
* Introducerade tydligare säkerhetsfunktioner.

✅ **Windows Vista (2006)**

* Kort nämnt: anses ofta som mindre lyckat.
* Införde dock viktiga tekniker som UAC (User Account Control) för säkerhet.

✅ **Windows 7 (2009)**

* Förbättrad version av Vista – mycket populärt och stabilt.
* Förbättrat gränssnitt, bättre drivrutinssupport.

✅ **Windows 8 och 8.1 (2012–2013)**

* Stort designomslag för att passa pekskärmar.
* Startmenyn ersattes av Startskärmen – något som var impopulärt hos många användare.
* Windows 8.1 förbättrade balansen mellan skrivbord och touch.

✅ **Windows 10 (2015)**

* Återintroducerade Startmenyn i förbättrad form.
* Designat som en “plattform” för ständiga uppdateringar.
* Inkluderade funktioner som Windows Defender, Cortana och den nya webbläsaren Edge.
* Första Windows som fick gratis uppdateringar på bred front.

✅ **Windows 11 (2021 och framåt)**

* Modernare gränssnitt med centrerad Startmeny.
* Fokus på design, säkerhet (t.ex. TPM-krav) och integration med molntjänster.
* Förbättrad hantering av virtuella skrivbord och fönsterlayout (Snap Layouts).
* Bygger fortfarande på NT-kärnan och har mycket gemensamt med Windows 10 under huven.

✅ **Windows Server**

* Viktigt att förstå att Windows Server-versionerna är **byggda på samma grund** som desktop-versionerna.
* Använder samma NT-kärna och liknande gränssnitt (exempel: Server 2008 R2 ≈ Windows 7, Server 2022 ≈ Windows 10/11).
* Anpassat för serverroller som AD (Active Directory), fil- och printtjänster, DNS/DHCP, webbserver (IIS) m.m.
* Eleverna kommer inte arbeta praktiskt med Server i årskurs 2, men ska känna till att det **i grunden är samma operativsystem** med fler funktioner och annan licensmodell.

### **5.1.2 Diskussions- och kontrollfrågor**

1. Vad var skillnaden mellan Windows 95 och tidigare versioner?
2. Varför skapade Microsoft en separat NT-linje?
3. Vad gjorde Windows XP så populärt?
4. Varför blev Windows 8 kritiserat?
5. Hur skiljer sig Windows 11 från Windows 10?
6. Vad är skillnaden mellan Windows Desktop och Server?

### **5.1.3 Fördjupande länkar**

* Microsoft – History of Windows  
   <https://learn.microsoft.com/en-us/windows/whats-new/windows-history>
* Wikipedia – History of Microsoft Windows  
   <https://en.wikipedia.org/wiki/History_of_Microsoft_Windows>
* How-To Geek – A Brief History of Windows  
   <https://www.howtogeek.com/721525/a-brief-history-of-microsoft-windows/>
* Microsoft – Windows Server Documentation (för den som vill se skillnader)  
   <https://learn.microsoft.com/en-us/windows-server/>
* Computer Hope – Microsoft Windows Versions  
   <https://www.computerhope.com/issues/ch000088.htm>

### **5.1.4 Egna anteckningar**

(Här kan du som elev skriva egna anteckningar om Windows historia, frågor eller reflektioner)

## 5.2 Uppbyggnad

Windows är ett modernt operativsystem med många komponenter som samarbetar för att ge en användarvänlig och flexibel plattform. Här går vi igenom några grundläggande byggstenar i Windows: enheter, filsystem, rättigheter (NTFS) och registret.

### **5.2.1 Enheter och enhetsbokstäver**

* Windows identifierar lagringsenheter med **enhetsbokstäver**, t.ex. C:, D:, E:.
* Systemdisken är oftast **C:** och innehåller Windows-installationen.
* USB-minnen, externa hårddiskar och optiska enheter får automatiskt nästa lediga bokstav.
* Du kan ändra enhetsbokstäver i **Diskhantering** (diskmgmt.msc).

### **5.2.2 Filsystem och NTFS**

* Det vanligaste filsystemet i Windows är **NTFS (New Technology File System)**.
* NTFS har många fördelar jämfört med äldre FAT32:
  + Stöd för stora filer och diskar.
  + Möjlighet till filkomprimering.
  + Journalföring (skydd mot datakorruption).
  + Avancerade **rättigheter och säkerhetsinställningar**.
* Alla moderna Windows-versioner (inkl. Server) använder NTFS som standard på systemdisken.

### **5.2.3 Rättigheter och behörigheter**

* NTFS-rättigheter styr **vem som får göra vad** med filer och mappar.
* Du kan ange rättigheter via **Egenskaper → Säkerhet** i Utforskaren.
* Exempel på rättigheter: Läs, Skriv, Ändra, Fullständig behörighet.
* Rättigheter kan **ärvas** från mappstrukturen – bra för att hantera stora filsystem.
* Administratörer kan alltid ta ägarskap och ändra behörigheter vid behov.
* Viktigt för **säkerhet** i både hem- och företagsmiljöer.

### **5.2.4 Registret**

* **Windows-registret** är en hierarkisk databas som lagrar inställningar för operativsystemet och installerade program.
* Består av nycklar och värden – ungefär som mappar och filer.
* Centralt för konfiguration av Windows, drivrutiner och applikationer.
* Kan redigeras med **Registereditorn (regedit)** – kraftfullt men riskabelt.
* Viktigt att vara försiktig: felaktiga ändringar kan göra systemet instabilt eller obrukbart.

### **5.2.5 Sammanfattning**

Windows bygger på en kombination av tydliga enhetsbokstäver, avancerat filsystem med säkerhetsfunktioner (NTFS), och ett centralt register som styr konfigurationen. Att förstå dessa delar är viktigt för både grundläggande användning och mer avancerad systemadministration.

### **5.2.6 Diskussions- och kontrollfrågor**

1. Vad är enhetsbokstäver i Windows och vad används de till?
2. Vad är fördelarna med att använda NTFS jämfört med FAT32?
3. Hur kan du ändra rättigheter för en fil eller mapp i Windows?
4. Vad är Windows-registret och vad används det till?
5. Varför ska man vara försiktig när man ändrar i registret?

### **5.2.7 Fördjupande länkar**

* Microsoft – File Systems in Windows  
   <https://learn.microsoft.com/en-us/windows/win32/fileio/file-systems>
* Microsoft – NTFS Overview  
   <https://learn.microsoft.com/en-us/windows/win32/fileio/ntfs-technical-reference>
* Microsoft – Change Permissions for Files and Folders  
   <https://support.microsoft.com/en-us/windows/change-permissions-for-files-and-folders-> in-windows-8bfef146-0b22-47f6-9f77-5ef9f4e6a9d4
* Microsoft – What is the Windows Registry?  
   <https://support.microsoft.com/en-us/windows/what-is-the-windows-registry-776707f4-66f0-7f80-0285-3c6441d04d46>
* Wikipedia – Windows Registry  
   <https://en.wikipedia.org/wiki/Windows_Registry>

### **5.2.8 Egna anteckningar**

(Här kan du som elev skriva egna anteckningar om Windows uppbyggnad, frågor eller reflektioner)

## 5.3 Fördelar och nackdelar med Windows som operativsystem

Windows är världens mest använda stationära operativsystem. Enligt olika uppskattningar (2024) har **Windows över 1,4 miljarder aktiva användare** globalt, vilket gör det till en mycket dominerande plattform för både privatpersoner och företag.

Att förstå styrkor och svagheter med Windows är viktigt både som användare och som blivande IT-tekniker.

### 5.3.1 Skillnader mellan Windows-versioner

Windows finns i flera olika utgåvor, anpassade för olika användargrupper. I denna kurs arbetar vi främst med Windows 11 Pro, eftersom den är vanlig i skol- och företagsmiljö. Men det är bra att känna till skillnaderna mellan Home, Pro och Enterprise.

Tabellen på nästa sida visar viktiga skillnader:

| Funktion / Egenskap | Home 🏠 | Pro 👨‍💻 | Enterprise 🏢 |
| --- | --- | --- | --- |
| Målgrupp | Privatpersoner | Företag / avancerade användare | Stora företag / organisationer |
| BitLocker (diskkryptering) | ❌ | ✅ | ✅ |
| Gruppolicy-editor (gpedit.msc) | ❌ | ✅ | ✅ |
| Fjärrskrivbord (Remote Desktop Server) | ❌ | ✅ | ✅ |
| Hyper-V (inbyggd virtualisering) | ❌ | ✅ | ✅ |
| Domänanslutning (AD / Azure AD) | ❌ | ✅ | ✅ |
| Windows Update for Business | ❌ | ✅ | ✅ |
| Windows Sandbox | ❌ | ✅ | ✅ |
| AppLocker och avancerad kontroll | ❌ | ❌ | ✅ |
| Microsoft Defender Credential Guard | ❌ | ❌ | ✅ |
| Priset (ca) | Lägst | Medel | Högst (volymlicens) |

💡 Att tänka på:

* Home-versionen fungerar för hemmabruk men saknar funktioner som krävs i en IT-miljö.
* Pro är den vanligaste versionen i utbildning, företag och labbmiljöer.
* Enterprise används främst i större organisationer och hanteras via volymlicens och fjärradministration.

### **5.3.2 Fördelar**

✅ **Användarvänligt gränssnitt**

* Kända menyer, Start-knappen, och grafiskt skrivbord.
* Lätt att komma igång även för nybörjare.

✅ **Brett programstöd**

* Stort utbud av programvara, spel och affärsapplikationer.
* Många leverantörer utvecklar främst för Windows.

✅ **Hårdvarukompatibilitet**

* Stöd för en mängd olika tillverkare och drivrutiner.
* Enklare att få hårdvara att “bara fungera” utan kompilering eller manuell konfiguration.

✅ **Backat av Microsoft**

* Långsiktig support, regelbundna uppdateringar.
* Microsoft 365-integration och molntjänster.

✅ **Professionella verktyg och serverfunktioner**

* Active Directory, Group Policy, Hyper-V m.m.
* Möjlighet att växa från hemmabruk till avancerade företagsmiljöer.

### **5.3.3 Nackdelar**

⚠️ **Licenskostnad**

* Inte gratis – licens krävs för varje dator.
* Serverlicenser ännu dyrare.

⚠️ **Stängd källkod**

* Inte möjligt att själv se eller modifiera systemets källkod.
* Användare är beroende av Microsofts utveckling och prioriteringar.

⚠️ **Resurskrävande**

* Nya versioner kan kräva kraftfullare hårdvara.
* Äldre datorer kan bli långsamma eller obrukbara.

⚠️ **Säkerhetsmål**

* På grund av sin stora spridning är Windows en vanlig måltavla för virus och malware.
* Kräver aktivt underhåll som uppdateringar och antivirus.

⚠️ **Komplexitet i företagsmiljöer**

* Kan vara svåradministrerat i större nätverk utan rätt kunskap.
* Många olika versioner och licensmodeller att hålla reda på.

### **5.3.4 Sammanfattning**

Windows är ett kraftfullt, välkänt och flexibelt operativsystem med mycket brett stöd – både i mjukvara och hårdvara. Samtidigt har det en del utmaningar kring licenskostnad, säkerhet och resurshantering.

Att förstå dessa styrkor och svagheter är viktigt för att kunna välja rätt system för rätt behov – både för privat användning och professionella IT-lösningar.

### **5.3.5 Diskussions- och kontrollfrågor**

1. Varför är Windows så vanligt som operativsystem?
2. Nämn två fördelar med Windows som gör det lätt för nybörjare att använda.
3. Varför kan Windows bli en säkerhetsrisk om det inte underhålls?
4. Vad är en nackdel med att Windows är stängd källkod?
5. Varför kan licenskostnaden vara en nackdel för vissa användare?

### **5.3.6 Fördjupande länkar**

* Microsoft – Windows 11 Overview  
   <https://www.microsoft.com/en-us/windows/windows-11>
* Wikipedia – Usage share of operating systems  
   <https://en.wikipedia.org/wiki/Usage_share_of_operating_systems>
* Microsoft Learn – What is Windows?  
   <https://learn.microsoft.com/en-us/windows/whats-new/>
* How-To Geek – Pros and Cons of Windows vs Linux  
   <https://www.howtogeek.com/117635/windows-vs.-linux-which-operating-system-is-better/>
* TechRadar – Windows 11 review  
   <https://www.techradar.com/reviews/windows-11>

### **5.3.7 Egna anteckningar**

(Här kan du som elev skriva egna anteckningar om Windows fördelar och nackdelar, frågor eller reflektioner)

## 5.4 Installation av Windows 11 i fysisk maskin/VM

Att kunna installera Windows 11 är en grundläggande färdighet både för privatpersoner och IT-tekniker. Här går vi igenom **hur man laddar ner en ISO-fil, skapar ett startbart USB-minne, ändrar bootordning i BIOS/UEFI och installerar systemet**, både på fysisk hårdvara och i en virtuell maskin.

### 5.4.1 Förberedelser

✅ **Kontrollera systemkrav**

* Processor: 64-bitars, 1 GHz eller snabbare, minst 2 kärnor.
* RAM: Minst 4 GB (rekommenderat 8+ GB).
* Lagring: 64 GB eller mer.
* TPM 2.0 och Secure Boot (krav för Windows 11).

### 5.4.2 Ladda ner Windows 11 ISO

Microsoft tillhandahåller ett officiellt verktyg som heter **Media Creation Tool**.

✅ **Steg för steg:**

1. Gå till den officiella nedladdningssidan:  
    <https://www.microsoft.com/software-download/windows11>
2. Klicka på *Download now* under **Media Creation Tool**.
3. Kör verktyget när det laddats ner.
4. Välj *Create installation media (USB flash drive, DVD, or ISO file)*.
5. Välj språk och edition.
6. Välj att skapa **ISO-fil** och spara den på datorn.
   1. Alternativ: Skapa direkt USB om du redan har en sticka ansluten.

✅ För VM-installation räcker det att ha ISO-filen sparad lokalt.

### 5.4.3 Skapa en startbar USB-sticka

Om du ska installera på en **fysisk dator** behöver du göra ett USB-minne startbart.

✅ Media Creation Tool kan göra detta åt dig direkt:

* Välj *USB flash drive* istället för ISO i steg 6 ovan.
* Välj din USB-enhet (minst 8 GB rekommenderas).
* Verktyget laddar ner och skriver filerna automatiskt.

✅ Alternativt kan du använda **Rufus**:

1. Hämta Rufus: [https://rufus.ie](https://rufus.ie/)
2. Starta programmet.
3. Välj din USB-enhet.
4. Välj din ISO-fil.
5. Kontrollera partitionstyp (GPT för UEFI).
6. Klicka *Start*.

### 5.4.4 Ändra bootordning i BIOS/UEFI

För att starta från USB behöver du ofta ändra bootordning:

✅ **Steg för steg:**

1. Starta om datorn.
2. Tryck rätt tangent för BIOS/UEFI (vanligtvis Del, F2, F10 eller Esc – står oftast på skärmen).
3. Leta upp *Boot Order* eller *Boot Priority*.
4. Sätt USB-enheten högst upp i listan.
5. Spara och avsluta (Save & Exit).
6. Datorn startar om från USB-stickan.

### 5.4.5 Installera Windows 11

✅ **Steg för steg-installation:**

1. Välj språk, tid och tangentbord.
2. Klicka på *Install now*.
3. Ange produktnyckel eller välj *I don’t have a product key*.
4. Välj edition (Pro eller Home).
5. Acceptera licensvillkoren.
6. Välj *Custom: Install Windows only (advanced)*.
7. Välj/Skapa partition (ta bort gamla om du vill rensa disken helt).
8. Installationen körs automatiskt och datorn startar om flera gånger.
9. Konfigurera språk, region och tangentbord efter första uppstart.

### 5.4.6 Skapa lokalt konto på Windows 11 (offline)

För att slippa Microsoft-konto kan du göra så här vid kontoinställning:

✅ **Steg för steg:**

1. När du kommer till inloggningssteget – koppla ur nätverkskabel eller inaktivera Wi-Fi.
2. Windows försöker tvinga onlinekonto men erbjuder alternativ för offlinekonto/offlinekonto eller “Limited Experience”.
3. Om du inte ser alternativet:
   1. Tryck **Shift + F10** för att öppna kommandoprompt.
   2. Skriv: start ms-cxh:localonly
   3. Tryck **Enter**.
4. Skärmen för lokalt konto visas.
5. Ange användarnamn och lösenord.

✅ Detta fungerar även i VM om du "stänger av nätverk" under installationen.

### 5.4.7 Installation i virtuell maskin

Att installera Windows 11 i en virtuell maskin (t.ex. VirtualBox eller VMware) är snarlikt:

✅ **Steg för steg:**

1. Skapa en ny VM med typ **Windows 11**.
2. Tilldela minst 4 GB RAM och 64 GB disk (rekommenderat mer för bättre prestanda).
3. Montera ISO-filen som optisk enhet i VM-inställningarna.
4. Starta VM – installationsprogrammet startar direkt.
5. Följ samma installationssteg som på fysisk maskin.
6. För offlinekonto – koppla ur nätverkskortet i VM-inställningarna eller använd Shift + F10-tricket.

### 5.4.8 Diskussions- och kontrollfrågor

1. Vilket verktyg tillhandahåller Microsoft för att ladda ner och skapa installationsmedia?
2. Vad behöver du göra i BIOS/UEFI för att starta från USB?
3. Varför behöver man ibland ändra bootordning?
4. Hur kan man skapa ett lokalt offlinekonto på Windows 11?
5. Vad är en fördel med att installera i en virtuell maskin jämfört med på fysisk hårdvara?

### 5.4.9 Egna anteckningar

(Här kan du som elev skriva egna anteckningar om installationen, kommandon, eller saker du vill komma ihåg)

## 5.5 Konfiguration och hantering av Windows

När Windows är installerat behöver du kunna hantera och konfigurera det på rätt sätt. Här går vi igenom grunderna i **att installera och avinstallera program, filhantering, användarhantering och rättigheter samt felsökning och underhåll**.

### 5.5.1 Installera och avinstallera program

✅ **Från webben**

* Ladda ner installationsfil (ofta .exe eller .msi).
* Kör filen som administratör vid behov.
* Följ installationsguiden.
* Programmet hamnar vanligtvis i *C:\Program Files* eller *C:\Program Files (x86)*.

✅ **Från Microsoft Store**

* Öppna **Microsoft Store**.
* Sök efter appen.
* Klicka *Installera*.
* Appar från Store är ofta enklare att uppdatera och hantera.

✅ **Avinstallera program**

* Gå till **Inställningar → Appar → Appar och funktioner**.
* Välj programmet och klicka *Avinstallera*.
* Alternativt via **Kontrollpanelen → Program och funktioner**.

### 5.5.2 Filhantering

✅ **Skapa mappar**

* Högerklicka i Utforskaren → *Ny mapp*.
* Namnge mappen direkt.

✅ **Flytta och kopiera**

* Dra och släpp mellan mappar eller diskar.
* Ctrl+C (kopiera), Ctrl+V (klistra in), Ctrl+X (klipp ut).
* Högerklicka → *Kopiera* eller *Klipp ut*, och *Klistra in*.

✅ **Sökvägar**

* Visar var filer finns på disken.
* Exempel:
  + *C:\Users\Namn\Documents*
  + *D:\Backup\Bilder*
* Viktigt att hålla ordning för att snabbt hitta filer.

✅ **Visa filändelser**

* Utforskaren → *Visa* → markera *Filnamnstillägg*.
* Bra för att se filtyper (.exe, .txt, .jpg).

### 5.5.3 Användarhantering, grupper och rättigheter (Windows 11 Pro)

#### Kontotyper

Windows har olika kontotyper som styr vad användaren får göra på datorn:

**Standardanvändare (User):**

* Kan använda installerade program.
* Kan ändra sina egna inställningar och filer.
* Kan inte installera eller ta bort program för alla användare.
* Kan inte ändra systeminställningar som påverkar andra.

**Administratör (Administrator):**

* Full behörighet att ändra alla inställningar.
* Kan installera och ta bort program för alla användare.
* Kan skapa eller ta bort användarkonton.
* Kan ändra säkerhetsinställningar och köra alla program som administratör.

✅ Viktigt: För att skydda systemet bör **vanliga användare inte alltid köra som administratör**.

#### Grupper och hur de påverkar rättigheter

Windows använder **grupper** för att förenkla hantering av behörigheter. En grupp är en samling användare med samma rättigheter.

Vanliga grupper:

* **Administrators** – fullständig kontroll över datorn.
* **Users** – standardanvändare med begränsad åtkomst.
* **Guests** – mycket begränsad åtkomst för tillfälliga användare.

✅ Exempel:  
 Om du lägger en användare i gruppen *Administrators* får hen automatiskt alla administratörsrättigheter.

✅ Hantering av grupper och konton:

* **Inställningar → Konton → Andra användare** – för enkel hantering.
* **Lokal användar- och grupphantering (lusrmgr.msc)** – i Pro-versioner för att mer avancerat skapa, ändra eller ta bort konton och grupper.

✅ Viktigt att förstå:  
 Grupper används för att **enkelt tilldela rättigheter** till flera användare utan att behöva sätta rättigheter individuellt på varje användare.

#### Rättigheter i filsystemet (NTFS)

Windows använder **NTFS (New Technology File System)** för att hantera rättigheter på filer och mappar. Det låter dig styra exakt vad olika användare eller grupper får göra.

✅ **Exempel på rättigheter:**

* **Läsa:** Öppna filer, visa innehåll.
* **Skriva:** Skapa eller ändra filer.
* **Ändra:** Kombinerar läsa, skriva och ta bort.
* **Fullständig behörighet:** Alla ovanstående + ändra ägare och rättigheter.

✅ **Ärvda rättigheter**

* Som standard **ärver** nya filer och mappar rättigheter från den mapp de ligger i.
* Det gör det enklare att hantera behörighet i hela mappstrukturer.
* Du kan bryta ärvning om du behöver anpassa rättigheterna för en specifik fil eller mapp.

✅ **Hantera rättigheter**

* Högerklicka på fil eller mapp → *Egenskaper* → *Säkerhet*.
* Här kan du se och ändra vilka användare och grupper som har rättigheter.
* Under *Avancerat* kan du se ärvda rättigheter och ändra ägarskap.

✅ **Viktigt att veta:**

* Administratörer kan alltid ta ägarskap över filer eller mappar, även om de saknar andra rättigheter.
* Att förstå och använda rättigheter korrekt är en viktig del av säkerheten både hemma och i företagsmiljö.

### 5.5.4 Felsökning och underhåll

Windows har många inbyggda verktyg för att hålla systemet i gott skick. Här är viktiga exempel – **med hur du hittar dem och vad de gör**:

✅ **Standardprogram (Default apps)**

* **Hur hittar du det?**
  + Inställningar → Appar → Standardappar.
* **Vad gör den?**
  + Styr vilket program som öppnar olika filtyper, t.ex. webbsidor eller bilder.

✅ **Diskrensning (Disk Cleanup)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Diskrensning*.
* **Vad gör den?**
  + Tar bort tillfälliga filer och frigör utrymme på disken.

✅ **chkdsk**

* **Hur hittar du det?**
  + Kommandoprompt som administratör → skriv chkdsk.
* **Viktiga växlar:**
  + /f (fixar fel), /r (letar dåliga sektorer och återställer data).
* **Vad gör den?**
  + Kontrollerar och reparerar filsystemfel på disken.

✅ **Defragmentera / Optimize Drives**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Defragmentera* eller *Optimize Drives*.
* **Vad gör den?**
  + Omorganiserar filer på hårddisken för snabbare åtkomst (SSD använder trim).

✅ **Felsökning (Troubleshoot)**

* **Hur hittar du det?**
  + Inställningar → System → Felsökning.
* **Vad gör den?**
  + Guidade lösningar för vanliga problem (ljud, nätverk, uppdateringar).

✅ **Kompatibilitetsläge (Compatibility settings)**

* **Hur hittar du det?**
  + Högerklick på program → *Egenskaper* → *Kompatibilitet*.
* **Vad gör den?**
  + Gör att äldre program kan köras som om de vore på tidigare Windows-versioner.

✅ **Tillförlitlighetshistorik (Reliability Monitor)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Tillförlitlighetshistorik*.
* **Vad gör den?**
  + Visar stabilitetsproblem, krascher och ger lösningsförslag.

✅ **Aktivitetshanteraren (Task Manager)**

* **Hur hittar du det?**
  + Ctrl+Shift+Esc eller högerklick på aktivitetsfältet.
* **Vad gör den?**
  + Visar aktiva program, processer och resursanvändning.

✅ **Resursövervakaren (Resource Monitor)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Resursövervakaren*.
* **Vad gör den?**
  + Ger detaljerad information om CPU, minne, disk och nätverk.

✅ **Msconfig / Systemkonfiguration**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Systemkonfiguration* eller kör msconfig.
* **Vad gör den?**
  + Hanterar uppstartsalternativ, tjänster och boot-inställningar.

✅ **Minnesdiagnostik**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Windows Memory Diagnostic*.
* **Vad gör den?**
  + Testar datorns RAM-minne för fel.

✅ **Lokal säkerhetsprincip (Local Security Policy)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Local Security Policy* (Pro/Enterprise).
* **Vad gör den?**
  + Hanterar säkerhetsregler som lösenordspolicyer och användarrättigheter.

✅ **Fjärrskrivbord (Remote Desktop)**

* **Hur hittar du det?**
  + Inställningar → System → Fjärrskrivbord.
* **Vad gör den?**
  + Gör att du kan fjärransluta till datorn via RDP.

✅ **Registereditorn**

* **Hur hittar du det?**
  + Sök i Startmenyn: *regedit*.
* **Vad gör den?**
  + Visar och låter dig redigera Windows-registret (avancerat).

✅ **Enhetshanteraren (Device Manager)**

* **Hur hittar du det?**
  + Högerklick på Start → *Enhetshanteraren*.
* **Vad gör den?**
  + Visar och hanterar drivrutiner och anslutna enheter.

✅ **Schemaläggaren (Task Scheduler)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Schemaläggaren*.
* **Vad gör den?**
  + Skapa och hantera automatiska uppgifter.

✅ **diskpart**

* **Hur hittar du det?**
  + Öppna Kommandoprompt som administratör → skriv diskpart.
* **Vad gör den?**
  + Avancerat verktyg för att hantera diskar och partitioner.

✅ **Utskriftshantering (Print Management)**

* **Hur hittar du det?**
  + Sök i Startmenyn: *Print Management* (Pro/Enterprise).
* **Vad gör den?**
  + Hanterar skrivare och utskriftsköer.

✅ **Hjälpmedel (Accessibility)**

* **Hur hittar du det?**
  + Inställningar → Hjälpmedel.
* **Vad gör den?**
  + Anpassar datorn för olika funktionshinder (förstoringsglas, skärmläsare, högkontrast).

### 5.5.5 Diskussions- och kontrollfrågor

1. Vad är skillnaden mellan en standardanvändare och en administratör?
2. Hur används grupper för att styra rättigheter i Windows?
3. Vad betyder att rättigheter ärvs i en mappstruktur?
4. Hur kan du kontrollera och ändra rättigheter på en fil eller mapp?
5. Ge exempel på tre Windows-verktyg du kan använda för att felsöka eller underhålla datorn.

### 5.5.6 Egna anteckningar

(Här kan du som elev skriva egna anteckningar om användarhantering, rättigheter eller felsökning i Windows)

# **6 Windows Server**

## **6.1 Installation av Windows Server**

Att installera Windows Server skiljer sig inte mycket från installationen av Windows 11, men det finns några viktiga skillnader att känna till. Framför allt finns det olika editioner och möjligheten att installera med eller utan grafiskt gränssnitt (GUI).

I denna instruktion utgår vi ifrån att du redan har en ISO-fil tillgänglig.

### **6.1.1 Skapa installationsmedia med Rufus**

För att installera Windows Server på en fysisk dator behöver du ett bootbart USB-minne.

1. **Ladda ner och starta Rufus** (<https://rufus.ie/>)
2. Välj din USB-enhet under *Device*
3. Klicka på *SELECT* och välj ISO-filen för Windows Server
4. Partition scheme: MBR (för äldre BIOS) eller GPT (för UEFI)
5. Klicka på *START*
6. När klart – starta om datorn och ändra bootordning i BIOS/UEFI

### **6.1.2 Installera i en virtuell miljö**

Om du använder t.ex. **Proxmox, VirtualBox eller Hyper-V**:

1. Skapa en ny virtuell maskin
2. Tilldela RAM (minst 4 GB) och lagringsutrymme (minst 40 GB)
3. Montera ISO-filen som virtuell CD/DVD
4. Starta upp och följ installationsguiden

### **6.1.3 Editioner: Standard, Datacenter och Core/Desktop Experience**

Vid installation kommer du att få välja mellan olika versioner. Här är en kort förklaring:

|  |  |  |
| --- | --- | --- |
| **Version** | **GUI** | **Användningsområde** |
| **Standard** | Ja/Nej | Vanligaste versionen, passar för de flesta servrar |
| **Datacenter** | Ja/Nej | För mycket stora organisationer, stöder fler VM:ar |
| **Core (utan GUI)** | Nej | Kommandoradsbaserad, används i större servermiljöer |
| **Desktop Experience** | Ja | Fullt GUI – rekommenderas för nybörjare och elever |

✅ *Välj alltid* **Desktop Experience** om du inte är bekväm med kommandoraden ännu.

### **6.1.4 Steg-för-steg-installation**

När ISO är startad:

1. Välj språk, tangentbord och tid → Klicka *Nästa*
2. Klicka *Install now*
3. Välj rätt edition (t.ex. "Standard Desktop Experience")
4. Acceptera licensvillkor
5. Välj *Custom installation*
6. Skapa/markera en partition → *Next*
7. Vänta medan Windows installeras
8. Skapa administratörslösenord

🛠 Efter installationen bör du:

* Ge servern ett namn
* Ställa in IP-adress
* Köra Windows Update
* Eventuellt lägga till roller/funktioner via **Server Manager**

### **6.1.5 Systemkrav och rekommendationer**

**Minimikrav för Windows Server 2022:**

* 1.4 GHz 64-bitars processor
* 512 MB RAM (2 GB med Desktop Experience)
* 32 GB ledigt lagringsutrymme
* UEFI med Secure Boot (rekommenderas)
* Internetanslutning för aktivering/uppdateringar

**Rekommenderade krav för smidig drift:**

* 2+ GHz flertrådad CPU
* 8+ GB RAM
* 60+ GB SSD
* 1 Gbit nätverkskort

## **6.2 Workgroup, Domain och Active Directory**

För att förstå hur Windows Server används i nätverk är det viktigt att känna till tre centrala begrepp: **Workgroup**, **Domain** och **Active Directory (AD)**.

### **6.2.1 Vad är en Workgroup?**

En **Workgroup** är en enkel typ av nätverksstruktur där varje dator är oberoende och hanterar sina egna användare och resurser.

* Ingen central styrning – varje dator har sina egna användarkonton och lösenord
* Fildelning sker via delade mappar, skrivare och rättigheter per maskin
* Alla datorer måste vara i samma *workgroup*-namn för att hitta varandra enklare

🧠 En workgroup är som ett kollektiv – alla ansvarar för sig själva.

❗ **Viktigt att förstå:**  
 I en workgroup måste **datorn som delar ut resurser vara påslagen**. Om någon delar ut en skrivare eller en mapp från sin dator, så slutar den vara tillgänglig i nätverket om datorn stängs av.

🔧 **Exempel:**  
 Bob har delat ut sin skrivare i hemnätverket. Om Bob stänger av sin dator, kan ingen annan längre skriva ut via den skrivaren. Det finns **ingen central server som tar över delningen** – vilket är en av de största begränsningarna med workgroup jämfört med en domän.

### **6.2.2 Vad är en Domain?**

En **Domain** är en centraliserad nätverksmodell som styrs av en **domänkontrollant** (Domain Controller).

* Alla användarkonton och resurser hanteras centralt
* Logga in från vilken dator som helst med ett enda konto
* Kräver en server som kör **Active Directory Domain Services (AD DS)**
* Används i skolor, företag och organisationer med behov av central administration

💼 En domän fungerar som ett företag – du loggar in och får tillgång till det du är behörig till, var du än sitter.

### **6.2.3 Vad är Active Directory?**

**Active Directory (AD)** är tekniken bakom domäner i Windows-världen. Den tillhandahåller:

* En katalogtjänst över alla användare, datorer, grupper, skrivare etc.
* Möjlighet att styra policies (GPO – Group Policy Objects)
* Central hantering av konton och rättigheter
* Flexibilitet att organisera resurser i *Organizational Units (OU)*

🔐 AD gör det möjligt att sköta **säkerhet, åtkomst och struktur** i stora nätverk.

### **6.2.4 Sammanfattning: skillnader mellan Workgroup och Domain**

|  |  |  |
| --- | --- | --- |
| **Funktion** | **Workgroup** | **Domain** |
| Konto-hantering | Lokalt på varje dator | Centralt via domänkontrollant |
| Skalbarhet | Liten miljö (≤10 datorer) | Stor miljö (10+ datorer) |
| Inloggning | Lokalt konto | Domänkonto |
| Resurshantering | Manuellt | Centraliserat via AD |
| Säkerhetspolicies | Manuellt per dator | Centralt via Group Policy |
| Kräver ständig tillgänglighet? | Ja, resursdelande datorer måste vara igång | Nej, resurser styrs centralt av server |

### **6.2.5 När används vad?**

|  |  |
| --- | --- |
| **Scenario** | **Rekommendation** |
| Hemma eller mycket litet företag | Workgroup |
| Skola, företag eller organisation | Domain med Active Directory |
| Både lokala och fjärranslutna användare | Domain |

## **6.3 Diskussions- och kontrollfrågor**

1. Vad är skillnaden mellan Standard och Datacenter edition av Windows Server?
2. När bör du använda Desktop Experience istället för Core-installation?
3. Vad är syftet med en domän?
4. Hur skiljer sig en workgroup från en domain i användarhantering?
5. Vad används Active Directory till?

## **6.4 Fördjupningslänkar**

* Microsoft Learn – Windows Server Installation Overview  
   <https://learn.microsoft.com/en-us/windows-server/get-started/>
* Microsoft Docs – Compare Windows Server editions  
   <https://learn.microsoft.com/en-us/windows-server/get-started/editions-comparison>
* What is Active Directory? (JumpCloud)  
   <https://jumpcloud.com/blog/what-is-active-directory>
* Group Policy Overview  
   <https://learn.microsoft.com/en-us/windows-server/administration/windows-commands/group-policy>
* Workgroup vs Domain (How-To Geek)  
   <https://www.howtogeek.com/195305/whats-the-difference-between-a-domain-a-workgroup-and-a-homegroup/>

## **6.5 Egna anteckningar**

(Här kan du som elev skriva egna anteckningar om Windows Server-installation, skillnader mellan Workgroup och Domain eller vad Active Directory egentligen gör)

# 7. Linux – Historia, filosofi och ekosystem

## 7.1 Vad är Linux?

Linux är inte ett operativsystem i sig – det är en **kärna**, en central del av ett operativsystem som hanterar resurser som minne, CPU och hårddiskar. Linuxkärnan utvecklades av **Linus Torvalds** 1991 som ett hobbyprojekt medan han studerade vid Helsingfors universitet. Han ville skapa ett fritt alternativ till UNIX, som på den tiden var kraftfullt men ofta dyrt och stängt.

Linuxprojektet växte snabbt tack vare ett engagerat community och en öppen utvecklingsmodell. Med tiden började andra utvecklare bidra, och tillsammans med komponenter från projekt som **GNU** skapades kompletta operativsystem som kunde användas av både privatpersoner och företag.

Idag är Linux **ett av världens mest använda operativsystem** – inte för att det syns i butikshyllan, utan för att det körs **i bakgrunden överallt**. Du hittar Linuxkärnan i:

* **Android-telefoner**
* **Servrar och superdatorer**
* **Internet of Things (IoT)**
* **Molntjänster som Google Cloud och AWS**
* **Steam Deck och andra spelsystem**
* **Bilar, flygplan och smarta TV-apparater**

Linux finns **överallt**, men märks sällan.

### **7.1.2 Upphovsrätt, öppen källkod och programvarutyper**

För att förstå Linux på riktigt måste man förstå **open source-rörelsen**. Linux är mer än bara kod – det är en idé om att **kunskap ska vara fri och tillgänglig för alla**. Här är några centrala begrepp:

#### **Open Source**

Programvara med **öppen källkod** där vem som helst får läsa, ändra och distribuera koden. Det gör att tusentals personer kan bidra till samma projekt – som Linux – från hela världen.

#### **Free Software**

Inte “gratis” i första hand – utan **frihet**. Free Software Foundation, startad av **Richard Stallman**, förespråkar att användare ska ha rätten att använda, ändra och dela programvara. Ett klassiskt citat är:

“Think free as in free speech, not free beer.”

#### **Freeware**

Gratis programvara, men ofta **utan tillgång till källkoden**. Exempel: Skype, Spotify (gratisversion). Det är alltså inte samma sak som open source.

#### **Shareware**

Programvara som är gratis att testa, men sedan kräver betalning för full funktionalitet. Vanligt före internet-eran.

#### **Proprietär programvara**

**Proprietär (eller sluten) programvara** är motsatsen till open source. Här är källkoden hemlig, och användaren får inte ändra, kopiera eller distribuera den fritt.  
 Exempel på proprietär programvara är Microsoft Windows, Adobe Photoshop och drivrutiner från t.ex. NVIDIA.

De flesta Linuxdistributioner erbjuder möjligheten att installera vissa **proprietära drivrutiner** (som inte finns i den öppna källkoden) för att få bättre stöd för grafik, ljud och nätverk – särskilt viktigt om datorn innehåller NVIDIA-grafikkort eller vissa Wi-Fi-kretsar.

#### **Git**

Ett versionshanteringssystem skapat av Linus Torvalds (ironiskt nog, eftersom han inte gillade alternativen). Git används för att **spåra ändringar i kod** och samarbeta i stora projekt. Github är idag den största plattformen för Git-projekt.

#### **GNU**

Ett projekt som ville skapa ett helt fritt operativsystem. De lyckades nästan – men saknade en kärna. När Linuxkärnan kom, kunde GNU + Linux kombineras till ett helt fungerande system. Därför kallas Linux ibland för “GNU/Linux”.

Linuxvärlden bygger på **samarbete, delning och frihet** – både tekniskt och juridiskt.

### **7.1.3 Linuxkärnans grenar (distributioner)**

Eftersom Linux är **fri att använda och modifiera**, har det vuxit fram många olika “smaker” av Linux – så kallade **distributioner** eller “distros”.

De flesta distros bygger på **en av tre stora familjer**:

* **Debian** – stabil och populär, med barn som Ubuntu och Linux Mint. Vanlig i skolor, servrar och nybörjarsystem.
* **Red Hat** – affärsfokuserad med stöd och certifiering. Fedora är dess testplattform, medan CentOS och AlmaLinux är communityversioner.
* **Arch** – minimalistisk och avancerad. “DIY-Linux” där man bygger allt själv. Populär bland entusiaster.

Man kan se Linux som en **motor (kärnan)**, och varje distro som en **bilmodell byggd kring motorn**, med olika inredning, reglage och målgrupper.

### **7.1.4 Fördelar med Linux**

* Gratis att använda **–** De flesta distributioner är helt fria att ladda ner och använda, vilket gör dem perfekta för skolor, hobbyister och utvecklare.
* Öppen källkod – Du kan granska, ändra och anpassa koden själv, vilket skapar insyn och möjlighet till innovation.
* Stabilt och säkert – Linux används ofta på servrar eftersom det sällan kraschar och har färre sårbarheter än vissa kommersiella alternativ.
* Flexibelt och anpassningsbart – Du kan välja precis vilken skrivbordsmiljö, mjukvara och konfiguration du vill ha – från minimalistiskt till fullt GUI.
* Stort community – Det finns tusentals guider, forum och hjälpsidor, vilket gör det lättare att få hjälp vid problem.

### **7.1.5 Nackdelar med Linux**

* Inte alla program finns tillgängliga **–** Vissa program (t.ex. Adobe Photoshop eller Microsoft Office) har inte fullständigt Linux-stöd,vilket kan vara ett problem för vissa användare.
* Kan vara svårt att komma igång för nybörjare – Kommandoraden kan kännas överväldigande, särskilt för den som är van vid Windows.
* Begränsat stöd från vissa tillverkare – Drivrutiner och hårdvarustöd kan ibland saknas, särskilt för nyare eller ovanlig utrustning.
* Spelstöd inte alltid lika bra – Även om det blivit mycket bättre med Steam och Proton, finns fortfarande vissa begränsningar.
* Fler steg vid installation och underhåll – Du förväntas ibland hantera saker manuellt, till exempel montering av diskar eller konfigurering av nätverk.

### **7.1.6 Fördjupningslänkar**

1. Linux Foundation – [https://www.linuxfoundation.org](https://www.linuxfoundation.org/)
2. History of Linux – <https://www.linux.com/what-is-linux/>
3. GNU Project – <https://www.gnu.org/>
4. What is Git – [https://git-scm.com](https://git-scm.com/)
5. Ubuntu – [https://ubuntu.com](https://ubuntu.com/)
6. Debian – [https://www.debian.org](https://www.debian.org/)
7. Red Hat – [https://www.redhat.com](https://www.redhat.com/)
8. Arch Linux – [https://www.archlinux.org](https://www.archlinux.org/)
9. Free Software Foundation – [https://www.fsf.org](https://www.fsf.org/)
10. Steam Deck (Linux-baserad) – [https://www.steamdeck.com](https://www.steamdeck.com/)

### **7.1.7 Kontroll- och diskussionsfrågor**

1. Vem skapade Linux och varför?
2. Vad är skillnaden mellan Linuxkärnan och ett operativsystem?
3. Nämn tre platser där Linux används idag utan att man tänker på det.
4. Förklara begreppet "free software".
5. Vad är skillnaden mellan open source och freeware?
6. Vad är GNU och varför är det viktigt i Linuxhistorien?
7. Beskriv skillnaden mellan Debian-, Red Hat- och Arch-familjerna.
8. Nämn två fördelar med att använda Linux.
9. Nämn två nackdelar med att använda Linux.
10. Vad är Git och varför är det viktigt inom öppen källkod?

### **7.1.8 Egna anteckningar**

*(Här lämnas plats för elevens egna reflektioner, tankar och eventuella kompletteringar under lektionen.)*

## **7.2 Installation av Ubuntu Desktop**

Ubuntu är idag **världens mest använda Linuxdistribution** för både privatpersoner och företag. Den är känd för att vara **användarvänlig, stabil och modern**, och används som grund för många andra Linux-varianter – som Linux Mint, Pop!\_OS och Elementary OS. Det gör att Ubuntu fungerar som en slags **"ledare"** inom Linuxvärlden när det gäller teknisk utveckling, paketsystem och stöd för ny hårdvara.

Bakom Ubuntu står **Canonical**, ett brittiskt företag och en stiftelse som ansvarar för utvecklingen. Canonical bidrar även till molntjänster, servrar och Internet of Things, men det är deras skrivbordsversion – Ubuntu Desktop – som ofta används i undervisning och som introduktion till Linux.

Ubuntu släpper **två versioner per år**:

* En i **april** (04)
* En i **september** (09)

Versionerna numreras enligt formatet **ÅR.MÅNAD**, t.ex. **24.04** = april 2024.

Det finns två typer av versioner:

* **LTS – Long Term Support**  
   En LTS-version ges ut vartannat år (jämna år i april: 20.04, 22.04, 24.04...)  
   Dessa har **5 års support**, vilket innebär säkerhetsuppdateringar, buggfixar och stabilitet över lång tid.  
   De används ofta i **företag, skolor och produktion** eftersom de är testade extra noggrant.
* **STS – Short Term Support**  
   Släpps mellan LTS-versionerna (t.ex. 23.10, 24.10) och har **9 månaders support**.  
   Dessa versioner innehåller ofta **nyare teknik och funktioner**, vilket gör dem intressanta för utvecklare och entusiaster som vill ligga i framkant – men de är inte lika långsiktigt stabila som LTS.

I denna guide använder vi **Ubuntu 24.04 LTS**, vilket är den version som rekommenderas för skolor, företag och nybörjare – tack vare dess stabilitet och långsiktiga support.

### **Viktigt om Secure Boot**

Ubuntu är en av få Linuxdistributioner som fungerar direkt med **Secure Boot** aktiverat. Däremot kräver vissa drivrutiner (t.ex. NVIDIA eller vissa Wi-Fi-kretsar) att man **signerar moduler** och registrerar en nyckel med hjälp av **MOK (Machine Owner Key)**.  
 För mer information om vad Secure Boot är, se **kapitel 2.7.1**.

### **Steg för steg: Installera Ubuntu Desktop**

#### **1. Ladda ner Ubuntu**

Besök <https://ubuntu.com/download/desktop> och ladda ner den senaste LTS-versionen (t.ex. *Ubuntu 24.04 LTS*).

#### **2. Skapa en startbar USB**

Använd ett verktyg som:

* **Rufus** (Windows)
* **Balena Etcher** (Windows/macOS/Linux)
* **Startup Disk Creator** (på annan Ubuntu)

Välj .iso-filen och skriv till ett USB-minne (minst 8 GB).

#### **3. Starta datorn från USB**

Starta om datorn och välj att boota från USB-enheten. Detta görs ofta via en särskild tangent (t.ex. F12, ESC, F2 beroende på tillverkare).

⚠️ På de flesta datorer behöver man **inte** stänga av Secure Boot – Ubuntu fungerar direkt med det aktiverat.

#### **4. Välj språk och tangentbord**

I installationsmenyn:

* Välj **språk**
* Välj **tangentbordslayout** (Svenskt tangentbord är standard i Sverige)

#### **5. Anslut till Wi-Fi eller nätverk**

Det är **rekommenderat att ha internetanslutning** under installationen. Då kan Ubuntu automatiskt installera drivrutiner och uppdateringar.

#### **6. Installationsalternativ**

Du får nu välja mellan:

* **Normal installation**: inkluderar webbläsare, kontorsprogram, mediauppspelare m.m.
* **Minimal installation**: endast webbläsare och grundläggande verktyg

Markera också:

☑ Install third-party software for graphics and Wi-Fi hardware  
☑ Download updates while installing Ubuntu

☝️ Den första rutan är viktig – den ser till att proprietära drivrutiner installeras, vilket kan kräva MOK efter omstart.

#### **7. Välj installationsläge**

* Välj “Erase disk and install Ubuntu” om du vill att Ubuntu använder hela hårddisken
* Eller “Something else” om du vill göra avancerad partitionering (rekommenderas endast för vana användare)

Ubuntu sköter automatiskt partitionering och skapar swap om inget annat anges.

#### **8. Skapa användare**

Ange:

* Fullständigt namn
* Datornamn
* Användarnamn
* Lösenord (används även för att bekräfta systemändringar via sudo)

#### **9. Installationen startar**

Ubuntu kopierar filer och konfigurerar systemet. Detta tar ca 5–15 minuter beroende på datorns hastighet.

### **Efter installationen: MOK och drivrutiner**

Vid första omstart kan du få upp ett **blått MOK-fönster** om du installerat proprietära drivrutiner (som kräver Secure Boot-signering). Gör då följande:

1. Välj “**Enroll MOK**”
2. Ange det **lösenord du valde** under installationen (du blir tillfrågad om det innan omstart)
3. Bekräfta och starta om

Detta registrerar Canonicals signeringsnyckel, så att drivrutiner kan laddas även med Secure Boot aktivt.

### **Justera drivrutiner i efterhand**

Om något inte fungerar (t.ex. Wi-Fi, grafik), kan du:

1. Gå till **Programvara och uppdateringar**
2. Välj fliken **Ytterligare drivrutiner**
3. Aktivera alternativ som “Proprietär, testad”
4. Bekräfta och starta om

### **7.2.1 Fördjupningslänkar**

1. Ubuntu Desktop Download – <https://ubuntu.com/download/desktop>
2. Canonical – [https://canonical.com](https://canonical.com/)
3. What is Secure Boot – <https://wiki.ubuntu.com/UEFI/SecureBoot>
4. Installing Proprietary Drivers – <https://help.ubuntu.com/community/BinaryDriverHowto>
5. Ubuntu LTS vs STS – <https://ubuntu.com/blog/ubuntu-release-cycle-explained>

### **7.2.2 Kontrollfrågor**

1. Vad är Ubuntu och varför är det en av de mest använda Linuxdistributionerna?
2. Vad heter organisationen som står bakom Ubuntu?
3. Hur döps Ubuntu-versioner och vad betyder till exempel “24.04”?
4. Vad innebär LTS och hur länge får en sådan version support?
5. Vad är skillnaden mellan en LTS-version och en STS-version?
6. Vilket verktyg kan du använda i Windows för att skapa ett startbart USB-minne?
7. Vad är Secure Boot, och varför behöver man känna till det vid Linuxinstallation?
8. Vad är MOK och när kan det behövas?
9. Varför bör man markera alternativet “Install third-party software…” under installationen?
10. Hur kan man aktivera proprietära drivrutiner i efterhand?

### **7.9.3 Egna anteckningar**

*(Här kan du som elev skriva ner egna tankar, saker du upptäckte under installationen eller frågor du vill ställa vid nästa lektion.)*

## **7.3 Konfiguration av Ubuntu Desktop**

Efter installationen är det dags att börja utforska Ubuntu och förstå hur systemet är uppbyggt och hanteras – både via terminalen och det grafiska gränssnittet. Till skillnad från Windows är Ubuntu **flexibelt**: nästan allt du kan göra i grafiskt gränssnitt kan också göras i terminalen – och vice versa.

Här går vi igenom grunderna i hur du håller systemet uppdaterat, installerar program, förstår filsystemet och hanterar rättigheter.

### **7.3.1 Uppdatera systemet**

Ubuntu hämtar program och uppdateringar från **repositories** – det är samlingar av godkända paket som är säkra att installera. Man kan jämföra det med en appbutik fast för hela systemet. När du uppdaterar Ubuntu kontaktar systemet dessa källor och laddar ner de senaste versionerna.

**Uppdatering via terminalen:**

sudo apt update  
sudo apt upgrade

**Uppdatering via grafiskt gränssnitt:**

* Öppna **Programuppdaterare** (Software Updater)
* Klicka på **Installera uppdateringar**

### **7.3.2 Installera och avinstallera program**

Ubuntu erbjuder flera sätt att installera och ta bort program. Oavsett om du föredrar GUI eller terminal kan du hantera systemet på dina egna villkor.

#### **Via terminalen:**

sudo apt install vlc  
sudo apt remove vlc

#### **Via Synaptic (pakethanterare):**

* Installera Synaptic om det inte finns:

sudo apt install synaptic

* Starta Synaptic och sök efter program, markera och installera

#### **Via Ubuntu Software Center:**

* Öppna **Program** (Software)
* Sök och installera appar med ett klick

🧠 **Viktigt att förstå:** Allt går att göra både i GUI och terminalen – **det är samma paket i bakgrunden.** Det som skiljer är **hur** du interagerar med systemet.

### **7.3.3 Filsystem i Linux**

Linux använder ett **filsystem med en gemensam rot (/)** – alla filer och enheter placeras någonstans under denna rot.

Här är några viktiga mappar:

|  |  |
| --- | --- |
| **Mapp** | **Beskrivning** |
| / | Rotkatalogen – allt börjar här. Alla andra filer och mappar hänger under denna. |
| /bin | Grundläggande kommandon som ls, cp, mv, cat. |
| /boot | Boot-filer för att starta systemet, inkl. kärnan och grub. |
| /dev | Virtuella filer som representerar hårdvara, t.ex. sda, tty, null. |
| /etc | Inställningsfiler och konfigurationer för system och program. |
| /home | Användarnas hemkataloger – t.ex. /home/robin. |
| /lib | Bibliotek som behövs av program i /bin och /sbin. |
| /media | Monteringspunkt för t.ex. USB-minnen. |
| /mnt | Tillfällig monteringspunkt för manuell användning. |
| /opt | Tilläggsprogram (t.ex. tredjepartsappar). |
| /proc | Information om systemet och processer. |
| /root | Root-användarens hemkatalog. |
| /run | Temporär information om systemets tillstånd. |
| /sbin | Systemkommandon för root. |
| /srv | Data som erbjuds via t.ex. webb- eller FTP-servrar. |
| /sys | Information om kernel och drivrutiner. |
| /tmp | Temporära filer som kan tas bort efter omstart. |
| /usr | Program, dokumentation och bibliotek. |
| /usr/bin | De flesta användarkommandon. |
| /usr/sbin | Systemverktyg för root. |
| /usr/local | Program som installeras manuellt. |
| /var | Variabel data som loggar, databaser m.m. |

🔧 **Till skillnad från Windows** används **inte enhetsbokstäver (C:, D:)**. Nya diskar **monteras in i det befintliga filsystemet**, t.ex. /media/usb.

### **7.3.4 Rättigheter och grupper i Linux**

Linux använder ett rättighetssystem som styr **vem som får göra vad med varje fil**. Systemet bygger på tre delar:

* **Ägare (User)** – användaren som äger filen
* **Grupp (Group)** – gruppen filen tillhör
* **Övriga (Others)** – alla andra användare

Varje fil har tre rättighetstyper:

* **r** – read (läsa)
* **w** – write (skriva)
* **x** – execute (köra)

Exempel på rättigheter:

-rw-r--r-- 1 robin itlärare myfil.txt

Det betyder:

* **rw-** (ägaren får läsa och skriva)
* **r--** (gruppen får läsa)
* **r--** (övriga får läsa)

#### **Oktal notation**

Rättigheter kan också skrivas i **siffraform**, så kallad oktal notation:

|  |  |  |
| --- | --- | --- |
| **Oktal** | **Rättighet** | **Förklaring** |
| 7 | rwx | läsa, skriva, köra |
| 6 | rw- | läsa, skriva |
| 5 | r-x | läsa, köra |
| 4 | r-- | endast läsa |
| 0 | --- | inga rättigheter |

Exempel:

chmod 755 script.sh

Betyder:

* Ägare: rwx (7)
* Grupp: r-x (5)
* Övriga: r-x (5)

#### **Grupper i Linux**

Användare kan tillhöra en **primär grupp** och flera **sekundära grupper**.

* Den primära gruppen skapas ofta automatiskt vid kontoskapande.
* Sekundära grupper kan användas för att ge särskilda rättigheter (t.ex. sudo, docker, www-data).

Kommandon:

groups robin  
sudo usermod -aG docker robin

🧠 Genom att använda grupper kan du ge **många användare tillgång till samma resurser**, utan att behöva ge dem root-rättigheter.

## **7.3.5 Felsökning och underhåll**

Precis som i Windows uppstår ibland problem i Ubuntu – det kan handla om nätverk, drivrutiner, program som inte startar eller uppdateringar som krånglar. Lyckligtvis erbjuder Ubuntu en mängd verktyg för att felsöka och lösa dessa problem – både grafiskt och via terminal.

Här är några vanliga områden och metoder att känna till:

#### **7.3.5.1 Uppdateringsproblem**

Om uppdateringar misslyckas eller låser sig:

sudo apt update  
sudo apt upgrade  
sudo apt --fix-broken install

Tips: Använd sudo apt autoremove för att rensa bort gamla paket du inte längre behöver.

#### **7.3.5.2 Nätverksproblem**

För att kontrollera nätverksstatus:

ip a  
ping 8.8.8.8  
nmcli device status

Om nätverket inte fungerar:

* Kontrollera om du är ansluten till rätt nätverk
* Starta om nätverkstjänsten:

sudo systemctl restart NetworkManager

#### **7.3.5.3 Program kraschar eller vägrar starta**

Du kan starta program från terminalen för att se felmeddelanden, t.ex.:

firefox

Om det inte fungerar, prova att installera om:

sudo apt remove firefox  
sudo apt install firefox

#### **7.3.5.4 Systemloggar**

Loggfiler hjälper dig att se vad som gått fel:

* **Grafiskt:** Loggboken (Logs) från programmenyn
* **Terminalen:**

journalctl -xe  
dmesg  
tail -f /var/log/syslog

#### **7.3.5.5 Användbara verktyg för underhåll**

|  |  |
| --- | --- |
| **Verktyg** | **Beskrivning** |
| top / htop | Visar CPU-, RAM- och processanvändning |
| df -h | Visar diskutrymme |
| free -h | Visar minnesanvändning |
| systemctl | Starta/stoppa/titta på systemtjänster |
| ubuntu-drivers | Hantera drivrutiner från terminal |

#### **7.3.5.6 Återställa grafiskt gränssnitt (vid GUI-problem)**

Ibland kraschar skrivbordsmiljön (t.ex. GNOME). Då kan du starta om den:

sudo systemctl restart gdm3

#### **7.3.5.7 Skapa systemrapport (t.ex. vid support)**

Ubuntu har ett verktyg som samlar systeminformation:

sudo ubuntu-report

#### **7.3.5.8 Säkerhetskopiering och återställning**

Använd gärna **Déjà Dup** (Backups) – det är ett inbyggt verktyg i Ubuntu.

Starta det via “Backuper” och ställ in:

* Vad som ska sparas
* Var (t.ex. USB, nätverksenhet)
* Hur ofta

### **7.3.6 Kontrollfrågor**

1. Vad är ett repository och varför behövs det?
2. Nämn tre sätt att installera program i Ubuntu.
3. Vad är skillnaden mellan /etc och /home?
4. Hur monteras en USB-sticka in i Linuxfilsystemet?
5. Vad gör kommandot chmod 644 fil.txt?
6. Vad betyder -rw-r--r-- i rättighetssträngen?
7. Hur lägger du till en användare i en grupp?
8. Hur kan du starta om nätverksanslutningen i Ubuntu?
9. Vad gör journalctl -xe och när är det användbart?
10. Vilket grafiskt verktyg kan du använda för backup i Ubuntu?

### **7.3.7 Fördjupningslänkar**

1. Ubuntu – Basic Troubleshooting Guide  
   <https://help.ubuntu.com/community/LinuxQuestionsHowTo>
2. How-To Geek – 8 Deadly Commands You Should Never Run on Linux  
   <https://www.howtogeek.com/125157/8-deadly-commands-you-should-never-run-on-linux/>
3. Linux Handbook – How to Check Logs in Linux  
   <https://linuxhandbook.com/view-log-files-linux/>
4. DigitalOcean – How to Use fsck to Check and Repair File System Issues  
   <https://www.digitalocean.com/community/tutorials/fsck-command-linux>
5. Linuxize – How to Use the top Command  
   <https://linuxize.com/post/linux-top-command/>

### **7.3.8 Egna anteckningar**

*(Här kan du skriva ner egna kommandon, lösningar, tips eller saker du vill komma ihåg.)*

## **7.4 Installation av Ubuntu Server och Webmin**

Ubuntu Server är ett resurssnålt, textbaserat operativsystem som används för att driva allt från webbservrar till filservrar och databaser. Till skillnad från Ubuntu Desktop innehåller serverversionen **inget grafiskt användargränssnitt**, vilket gör den snabbare och mer effektiv – särskilt i virtuella miljöer.

Att installera Ubuntu Server är inte svårt, men kräver att man är noggrann med inställningarna under installationen, eftersom valet av tjänster och nätverkskonfiguration får direkt påverkan på hur servern fungerar.

### **Steg för steg: Installera Ubuntu Server**

#### **1. Ladda ner Ubuntu Server**

Besök <https://ubuntu.com/download/server> och ladda ner den senaste LTS-versionen (t.ex. *Ubuntu Server 24.04 LTS*).

#### **2. Skapa en startbar USB**

Använd till exempel:

* **Rufus** (Windows)
* **Balena Etcher**
* **Startup Disk Creator** (på Linux)

Skriv .iso-filen till ett USB-minne på minst 4 GB.

#### **3. Starta datorn från USB**

Starta om och välj att boota från USB-enheten. Detta görs vanligtvis via F12, F2, ESC eller DEL beroende på tillverkare.

#### **4. Välj språk, tangentbord och tidszon**

#### **5. Nätverksinställningar**

Om du har en DHCP-server (t.ex. i Proxmox eller OPNsense) så får servern IP automatiskt.  
 Annars kan du ställa in en **statisk IP-adress** direkt.

#### **6. Användare och lösenord**

Skapa ett användarnamn och välj ett starkt lösenord.  
 Du får även möjlighet att installera **OpenSSH-server**, vilket rekommenderas så att du kan ansluta till servern via SSH i efterhand.

#### **7. Lagring och partitionering**

Välj:

* **Use an entire disk** (om du vill låta Ubuntu automatiskt partitionera)
* **Custom storage layout** (om du vill lägga upp partitioner själv)

#### **8. Installera valfria tjänster**

Ubuntu Server ger möjlighet att installera färdiga paketgrupper:

* OpenSSH
* Samba
* LAMP/LEMP
* Docker
* Print server
* etc.

I skolsammanhang är det oftast bäst att **börja rent och installera det du behöver i efterhand**.

#### **9. Starta installationen**

Installationen tar 5–10 minuter.

### **Webmin – ett grafiskt gränssnitt för serversystem**

Att jobba i terminalen är kraftfullt – men kan vara avskräckande för nya användare. Här kommer **Webmin** in i bilden. Det är ett **webbaserat gränssnitt för att administrera Linux-servrar**, där du enkelt kan hantera:

* Användare och grupper
* Brandvägg och nätverksinställningar
* Paketinstallationer
* Tjänster som SSH, Samba, FTP och mycket mer

Webmin gör det **enklare att förstå hur en server fungerar**, eftersom du ser helheten – inte bara kommandon. Det är ett utmärkt sätt att kombinera **pedagogik med riktig systemadministration**.

### **Så här installerar du Webmin på Ubuntu Server**

Webmin har numera ett **officiellt installationsscript** som automatiserar allt du behöver:

curl -o webmin-setup-repo.sh https://raw.githubusercontent.com/webmin/webmin/master/webmin-setup-repo.sh  
sh webmin-setup-repo.sh

Efter installationen kommer Webmin att vara tillgängligt på:

https://<serverns-ip>:10000

⚠️ Webmin använder HTTPS med ett självsignerat certifikat, så webbläsaren kommer att varna – det är normalt.  
 Logga in med det användarnamn och lösenord du skapade under installationen av Ubuntu Server.

### **7.4.1 Fördjupningslänkar**

1. Ubuntu Server Download – <https://ubuntu.com/download/server>
2. Ubuntu Server Guide (official) – <https://ubuntu.com/server/docs>
3. Webmin – [https://www.webmin.com](https://www.webmin.com/)
4. Webmin GitHub Repository – <https://github.com/webmin/webmin>
5. Install Webmin on Ubuntu – <https://computingforgeeks.com/how-to-install-webmin-on-ubuntu/>

### **7.4.2 Kontrollfrågor**

1. Vad är skillnaden mellan Ubuntu Desktop och Ubuntu Server?
2. Vad är fördelen med att Ubuntu Server saknar grafiskt gränssnitt?
3. Vad kan du göra under installationen om servern inte får någon IP-adress automatiskt?
4. Vilken typ av användare och lösenord skapar du under installationen?
5. Vad används OpenSSH till och varför bör det installeras på en server?
6. Varför rekommenderas det att börja med en "ren" installation utan extra tjänster?
7. Vad är Webmin och varför kan det vara ett bra verktyg i en undervisningsmiljö?
8. Hur når du Webmin när det är installerat?
9. Vad betyder det att Webmin använder ett självsignerat certifikat?
10. Vilka två kommandon används för att installera Webmin med det officiella installationsscriptet?

### **7.4.3 Egna anteckningar**

*(Här kan du skriva ner vad du lärde dig under installationen, vilka val du gjorde, eller hur det kändes att arbeta med servern via Webmin istället för terminalen.)*

# 8 OSI-modellen

## 8.1 Uppkomst – Bakgrunden till OSI

När datornätverk började växa fram på 1970-talet fanns det ett stort problem: olika tillverkare och organisationer skapade sina egna sätt att kommunicera mellan datorer. Det fanns inga gemensamma regler eller standarder. En dator från IBM kunde inte självklart prata med en från DEC, och alla skapade sina egna protokoll. Det blev snabbt rörigt.

För att lösa detta började man på 1970-talet att diskutera behovet av en gemensam modell som alla kunde utgå från. Organisationen **ISO (International Organization for Standardization)** tog initiativet, och 1984 publicerade de vad som kom att kallas **OSI-modellen** – en förkortning för **Open Systems Interconnection**. Modellen skulle inte vara ett faktiskt protokoll eller mjukvara, utan en **teoretisk referensmodell**. Den delade upp nätverkskommunikation i **sju lager**, där varje lager ansvarar för en specifik del av kommunikationen.

## 8.2 Syfte – Varför finns OSI-modellen?

OSI-modellen har flera syften, men det främsta är att:

1. **Standardisera kommunikationen** mellan olika system och tillverkare
2. **Förenkla felsökning och nätverksdesign** genom att tydligt separera olika funktioner
3. **Underlätta utveckling** av nätverksprotokoll, eftersom utvecklare kan fokusera på ett lager i taget

Modellen fungerar som ett **gemensamt språk**. Den hjälper tekniker att beskriva vad som händer i ett nätverk utan att behöva prata om specifika produkter. I stället för att säga "fel på filöverföring via TCP", kan man säga "problemet ligger troligen i transportlagret".

Den ger också en **modulär struktur** – om ett lager uppdateras (exempelvis ett nytt protokoll), behöver inte övriga lager påverkas. Det möjliggör en mer flexibel och framtidssäker nätverksdesign.

## 8.3 Användningsområde – Hur används OSI-modellen?

Trots att OSI-modellen är **teoretisk**, används den dagligen i IT-branschen. Den är ett **verktyg för felsökning, utbildning och analys**. Exempel på användningsområden är:

* **Felsökning av nätverk**: En tekniker kan börja analysera var ett fel uppstår – är det i applikationen (lager 7), i nätverket (lager 3), eller i kabeln (lager 1)?
* **Kommunikation mellan IT-personal**: Modellen ger en gemensam struktur som gör det lättare att beskriva och förstå problem.
* **Utbildning**: OSI används som pedagogiskt verktyg för att förklara hur data rör sig genom ett nätverk.
* **Utveckling av protokoll och system**: När man designar nya lösningar kan man definiera exakt vilket lager man jobbar med (exempelvis HTTPS som verkar på lager 7).

#### **De sju lagren i OSI-modellen:**

|  |  |  |
| --- | --- | --- |
| **Lager** | **Namn** | **Funktion** |
| 7 | Applikation | Här sker interaktionen med användaren – t.ex. webbläsare, mejlklient |
| 6 | Presentation | Översätter dataformat – t.ex. kryptering, komprimering |
| 5 | Session | Håller reda på dialogen mellan två enheter |
| 4 | Transport | Delar upp data i segment, ansvarar för felkontroll – t.ex. TCP/UDP |
| 3 | Nätverk | Skickar paket till rätt IP-adress – t.ex. IP, ICMP |
| 2 | Datalänk | Bestämmer hur data skickas över ett lokalt nätverk – t.ex. MAC-adresser |
| 1 | Fysiskt | Själva hårdvaran – kablar, elektriska signaler, kontakter |

Ett vanligt sätt att minnas lagren är med hjälp av minnesregler, exempelvis:

* **"All People Seem To Need Data Processing"** (uppifrån och ner)
* **"Please Do Not Throw Sausage Pizza Away"** (nerifrån och upp)

#### **Exempel: Hur en webbsida skickas via OSI-modellen**

När du skriver in en webbadress och trycker Enter i webbläsaren:

* Lager 7: Webbläsaren (applikationen) genererar en begäran om en webbsida.
* Lager 6: Data kan krypteras med TLS/SSL.
* Lager 5: En session skapas mellan dig och webbservern.
* Lager 4: Begäran delas upp i segment, och TCP ser till att allt kommer fram i rätt ordning.
* Lager 3: IP-protokollet lägger på avsändar- och mottagaradress.
* Lager 2: Ethernet lägger till MAC-adresser.
* Lager 1: Data skickas som elektriska signaler i nätverkskabeln.

När informationen når mottagaren går den uppåt igen genom lagren, tills webbsidan visas.

## 8.4 Diskussions- och kontrollfrågor

1. Vad står OSI för och vilken organisation utvecklade modellen?
2. Varför behövdes en standardmodell för nätverkskommunikation på 1970-talet?
3. Nämn tre fördelar med att använda OSI-modellen.
4. Hur många lager innehåller OSI-modellen och vad heter det översta lagret?
5. Vad är syftet med transportlagret?
6. Vad innebär det att OSI-modellen är en "referensmodell"?
7. Hur används OSI-modellen i felsökning av nätverk?
8. Vilket lager ansvarar för att hantera IP-adresser?
9. Ge ett exempel på ett protokoll som verkar på applikationslagret.
10. Vad är skillnaden mellan lager 1 och lager 2?

## 8.5 Fördjupningslänkar

* Cisco – OSI Model Explained  
   <https://www.cisco.com/c/en/us/about/press/internet-protocol-journal/blogs/2021/03/osi-model.html>
* Cloudflare – What is the OSI Model?  
   <https://www.cloudflare.com/learning/ddos/glossary/open-systems-interconnection-model-osi/>
* GeeksForGeeks – OSI Model  
   <https://www.geeksforgeeks.org/layers-of-osi-model/>
* IBM – OSI Model Overview  
   <https://www.ibm.com/docs/en/cics-ts/5.3?topic=communications-open-systems-interconnection-osi-model>
* CompTIA – OSI Reference Model  
   <https://www.comptia.org/content/guides/understanding-the-osi-model>

## 8.6 Egna anteckningar

# **9 Introduktion till IT-säkerhet**

## **9.1 Vad är IT-säkerhet?**

IT-säkerhet handlar om att skydda digital information, system och tjänster mot obehörig åtkomst, skada eller avbrott. Det inkluderar både tekniska lösningar, som brandväggar och kryptering, och organisatoriska åtgärder, som rutiner och policies.

En fungerande IT-säkerhet ser till att rätt personer har tillgång till rätt information vid rätt tidpunkt – och att obehöriga inte har det. Samtidigt handlar det om att kunna lita på att informationen inte manipulerats och att den finns tillgänglig när den behövs.

För företag och organisationer är IT-säkerhet avgörande. Inte bara för att skydda interna resurser, utan också för att uppfylla lagkrav som GDPR och för att upprätthålla kundernas förtroende. För privatpersoner handlar det ofta om att skydda personliga uppgifter, bilder, bankinformation och annan känslig data.

### 9.1.1 Hotbild och risker

IT-säkerhet hotas ständigt av olika typer av attacker, både avsiktliga och oavsiktliga. De vanligaste hoten kan delas in i följande kategorier:

* **Malware (skadlig kod)**: Virus, trojaner, maskar, spyware och ransomware används för att stjäla information, låsa datorer eller sprida sig till andra system.
* **Social engineering**: Manipulationstekniker där angriparen försöker lura användare att avslöja information eller klicka på farliga länkar, t.ex. via phishing-mail.
* **Dataintrång**: Angripare som bryter sig in i nätverk eller system för att komma åt känslig data.
* **Slarv eller misstag**: Mänskliga fel, t.ex. svaga lösenord eller felaktig konfiguration, står bakom många säkerhetsproblem.
* **Insiderhot**: Anställda eller tidigare anställda som utnyttjar sin behörighet för att skada verksamheten.
* **Avbrott och störningar**: DDoS-attacker, hårdvarufel eller elavbrott som gör system otillgängliga.

I takt med att allt fler tjänster flyttas till molnet och fler enheter kopplas upp (IoT), ökar attackytan – och behovet av god säkerhet ökar i samma takt.

### 9.1.2 CIA-triaden

CIA-triaden är en grundmodell inom IT-säkerhet och beskriver tre viktiga principer som all säkerhetsarbete bör utgå ifrån:

#### C – Confidentiality (Sekretess)

Endast behöriga personer ska kunna ta del av viss information. Exempel: lösenordsskyddade filer, krypterade meddelanden och åtkomstkontroll på nätverksresurser.

#### I – Integrity (Integritet)

Information ska vara korrekt och inte ha förändrats utan tillstånd. Exempel: kontrollsummor, loggning av ändringar och signering av dokument.

#### A – Availability (Tillgänglighet)

System och data ska vara tillgängliga när de behövs. Exempel: redundanta servrar, UPS (avbrottsfri strömförsörjning) och skydd mot DDoS-attacker.

Ett välbalanserat säkerhetsarbete tar hänsyn till alla tre. Om man skyddar informationen så hårt att den inte går att använda, är tillgängligheten hotad. Om man däremot släpper in alla, förloras både sekretess och integritet.

### 9.1.3 Kontrollfrågor

1. Vad innebär begreppet IT-säkerhet?
2. Nämn tre exempel på vanliga hot mot IT-system.
3. Vad är skillnaden mellan ett virus och en trojan?
4. Hur fungerar social engineering?
5. Varför är användarnas misstag ett säkerhetsproblem?
6. Vad innebär "insiderhot" inom IT-säkerhet?
7. Beskriv vad "C" i CIA-triaden står för och ge ett exempel.
8. Vad innebär "I" i CIA-triaden?
9. Hur skyddar man tillgängligheten i ett IT-system?
10. Hur kan man uppnå balans mellan sekretess, integritet och tillgänglighet?

### 9.1.4 Fördjupningslänkar

1. Microsoft – What is cybersecurity?  
    <https://www.microsoft.com/en-us/security/business/security-101/what-is-cybersecurity>
2. CISA – Cybersecurity Awareness  
    <https://www.cisa.gov/topics/cybersecurity>
3. OWASP Top 10 (hot och sårbarheter)  
    <https://owasp.org/www-project-top-ten/>
4. MSB – Så skyddar du dig mot nätfiske  
    <https://www.msb.se/sv/rad-till-privatpersoner/informationssakerhet/phishing/>
5. CERT-SE – Aktuella hot och sårbarheter  
    <https://www.cert.se/>

### 9.1.5 Egna anteckningar

*(Lämna plats för elevernas egna reflektioner och noteringar här)*

## 9.2 Skydd på olika nivåer

Fysisk säkerhet är en grundläggande men ofta förbisedd del av IT-säkerheten. Den handlar om att skydda själva utrustningen – servrar, datorer, nätverksutrustning och annan hårdvara – från fysiska hot som stöld, sabotage, brand eller oavsiktlig skada.

Ett serverrum bör till exempel ha lås, brandlarm, övervakning och skydd mot överspänning. Endast behörig personal ska ha tillgång, gärna genom passerkort eller biometriska system. Kablar bör inte vara lättillgängliga, särskilt inte om de leder till nätverksinfrastruktur eller kritisk utrustning.

Även bärbara datorer, USB-minnen och mobila enheter bör hanteras med försiktighet. En förlorad laptop utan kryptering kan leda till dataintrång, även om den aldrig kopplas upp mot ett nätverk.

### 9.2.1 Användarsäkerhet

#### Lösenord – varför längd och variation spelar roll

Ett lösenords styrka beror på:

1. **Hur långt det är**
2. **Vilka tecken det innehåller**
3. **Om det är lätt att gissa (t.ex. "password123")**

Ett vanligt sätt att förstå styrkan hos lösenord är att räkna hur många möjliga kombinationer som finns. Det görs med formeln:

**Möjliga kombinationer = antal möjliga teckenⁿ**, där *n* = lösenordets längd

#### **Exempel:**

|  |  |  |
| --- | --- | --- |
| **Teckenuppsättning** | **Antal möjliga tecken** | **Komb. vid 6 tecken** |
| Endast siffror (0–9) | 10 | 10⁶ = **1 000 000** |
| Små bokstäver (a–z) | 26 | 26⁶ ≈ **308 miljoner** |
| Små + stora bokstäver (a–z, A–Z) | 52 | 52⁶ ≈ **19 miljarder** |
| Bokstäver + siffror + specialtecken | ca 96 | 96⁶ ≈ **782 miljarder** |

Och med 8 tecken?

* 96⁸ ≈ **7 213 895 789 838 336** kombinationer – **över 7 biljarder**

Detta är varför ett lösenord på bara 6 tecken kan knäckas på sekunder vid en bruteforce-attack – medan ett starkt lösenord på 12 tecken kan ta flera tusen år att knäcka, även med kraftfull hårdvara.

**Tips till eleverna**: Ett säkert lösenord bör vara minst **12 tecken långt**, gärna en "lösenordsfras" (t.ex. KorvMedBröd!2025) som är både stark och lätt att komma ihåg.

#### Flerfaktorsautentisering (MFA) – vad det är och hur det funkar

Att ha ett starkt lösenord är bra – men det räcker inte alltid. Därför använder man ofta **flerfaktorsautentisering (MFA)**, som kräver minst två olika typer av bevis på att du är du.

#### **De tre autentiseringsfaktorerna:**

|  |  |  |
| --- | --- | --- |
| **Faktor** | **Beskrivning** | **Exempel** |
| **Vad du vet** | Något du kan utantill | Lösenord, PIN-kod |
| **Vad du har** | Något du fysiskt äger | Mobiltelefon, ID-kort, Yubikey |
| **Vem du är** | Något som är unikt för dig | Fingeravtryck, ansikte, iris |

För att räknas som **MFA** måste du kombinera **minst en från två olika kategorier**  
 (Två lösenord är **inte** MFA – det är bara två saker du *vet*)

#### **Exempel på MFA-lösningar:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Exempel** | **Faktor 1** | **Faktor 2** | **MFA?** |
| Lösenord + SMS-kod | Vad du vet | Vad du har | ✅ |
| Lösenord + fingeravtryck | Vad du vet | Vem du är | ✅ |
| PIN-kod + lösenord | Vad du vet | Vad du vet | ❌ |
| Lösenord + autentiseringsapp | Vad du vet | Vad du har | ✅ |
| Fingeravtryck + ansiktsigenkänning | Vem du är | Vem du är | ❌ |

MFA skyddar användarkonton även om lösenordet skulle bli stulet – eftersom angriparen saknar din telefon eller ditt fingeravtryck.

### 9.2.2 Programvarusäkerhet – Antivirus och uppdateringar

Ett vanligt misstag bland användare är att tro att ett installerat antivirusprogram automatiskt gör datorn "osårbar". Så är det inte. Antivirus är en **viktig del** av säkerheten – men bara om det hålls uppdaterat och används tillsammans med andra säkerhetsåtgärder.

Antivirusprogram fungerar ungefär som ett vaccin: de letar efter och blockerar kända hot, baserat på en databas med virusdefinitioner. Därför är det avgörande att dessa **viruslistor uppdateras regelbundet**, annars blir skyddet snabbt föråldrat. Ett antivirusprogram som inte uppdateras ger bara en **falsk trygghet**.

Det är också viktigt att **själva programmet** uppdateras, eftersom även antivirus kan innehålla sårbarheter som måste rättas till.

#### **🛡️ Exempel på vanliga antivirusprogram (kommersiella och gratis):**

1. Bitdefender – Känt för hög upptäcktsgrad och låg systempåverkan. Finns i både gratis- och betalversion.  
2. Avast – Populärt gratisprogram med tilläggsfunktioner som brandvägg och lösenordshanterare.  
3. Kaspersky – Ryskt ursprung men tekniskt välrenommerat. Används både av privatpersoner och företag.  
4. ESET NOD32 – Lättviktigt program med god realtidsövervakning och låg påverkan på prestanda.  
5. Windows Defender – Inbyggt i Windows 10/11 och ger ett grundskydd direkt ur lådan. Har förbättrats kraftigt de senaste åren och är numera ett fullvärdigt alternativ.

Att ha ett antivirusprogram betyder dock **inte** att man kan klicka på vad som helst eller surfa in på okända sidor utan risk. Antivirus är bara en del av ett större skydd. Det viktigaste skyddet är användaren själv – och det sunda förnuftet.

Alla antivirusprogram är inte bara skydd. I vissa fall har de visat sig själva vara en risk.

Ett uppmärksammat exempel är **Avast**, som under flera år samlade in stora mängder användardata via sitt antivirusprogram. Informationen såldes sedan vidare till tredjepartsföretag genom ett dotterbolag – något som ledde till stämningar, kraftig kritik och till slut att bolaget stängde ner verksamheten för datainsamlingen.

Även andra antivirusprogram samlar in **telemetridata**, alltså information om hur programmet används, vilka webbplatser som besöks, vilka filer som analyseras etc. Detta sker ofta "i bakgrunden" och finns ibland nämnt i användarvillkoren eller integritetspolicyn.

💡 **Tips:** Innan du installerar ett antivirusprogram – läs på! Kontrollera hur programmet hanterar din data, om det går att stänga av datainsamling, och hur företaget tjänar pengar på sin produkt (särskilt om den är gratis).

### 9.2.3 Nätverkssäkerhet

Nätverkssäkerhet handlar om att skydda all data som färdas mellan enheter – oavsett om det är inom ett lokalt nätverk eller över internet. Eftersom nätverket är den digitala motsvarigheten till en motorväg är det också ett favoritmål för angripare. Därför är det avgörande att övervaka, begränsa och skydda trafiken på olika nivåer.

#### Brandväggar – nätverkets väktare

En brandvägg är en säkerhetsmekanism som kontrollerar vilken trafik som får komma in eller ut ur ett nätverk eller en dator. Det finns två huvudtyper:

* **Nätverksbrandväggar**: Skyddar hela nätverk, ofta placerade i routrar eller dedikerade säkerhetsenheter.
* **Host-baserade brandväggar**: Körs direkt på en enhet, t.ex. Windows Firewall.

Regler kan skapas för att:

* Blockera eller tillåta specifika portar (t.ex. blockera FTP om det inte används)
* Begränsa trafik till vissa IP-adresser eller nätverk
* Förhindra oönskad inkommande trafik från internet

Ett vanligt misstag är att öppna för mycket. "Allow all" är bekvämt – men livsfarligt.

#### VPN – Krypterad tunnel för säker anslutning

Ett **Virtual Private Network (VPN)** krypterar trafiken mellan en användare och ett nätverk. Det är särskilt användbart för:

* Fjärrarbete (hemifrån, resa)
* Att skydda data i osäkra nätverk (t.ex. offentligt Wi-Fi)
* Att dölja IP-adressen eller plats (används både av privatpersoner och företag)

Med VPN skapas en "tunnel" där trafik inte kan läsas av obehöriga. Det är som att färdas i en privat, ogenomskinlig rörpost i det öppna internet.

#### IDS/IPS – Larmsystem i nätverket

Intrusion Detection Systems (IDS) och Intrusion Prevention Systems (IPS) är säkerhetslösningar som:

* **IDS** övervakar trafiken och larmar om misstänkta mönster upptäcks
* **IPS** försöker stoppa angreppet i realtid

De är särskilt effektiva mot:

* Portscanningar
* Brute-force-attacker
* Försök att utnyttja kända sårbarheter (exploit kits)

Moderna lösningar kombinerar IDS och IPS, ibland kallade **NIDS/NIPS** (Network-based Intrusion Detection/Prevention System).

#### Segmentering – att inte lägga alla ägg i samma korg

Ett vanligt nybörjarmisstag är att lägga alla enheter i samma nätverk. Genom att segmentera nätverket – t.ex. med **VLAN (Virtual LAN)** – kan man begränsa skadorna vid en attack.

Exempel:

* Gästenheter får endast internetåtkomst, inte till servrar eller skrivare
* IoT-enheter isoleras i ett eget nät
* Administration hanteras i ett separat segment

Segmentering bygger på principen om **minsta möjliga åtkomst** och gör det mycket svårare för en angripare att röra sig fritt inom nätverket.

### 9.2.4 Kontrollfrågor

1. Vad är skillnaden mellan en host-baserad brandvägg och en nätverksbrandvägg?
2. Varför är det viktigt att ha brandväggsregler som endast tillåter nödvändig trafik?
3. Vad gör ett VPN?
4. När kan ett VPN vara extra användbart?
5. Vad är ett IDS och vad är dess syfte?
6. Hur skiljer sig ett IPS från ett IDS?
7. Ge ett exempel på en typ av attack som IDS/IPS kan upptäcka.
8. Vad menas med nätverkssegmentering?
9. Ge ett exempel på hur VLAN kan öka säkerheten i ett nätverk.
10. Vad innebär principen om "minsta möjliga åtkomst"?

### 9.2.5 Fördjupningslänkar

1. Cisco – Vad är en brandvägg?  
    <https://www.cisco.com/c/sv_se/products/security/firewalls/what-is-a-firewall.html>
2. Cloudflare – Vad är VPN och hur fungerar det?  
    <https://www.cloudflare.com/learning/privacy/what-is-a-vpn/>
3. Fortinet – Vad är IDS och IPS?  
    <https://www.fortinet.com/resources/cyberglossary/intrusion-detection-system>
4. Palo Alto – Network Segmentation  
    <https://www.paloaltonetworks.com/cyberpedia/what-is-network-segmentation>
5. MSB – Segmentera ditt nätverk  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/tekniska-atgarder/segmentera-ditt-natverk/>

### 9.2.6 Egna anteckningar

*(Lämna plats för egna reflektioner och fördjupningar här.)*

## 9.3 Trådlös säkerhet

Trådlösa nätverk – som Wi-Fi – har blivit standard i både hem och företag. De är enkla att sätta upp, men också lätta att attackera om man inte tänker på säkerheten. Eftersom signalerna sprids genom luften, räcker det att befinna sig **inom räckvidd** för att kunna försöka avlyssna eller manipulera trafiken. Därför krävs särskilda skyddsåtgärder.

### 9.3.1 Historik – från WEP till WPA3

#### WEP – Wired Equivalent Privacy (1997)

När Wi-Fi började användas på 90-talet var **WEP** det första säkerhetsprotokollet. Namnet antydde att det skulle ge samma säkerhet som ett trådbundet nätverk. I verkligheten var det **katastrofalt svagt**.

Problem:

* Använde **statiska krypteringsnycklar** (ofta 64 eller 128 bitar)
* Byggde på **RC4**, en sårbar algoritm
* Kunde knäckas på några minuter med enkla verktyg

#### WPA och WPA2 (2003–2004)

När sårbarheterna i WEP blev kända utvecklades **WPA** (Wi-Fi Protected Access) som en tillfällig lösning. Sen kom **WPA2**, som använde **AES-kryptering** – mycket starkare än RC4.

* WPA använde **TKIP**, vilket var bättre än WEP men fortfarande inte perfekt.
* WPA2 införde **CCMP (AES-baserad kryptering)** och blev snabbt branschstandard.

#### WPA3 (2018)

Efter flera attacker mot WPA2 (bl.a. **KRACK**-attacken) kom **WPA3**, som:

* Använder **SAE** istället för PSK (förhindrar bruteforce offline)
* Har förbättrad kryptering för öppna nätverk
* Ger **framåtsekretess** (t.ex. skyddar gamla sessioner om lösenordet läcker)

❗ OBS: Alla routrar stöder inte WPA3 ännu. Ibland måste det aktiveras manuellt i inställningarna.

### 9.3.2 Skyddsåtgärder för trådlösa nätverk

Att bara "ha ett lösenord på Wi-Fi" räcker inte. Här är några centrala skyddsåtgärder:

|  |  |
| --- | --- |
| **Skyddsåtgärd** | **Förklaring** |
| **Använd WPA2 eller WPA3** | Undvik WEP eller öppen (okrypterad) Wi-Fi |
| **Stark Wi-Fi-lösenfras** | Minst 12 tecken, inte "admin123" eller "qwerty" |
| **Dölj eller byt SSID** | Byt från standardnamn som "TP-Link\_123" – det signalerar okunskap |
| **MAC-filter (begränsad nytta)** | Tillåt endast kända enheter – men MAC-adresser kan förfalskas (spoofing) |
| **Segmentera gästnätverk** | Ge gäster/IoT egna nätverk – utan tillgång till interna resurser |
| **Begränsa räckvidd** | Placera router centralt, minska sändningseffekt vid behov |

### 9.3.3 Vanliga attacker mot Wi-Fi

#### Evil Twin / Rogue Access Point

Angriparen sätter upp ett falskt nätverk med liknande namn som ditt riktiga – t.ex. "Skola\_Guest" → "Sk0la\_Guest". Om en användare ansluter kan all trafik avlyssnas eller manipuleras.

#### Deauthentication attacks

Med verktyg som *aireplay-ng* kan en angripare tvinga bort användare från nätverket, i hopp om att de ska återansluta – till ett falskt nätverk (Evil Twin).

#### Packet sniffing

Med t.ex. *Wireshark* kan trafik över ett öppet eller dåligt säkrat nätverk analyseras – och ibland även avlyssnas. Krypterad trafik skyddas, men metadata (vilka sidor som besöks, DNS-frågor etc.) kan fortfarande avslöjas.

#### Brute-force/WPS-attacker

Om WPS (Wi-Fi Protected Setup) är aktiverat, kan en angripare försöka gissa PIN-koden som används för att koppla upp sig – en funktion som bör **stängas av**.

🛡 **Tips**: Ha som regel att alltid stänga av WPS. Det är sällan värt bekvämligheten.

### 9.3.4 Kontrollfrågor

1. Vad stod WEP för och varför var det osäkert?
2. Vad är skillnaden mellan WPA och WPA2?
3. Vilken typ av kryptering använder WPA2?
4. Vad gör WPA3 mer säkert än tidigare versioner?
5. Vad är en "Evil Twin"-attack?
6. Hur kan man upptäcka eller skydda sig mot en rogue access point?
7. Vad innebär en deauthentication-attack?
8. Vad kan verktyg som Wireshark användas till?
9. Varför bör man undvika att använda WPS?
10. Ge tre konkreta tips på hur man säkrar ett trådlöst nätverk.

### 9.3.5 Fördjupningslänkar

1. Wikipedia – WEP  
    <https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy>
2. Comparitech – WPA vs WPA2 vs WPA3  
    <https://www.comparitech.com/blog/vpn-privacy/wpa-vs-wpa2-vs-wpa3/>
3. Wireshark – Official Site  
    <https://www.wireshark.org/>
4. Aircrack-ng Suite – Verktyg för trådlösa tester  
    <https://www.aircrack-ng.org/>
5. MSB – Säker användning av trådlösa nätverk  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/tekniska-atgarder/tradlost-natverk/>

### 9.3.6 Egna anteckningar

*(Lämna plats för egna reflektioner och fördjupningar här.)*

## **9.4 Säkerhet i Windowsmiljö**

### **9.4.1 Windows Security Center**

**Windows Security Center** (eller *Säkerhetscenter*) är den centrala platsen där användare och administratörer får en översikt över datorns säkerhetsstatus. Det introducerades redan i Windows XP Service Pack 2 och har utvecklats rejält i senare versioner, särskilt i Windows 10 och 11 där det numera heter **Windows Security** (*Windows-säkerhet*).

Här samlas flera viktiga funktioner i ett enda gränssnitt:

* **Virus- och hotskydd** (Windows Defender)
* **Kontoskydd** (UAC, inloggning, MFA)
* **Brandvägg och nätverksskydd**
* **App- och webbläsarkontroll** (SmartScreen, isolering)
* **Enhetssäkerhet** (t.ex. SecureBoot, TPM)
* **Enhetsprestanda och hälsa**
* **Familjealternativ** (föräldrakontroll)

**Syfte**: Att ge användaren ett tydligt, samlat läge där man snabbt ser om något är fel – t.ex. om antivirus är inaktiverat, om brandväggen är avstängd eller om det saknas viktiga uppdateringar.

👉 **Tips för användare**: Du hittar det via *Inställningar → Uppdatering och säkerhet → Windows-säkerhet* eller genom att söka på "Windows Security" i Start-menyn.

**Fun fact**: Om du installerar ett annat antivirusprogram (t.ex. Norton eller Avast), ersätts ofta vissa funktioner i Windows Security – men det fortsätter ändå att visa status och meddela om något saknas.

### **9.4.2 UAC – User Account Control**

**User Account Control (UAC)** är en säkerhetsfunktion som kontrollerar **vem som får göra vad** i systemet – särskilt när det gäller att ändra inställningar som påverkar hela datorn.

**Vad gör UAC?**  
 När du försöker göra något som kräver administratörsrättigheter (t.ex. installera program, ändra systemfiler, starta kommandotolk som admin), dyker det upp en **UAC-ruta** som frågar:  
 *”Vill du tillåta att den här appen gör ändringar på din enhet?”*

Om du är inloggad som standardanvändare måste du ange ett adminlösenord. Om du redan är admin måste du bekräfta ditt val.

**Varför finns det?**  
Innan UAC (t.ex. i Windows XP) kördes de flesta användare som administratörer hela tiden – vilket gjorde det enkelt för virus och skadlig kod att göra ändringar i systemet.  
Med UAC körs program som **vanliga användare** tills du aktivt godkänner att de får göra större förändringar. Det är ett slags digital spärr – som en säkerhetsvakt vid dörren.

**UAC-nivåer (Windows 10/11):**

* 🟥 *Alltid meddela* – Frågar även om du gör ändringar via kommandorad
* 🟧 *Meddela endast vid appar från tredje part* – Standardnivå
* 🟨 *Meddela inte* – Riskabelt, bör undvikas

👀 **Typiska scenarier där UAC dyker upp:**

|  |  |
| --- | --- |
| **Åtgärd** | **Kräver UAC?** |
| Installera ett nytt program | ✅ Ja |
| Byta inställningar i Kontrollpanelen | ✅ Ja |
| Skapa eller ta bort användarkonto | ✅ Ja |
| Öppna Word eller Chrome | ❌ Nej |

**UAC skyddar även mot skadlig kod** som försöker köra i bakgrunden utan att du vet om det – eftersom programmet måste få godkännande innan det får högre rättigheter.

### **9.4.3 Windows Defender (Microsoft Defender)**

**Windows Defender**, numera kallat **Microsoft Defender Antivirus**, är Microsofts inbyggda antiviruslösning i Windows 10 och 11. Det är **gratis**, **automatiskt aktiverat** och ger ett grundskydd direkt efter installation av operativsystemet.

**Funktioner i Microsoft Defender:**

* **Realtidsskydd** – söker efter och blockerar virus, trojaner, spyware m.m.
* **Molnskydd** – använder Microsofts molntjänster för att snabbt identifiera nya hot
* **Ransomware-skydd** – övervakar känsliga mappar mot kryptering
* **SmartScreen-filter** – varnar för farliga webbsidor och nedladdningar
* **Offline-genomsökning** – kan köras utanför Windows för att ta bort svårupptäckta hot

**Utveckling**: När det först lanserades hade det relativt dåligt rykte – men **idag presterar Microsoft Defender på samma nivå som många kommersiella alternativ** (enligt oberoende tester som AV-Test.org).

**Konfiguration och översikt**:  
Du hittar Defender via *Inställningar → Uppdatering och säkerhet → Windows-säkerhet → Virus- och hotskydd*.

**Bra att veta**:  
Om du installerar ett annat antivirusprogram, t.ex. Kaspersky eller Bitdefender, **stängs Defender automatiskt av** för att undvika konflikt. Du kan då behöva manuellt aktivera det igen om du tar bort det andra programmet.

**Defender är mer än antivirus**: I företagsmiljö används **Microsoft Defender for Endpoint**, som innehåller avancerad hotdetektion, analys, och integration med Azure och andra Microsoft 365-tjänster.

**Tips**: Defender räcker gott för de flesta elever och privatpersoner – så länge man har **försiktigt surfande, starka lösenord och regelbundna uppdateringar**.

### **9.4.4 Gruppolicy och säkerhetsmallar (Group Policy & Security Templates)**

**Group Policy** (gruppolicy) är ett kraftfullt verktyg i Windows – särskilt i företagsmiljöer – för att centralt styra hur datorer, konton och säkerhetsinställningar fungerar.

Tänk dig en skola med 100 datorer. Istället för att sitta och manuellt konfigurera varje dator kan man använda en **gruppolicy** för att bestämma allt från skrivbordsbakgrund till vilka program som får köras – och vad eleverna inte får göra.

**Exempel på vad du kan styra med Group Policy:**

* Tvinga starka lösenord och lösenordsbyte var 90:e dag
* Blockera åtkomst till Kontrollpanelen
* Förhindra att USB-minnen används
* Tvinga uppdateringar av antivirus eller system
* Automatisk inloggning av program
* Begränsa internetanvändning

**Säkerhetsmallar (Security Templates)**  
 Det finns färdiga mallar som innehåller rekommenderade säkerhetsinställningar. Dessa kan importeras via Group Policy för att snabbt säkra t.ex. arbetsstationer, servrar eller elevernas datorer.

**Gruppolicy Editor (gpedit.msc)**  
 Verktyget används lokalt på datorer för att visa och redigera policys. Dock endast tillgängligt i **Pro- eller Enterprise-versionerna av Windows** (inte Home).

**Domänmiljö och Active Directory (AD)**  
 I större nätverk hanteras Group Policies via **Group Policy Management Console (GPMC)** och tillämpas automatiskt på alla datorer i domänen. Då kan man ha olika policies beroende på användargrupp, avdelning, datorrum m.m.

### **9.4.5 NTFS-behörigheter – vem får göra vad?**

**NTFS (New Technology File System)** är det filsystem som används i moderna Windows-versioner, och det innehåller stöd för avancerad **behörighetskontroll**. Det gör att du kan styra exakt **vem som får läsa, skriva, ändra eller ta bort filer och mappar**.

**Varför är detta viktigt?**  
Tänk dig en skoldator där vem som helst kan radera andras filer – eller en server där elever kan läsa betygsfiler. Rätt behörigheter är en grundpelare i IT-säkerhet.

**De vanligaste NTFS-behörigheterna:**

|  |  |
| --- | --- |
| **Behörighet** | **Vad innebär det?** |
| **Läsa (Read)** | Kan visa innehåll, men inte ändra |
| **Skriva (Write)** | Kan skapa och ändra filer |
| **Ändra (Modify)** | Kan läsa, skriva, ändra och ta bort |
| **Fullständig behörighet** | Har total kontroll (inkl. ägarskap, rättigheter) |

Behörigheter kan sättas på **individer** eller **grupper** – t.ex. att alla i gruppen Lärare får läsa en mapp, men bara Admin får ändra.

**Ärvda behörigheter:**  
Mappar ärver ofta rättigheter från sin överordnade mapp. Det går att bryta ärvningen om man vill ha särskilda regler.

**Tips**:

* Använd **minsta nödvändiga behörighet** – användare bör inte ha mer rättigheter än de behöver.
* Använd **grupper** istället för att sätta rättigheter på varje användare. Det gör administrationen enklare.
* Undvik att ge "Alla" fullständig åtkomst till något – det är som att lämna dörren olåst.

### **9.4.6 BitLocker – kryptera hårddisken**

**BitLocker** är Windows inbyggda verktyg för att **kryptera hela hårddisken**. Det gör att **ingen kan läsa datan utan rätt nyckel**, inte ens om de fysiskt stjäl datorn.

**Hur fungerar det?**  
När BitLocker är aktiverat krypteras all data automatiskt i bakgrunden. Vid uppstart kontrolleras att systemet inte har manipulerats. För att låsa upp krävs en av följande:

* **TPM (Trusted Platform Module)** – en hårdvarumodul som lagrar krypteringsnyckeln säkert
* **PIN-kod**
* **USB-nyckel**
* **Microsoft-konto eller domäninloggning**

**Varför använda BitLocker?**

* Skyddar känsliga filer även om datorn blir stulen
* Krävs i många företag p.g.a. lagkrav och GDPR
* Kan användas tillsammans med Windows Hello och/eller TPM för smidig inloggning

**BitLocker To Go** – Kryptera även **USB-stickor** och **externa diskar**!

**Återställningsnyckel**:  
Vid aktivering skapas en återställningsnyckel. Den kan sparas i molnet, på ett USB-minne eller skrivas ut. **Tappa inte bort den!** Utan den är det omöjligt att komma åt datan om något går fel.

**Tips till elever**:  
 BitLocker är enkelt att slå på i Windows Pro. Gå till *Kontrollpanelen → BitLocker Drive Encryption*. På skolans datorer bör detta dock skötas centralt via IT.

### **9.4.7 Windows Firewall – skyddet mellan dig och internet**

**Windows Firewall** är en **inbyggd brandvägg** som styr vilken trafik som får passera in och ut från din dator. Den fungerar som en **väktare** som säger “ja” eller “nej” till olika typer av anslutningar.

**Två huvudtyper av brandväggsregler:**

|  |  |
| --- | --- |
| **Typ** | **Förklaring** |
| **Inkommande regler** | Bestämmer vad som får *komma in* till datorn från nätverket/internet |
| **Utgående regler** | Bestämmer vad som får *lämna* datorn mot nätverket/internet |

**Exempel:**

* Du vill tillåta att Spotify hämtar musik från internet (utgående)
* Du vill *inte* tillåta att någon fjärransluter till din dator (inkommande)

**Lägen i Windows Firewall:**

|  |  |
| --- | --- |
| **Nätverkstyp** | **Beskrivning** |
| **Privat** | Hemmanätverk – du litar på andra enheter |
| **Offentligt** | T.ex. kafé eller flygplats – mer restriktiv |
| **Domän** | Företags- eller skolnätverk, styrs via Group Policy |

Viktigt: När du kopplar upp dig på ett nytt nätverk får du frågan om du vill klassa det som privat eller offentligt – detta påverkar vilka regler brandväggen använder!

**Tips:**

* Stäng aldrig av brandväggen helt – det är som att lämna dörren olåst.
* För avancerade regler, gå till *“Windows Defender Firewall med avancerad säkerhet”* – där kan du t.ex. tillåta eller blockera specifika portar eller IP-adresser.
* Många program lägger automatiskt till undantag i brandväggen – dubbelkolla alltid att du litar på programmet!

### **9.4.8 Autentisering: Kerberos och NTLM**

När du loggar in på en Windows-dator i ett nätverk (särskilt i en domän), används ett autentiseringsprotokoll för att kontrollera att du verkligen är den du säger att du är. Windows använder främst **Kerberos** eller **NTLM**.

**Vad är NTLM?**

NTLM står för **NT LAN Manager** och är ett äldre autentiseringsprotokoll.

* Enkelt att använda
* Osäkert idag – bygger på hashbaserade lösenord
* Sårbart för "pass-the-hash"-attacker

**Vad är Kerberos?**

Kerberos är ett **modernt och säkrare protokoll**, baserat på *biljetter* som delas ut av en **Key Distribution Center (KDC)** – vanligtvis din domänkontrollant.

Fördelar:

* Krypterad kommunikation
* Inga lösenord skickas direkt
* Snabbare och säkrare än NTLM

**Hur det funkar i korthet:**

1. Du loggar in → KDC bekräftar att du är du
2. Du får en biljett (TGT – Ticket Granting Ticket)
3. När du vill nå en tjänst (t.ex. en filserver) använder du biljetten
4. Tjänsten litar på biljetten utan att be om lösenord igen

**NTLM används fortfarande som backup** om Kerberos inte fungerar (t.ex. om du inte är med i en domän).

**Tips:**

* I företagssammanhang bör **Kerberos alltid användas som standard**
* NTLM bör **avaktiveras** i moderna miljöer om det är möjligt

### **9.4.9 Skydd mot malware och phishing**

**Malware** (skadlig kod) och **phishing** (nätfiske) är två av de vanligaste hoten mot både privatpersoner och organisationer. De kan stjäla information, låsa datorer, eller till och med ta över hela nätverk.

**Vad är malware?**

Malware är ett samlingsnamn för olika typer av skadlig programvara:

|  |  |
| --- | --- |
| **Typ** | **Förklaring** |
| **Virus** | Kopierar sig själv och infekterar andra filer |
| **Mask (worm)** | Sprider sig automatiskt via nätverk |
| **Trojan** | Låtsas vara något användbart, men innehåller skadlig kod |
| **Spyware** | Samlar in information i smyg |
| **Ransomware** | Krypterar filer och kräver pengar för att låsa upp dem |

**Vad är phishing?**

Phishing handlar om att lura användare att ge ifrån sig känslig information – t.ex. lösenord eller kortuppgifter. Det sker ofta via:

* E-post ("Din faktura är sen – klicka här")
* Fejkade inloggningssidor (ser ut som t.ex. banken)
* Sms ("Verifiera din leverans via länken")
* Direktmeddelanden i sociala medier

**Skydd mot malware och phishing – vad du kan göra:**

**Använd ett uppdaterat antivirusprogram** (t.ex. Windows Defender)

**Aktivera realtidsskydd** – programmet varnar direkt om något misstänkt sker

**Håll operativsystem och program uppdaterade** – många attacker sker via gamla sårbarheter

**Använd webbläsartillägg** som varnar för misstänkta sajter (t.ex. Microsoft Defender SmartScreen)

**Tänk efter innan du klickar** – är det rimligt att du får ett mejl från Skatteverket kl. 02:48?

**Aktivera e-postfiltrering** – de flesta system (t.ex. Microsoft 365) har inbyggt skydd

**Tips:**

* Om ett mejl känns konstigt – dubbelkolla avsändaren
* Håll muspekaren över en länk innan du klickar, så ser du vart den leder
* Rapportera misstänkt phishing till din IT-avdelning eller Skatteverket om det gäller ekonomiskt bedrägeri

**Fakta:**  
 Microsofts säkerhetsrapporter visar att över 90 % av dagens cyberattacker börjar med phishing!

### **9.4.10 Group Policy och säkerhetsmallar**

I en Windows-domänmiljö (t.ex. i skolor eller företag) används **Group Policy (GPO)** för att styra inställningar på datorer och användarkonton – centralt, från en domänkontrollant.

Det kan handla om allt från att:

* Tvinga användare att ha säkra lösenord
* Förhindra åtkomst till vissa inställningar
* Automatisk installation av program
* Blockera USB-enheter
* Sätta skärmsläckare med lösenord

**Hur funkar det?**

GPO:er appliceras på:

* Användare
* Datorer
* Grupper
* OU (Organizational Units)

Dessa regler skickas ut varje gång en användare loggar in, eller en dator startas om.

**Säkerhetsmallar (Security Templates)**

Windows erbjuder färdiga mallar med rekommenderade säkerhetsinställningar. Dessa kan:

* Aktivera UAC
* Stänga av gamla protokoll (t.ex. SMBv1)
* Begränsa administrativa rättigheter
* Ställa in lösenordspolicy

Exempel: *"SECUREWS.INF"* är en mall för säkra arbetsstationer som kan importeras direkt i Group Policy.

**Tips:**

* Group Policy Editor = gpedit.msc (för lokala inställningar)
* För domänstyrda miljöer används *Group Policy Management Console (GPMC)*

**Varför är det viktigt?**

Utan centralt styrda policies blir det svårt att ha kontroll på säkerheten. Användare kan stänga av skydd, installera olämpliga program, eller ändra inställningar. Med GPO:er får du ordning och reda.

**Exempel från skolmiljö:**

* Elever kan inte öppna Inställningar
* Lärarkonton får administratörsrättigheter
* USB-portar är avstängda på elevdatorer

### 9.4.11Kontrollfrågor – Säkerhet i Windowsmiljö

1. Vad är syftet med Windows Security Center?
2. Hur fungerar User Account Control (UAC) och varför är det viktigt?
3. Vad är skillnaden mellan Microsoft Defender och tredjeparts antivirus?
4. Vad kan man göra med Group Policy i en domänmiljö?
5. Vad är NTFS-rättigheter och hur skiljer de sig från delningsrättigheter?
6. Vad skyddar BitLocker mot – och när är det verkningslöst?
7. Hur fungerar Windows-brandväggen på applikationsnivå?
8. Vad är skillnaden mellan Kerberos och NTLM?
9. Hur skyddar Windows mot phishingförsök och skadlig kod?
10. Varför är det viktigt att kombinera flera skydd (antivirus, brandvägg, UAC etc.)?

### 9.4.12 Fördjupningslänkar – Säkerhet i Windowsmiljö

1. Microsoft – Windows Security Center <https://support.microsoft.com/en-us/windows/security-center-in-windows>
2. Microsoft Learn – UAC Overview <https://learn.microsoft.com/en-us/windows/security/identity-protection/user-account-control>
3. Microsoft Defender Antivirus Documentation <https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/microsoft-defender-antivirus>
4. Microsoft – BitLocker Overview <https://learn.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview>
5. Kerberos vs. NTLM – Microsoft Docs <https://learn.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview>

### 9.4.13 Egna anteckningar

(Plats för elevens egna reflektioner, minnesregler eller fördjupningar.)

## 9.5 Säkerhet i Linuxmiljö

Linux används i allt från webbservrar och IoT-enheter till superdatorer och molntjänster. Därför är det avgörande att förstå hur man skyddar dessa system mot attacker. Många attacker mot Linux-system handlar inte om att bryta sig in via lösenord, utan om att utnyttja konfigurationsmissar, öppna portar, gamla versioner eller brist på loggning.

Viktigt! Denna del kommer i första hand utgå från Debian-baserade system såsom Ubuntu m.m. så vissa av kommandona kommer inte att stämma mot andra distributioner.

### 9.5.1 Grundläggande säkerhetsprinciper

När man arbetar med Linux, särskilt i serversammanhang, är det viktigt att inte bara tänka på vad systemet *kan* göra – utan också hur man minimerar riskerna. Det finns ett antal principer som utgör grunden för hur vi tänker kring säkerhet i Linuxmiljöer.

### **Minsta möjliga behörighet (Least Privilege)**

Detta är kanske den allra viktigaste principen inom IT-säkerhet – och i Linux är det inbyggt i systemets design. Alla kommandon behöver inte köras som root. Tvärtom: ju färre kommandon som körs med hög behörighet, desto mindre risk att en attack lyckas.

**Exempel**:  
 Om du installerar ett paket som vanlig användare, händer inget. Det kräver sudo. Detta är ett medvetet hinder – och ett skydd.

**Tips:**  
Skapa hellre specifika användare för olika uppgifter än att ge root-access till allt. En webbtjänst ska t.ex. inte kunna skriva till /etc eller läsa andras hemkataloger.

### **Defense in Depth – Flera lager av skydd**

En brandvägg är bra. Men det räcker inte. Vad händer om någon lyckas ta sig förbi den? Då vill du ha en logg som visar vad som hände. Kanske ett IDS-system som larmar. Och kryptering som förhindrar åtkomst till känslig information.

**Tänk på det som en lök – säkerheten byggs i lager.**

### **Minska attackytan**

Ju mer som körs, desto mer kan gå fel. Linux-servrar installeras ofta med så lite som möjligt – just för att minska risken.

Exempel på onödiga risker:

* Öppna portar för tjänster du inte använder (ex: FTP, Telnet)
* Användarkonton som inte används längre
* Programvara som inte uppdateras

**Tips:**  
Kör ss -tuln för att se vilka portar som är öppna. Avinstallera tjänster du inte behöver. Inaktivera oanvända användare och konton.

### **Standardinställningar är inte alltid säkra**

Linux är kraftfullt – men ibland lämnas konfigurationen i ett osäkert tillstånd efter installation. Exempel:

* SSH tillåter root-login
* Loggar roteras inte
* Systemd-tjänster kör med onödiga rättigheter

**Att härda systemet är en viktig del av driften – inte ett engångsjobb.**

### **Säkerhet är en process – inte en produkt**

Det går inte att installera ett "säkerhetsprogram" och sen luta sig tillbaka. Säkerheten måste underhållas:

* Patcha regelbundet
* Följa med i CVE-flöden
* Testa din miljö då och då

### 9.5.2 Kontrollfrågor

1. Vad innebär principen "minsta möjliga behörighet"?
2. Varför är det viktigt att undvika att köra kommandon som root i onödan?
3. Vad menas med "defense in depth"?
4. Ge tre exempel på olika säkerhetslager du kan använda på en Linux-server.
5. Vad är en "attackyta" i ett operativsystem?
6. Varför är det en dålig idé att ha onödiga tjänster igång?
7. Vad kan kommandot ss -tuln användas till?
8. Varför kan standardinställningar vara osäkra?
9. Vad är risken med att tillåta SSH-login som root?
10. Varför är säkerhet en process, inte en engångsåtgärd?

### 9.5.3 Fördjupningslänkar

* NIST – Least Privilege  
   <https://csrc.nist.gov/glossary/term/least_privilege>
* RedHat – Security Best Practices  
   <https://access.redhat.com/articles/763933>
* Ubuntu Hardening Wiki  
   <https://wiki.ubuntu.com/Security/Hardening>
* Linux Foundation – Defense in Depth  
   <https://training.linuxfoundation.org/blog/linux-security-basics-defense-in-depth/>
* CIS Benchmarks – General Linux Guidance  
   <https://www.cisecurity.org/benchmark/linux>

### 9.5.4 Egna anteckningar

(Lämna plats för egna reflektioner och fördjupningar här.)

### **9.6 OS-härdning (Operating System Hardening)**

Att ”härda” ett operativsystem betyder att minska dess sårbarheter genom att ta bort onödiga tjänster, stänga oanvända portar, aktivera säkerhetsfunktioner och kontrollera åtkomst. I Linuxmiljö handlar det om att ställa in systemet så säkert som möjligt, utan att påverka dess funktion.

**Grundregeln:** Ju färre funktioner som är aktiva – desto mindre finns att attackera.

#### **CIS Benchmarks – säkerhetsstandard att följa**

Center for Internet Security (CIS) publicerar detaljerade guider för hur man säkrar olika operativsystem, inklusive Ubuntu, Debian och andra Linux-distributioner.

CIS-benchmarks innehåller:

* Rekommenderade inställningar för användarkonton, loggning, nätverkstjänster
* Tips på filbehörigheter, bootloader-skydd, m.m.
* Kommandon för att kontrollera och verifiera säkerhetsnivån

Exempel på verktyg:

* lynis – säkerhetsscanner som kan jämföra ditt system mot t.ex. CIS
* cis-cat – ett Java-baserat verktyg från CIS som kontrollerar compliance

#### **AppArmor & SELinux – kontrollera vad processer får göra**

AppArmor (Ubuntu, Debian) och SELinux (Red Hat, Fedora, CentOS) är så kallade **Mandatory Access Control (MAC)**-system. De fungerar som extraskydd ovanpå vanliga filrättigheter (typ chmod).

**AppArmor**:

* Lättare att förstå och administrera
* Har profiler för vanliga program (ex. apache2, mysql, ping)
* Status kollas med sudo aa-status

**SELinux**:

* Mer avancerat men svårare att felsöka
* Används ofta i enterprisemiljöer
* Status: sestatus

Tips: Håll dig till AppArmor i Ubuntu-miljöer om du är nybörjare – det är tillräckligt kraftfullt men mer pedagogiskt.

#### **Secure Boot – skydda redan vid uppstart**

Secure Boot är ett UEFI-baserat säkerhetsläge som gör att datorn endast startar operativsystem och program som är signerade. Det hindrar t.ex. att angripare lägger in en rootkit i bootkedjan.

I Linux:

* Måste vara aktiverat i BIOS/UEFI
* Kräver ofta att kärnmoduler är signerade (t.ex. vid custom-kärnor eller externa drivrutiner)

#### **Kryptera diskar och partitioner**

Fysisk säkerhet är inte alltid nog – särskilt inte för bärbara enheter. Kryptering ser till att data inte kan läsas utan rätt nyckel, även om hårddisken plockas ur datorn.

Alternativ:

* **LUKS** – (Linux Unified Key Setup), standard i t.ex. Ubuntu
  + Aktiveras ofta under installation (”Kryptera hela disken”)
  + Verktyg: cryptsetup
* **eCryptfs** – tidigare populärt för att kryptera hemmakataloger
* **ZFS native encryption** – för avancerade användare

Tips: Använd alltid fullständig diskkryptering på bärbara datorer. För stationära system kan det räcka att kryptera specifika kataloger (t.ex. /home, /var).

#### **Partitionering som säkerhetsåtgärd**

Att dela upp disken i flera partitioner gör att ett fel (t.ex. loggar som växer) inte tar ner hela systemet. Det kan också minska skadan vid ett intrång.

Rekommenderade separata partitioner:

* /home – användardata
* /var – loggar och dynamiskt innehåll
* /tmp – tillfälliga filer
* /boot – ibland, om du behöver separera uppstartsdata

Mount-alternativ:

* noexec – inga körbara filer tillåtna (bra för /tmp)
* nosuid – förhindrar setuid-program
* nodev – inga enhetsfiler tillåtna

**Tipsruta:**

Vill du snabbt se vilka mount-alternativ som används?  
 Kör: mount | grep "^/" eller findmnt

### **9.6.1 Kontrollfrågor**

1. Vad menas med att "härda" ett operativsystem?
2. Vad är CIS Benchmarks och varför är de användbara?
3. Vad är skillnaden mellan AppArmor och SELinux?
4. Vad gör Secure Boot, och varför är det viktigt?
5. Vad är LUKS och när används det?
6. Vad är fördelen med att ha flera partitioner i Linux?
7. Vad betyder mount-alternativen noexec, nosuid och nodev?
8. Vad är skillnaden mellan fysisk säkerhet och systemhärdning?
9. Vilka Linux-kommandon kan användas för att kontrollera AppArmor-status?
10. Varför bör /tmp monteras med begränsningar?

### **9.6.2 Fördjupningslänkar**

* CIS Benchmarks – <https://www.cisecurity.org/cis-benchmarks>
* AppArmor Guide (Ubuntu) – <https://wiki.ubuntu.com/AppArmor>
* SELinux Project – <https://selinuxproject.org/page/Main_Page>
* Ubuntu LUKS Encryption – <https://help.ubuntu.com/community/Full_Disk_Encryption_Howto_2019>
* Red Hat – Linux Partitioning and Mount Options – <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ch-mounting-file-systems>

### **9.6.3 Egna anteckningar**

*(Här kan du som elev skriva ned viktiga reflektioner, kommandon du vill minnas, eller egna exempel på säkerhetsåtgärder du använder i dina system.)*

## **9.7** **Paketuppdatering och patchhantering**

Att hålla systemet uppdaterat är en av de viktigaste säkerhetsåtgärderna i Linux – och ändå är det något som ofta förbises. Ett opatchat system är som att lämna dörren på glänt – oavsett hur bra låset är.

De flesta sårbarheter som utnyttjas av angripare är **redan kända** – men systemen de angriper är inte uppdaterade. Det är alltså inte hacker-magi som krävs – bara en ignorans från administratören.

### **Varför är patchar så viktiga?**

Patchar åtgärdar bland annat:

* Kända säkerhetshål
* Buggar som kan utnyttjas för att eskalera behörigheter
* Kompatibilitetsproblem
* Nya hot (t.ex. zero-day exploits som snabbt måste åtgärdas)

I ett större nätverk kan en sårbar server bli en inkörsport för angripare. Genom att hålla systemet uppdaterat stänger man dessa portar innan de utnyttjas.

### **Verktyg i Debian-baserade system**

I Debian och Ubuntu använder man främst följande kommandon för uppdatering:

|  |  |
| --- | --- |
| **Kommando** | **Förklaring** |
| sudo apt update | Uppdaterar paketlistor från repositories |
| sudo apt upgrade | Installerar tillgängliga uppdateringar för installerade paket |
| sudo apt full-upgrade | Samma som ovan men hanterar även beroenden som kräver borttagning |
| apt list --upgradable | Visar vilka paket som har uppdateringar tillgängliga |

Man kan även använda apt-get, men apt är numera standard för interaktivt bruk.

### **Automatisk uppdatering med *unattended-upgrades***

På servrar där man inte loggar in så ofta kan det vara bra att automatisera vissa uppdateringar.

#### **Installation:**

sudo apt install unattended-upgrades

#### **Aktivera:**

sudo dpkg-reconfigure --priority=low unattended-upgrades

Detta aktiverar automatiska säkerhetsuppdateringar. Vill du justera vilka paket som uppdateras eller få loggar, kan du redigera:

/etc/apt/apt.conf.d/50unattended-upgrades

Det går också att sätta upp notifiering via mail eller loggfiler om något uppdateras automatiskt.

### **Tips för patchhantering**

* Kör apt update && apt upgrade **minst en gång i veckan** om du inte har automatisk hantering.
* Kontrollera /var/log/apt/history.log för att se vad som installerats.
* Följ med i säkerhetsflöden för din distribution – Ubuntu har ett eget säkerhetsflöde på <https://ubuntu.com/security/notices>
* Undvik att köra uppdateringar utan att läsa vad som uppdateras – speciellt på produktionsservrar.
* Tänk på att vissa uppdateringar kräver **omstart** (ex. kernel).

### **9.7.1 Kontrollfrågor**

1. Vad är syftet med en patch?
2. Vad är skillnaden mellan apt update och apt upgrade?
3. Vilket kommando visar vilka paket som har uppdateringar tillgängliga?
4. Vad innebär apt full-upgrade?
5. Varför är det farligt att inte uppdatera ett Linux-system?
6. Vad är unattended-upgrades och vad används det till?
7. Hur installerar man unattended-upgrades?
8. Var hittar du loggar över vad som uppdaterats på ett system?
9. Varför bör man vara försiktig med uppdateringar på en produktionsserver?
10. Hur ofta bör du uppdatera systemet om du inte har aktiverat automatisk uppdatering?

### **9.7.2 Fördjupningslänkar**

* Ubuntu Security Notices  
   <https://ubuntu.com/security/notices>
* Debian Security Information  
   <https://www.debian.org/security/>
* Ubuntu Wiki – Unattended Upgrades  
   <https://wiki.ubuntu.com/UnattendedUpgrades>
* APT Command Reference (Debian Wiki)  
   <https://wiki.debian.org/apt>
* Linux Handbook – How to Update Ubuntu  
   <https://linuxhandbook.com/update-upgrade-ubuntu/>

### **9.7.3 Egna anteckningar**

*(Lämna plats för egna reflektioner och anteckningar här.)*

## **9.8 Systemövervakning och loggning (journald, rsyslog, AIDE)**

Att övervaka och logga vad som händer på ett Linux-system är avgörande för att kunna upptäcka intrångsförsök, spåra fel eller följa upp vad som gjorts. Loggar är som systemets minne – de berättar vem som gjorde vad, när och hur. Utan loggar blir felsökning och säkerhetsanalys i stort sett omöjlig.

#### **Journald – systemd:s loggsystem**

De flesta moderna Linux-distributioner använder systemd som init-system. Tillsammans med journald hanterar det systemets kärnloggar, tjänstmeddelanden och andra systemrelaterade loggar.

Exempel på kommandon:

journalctl # Visa alla loggar  
journalctl -b # Visa loggar från senaste boot  
journalctl -u ssh # Visa loggar för SSH-tjänsten  
journalctl --since "1 hour ago" # Visa loggar från senaste timmen

Tips: Använd less för att bläddra, eller grep för att filtrera specifika meddelanden.

journald lagrar loggar binärt, vilket gör dem svårare att manipulera än vanliga textloggar – men du kan också exportera till text om du vill.

#### **rsyslog – klassisk textbaserad logghantering**

Utöver journald används ofta rsyslog för att skicka loggar till olika filer, vanligtvis under /var/log. Det här systemet är fortfarande vanligt och kan också användas för att:

* Centralisera loggar från flera servrar
* Skicka loggar till externa logghanteringsverktyg
* Definiera egna regler för vad som loggas och hur

Vanliga loggfiler:

* /var/log/auth.log – inloggningar, sudo-användning
* /var/log/syslog – allmän systemlogg
* /var/log/kern.log – kernel-meddelanden
* /var/log/apt/history.log – paketinstallationer
* /var/log/faillog – misslyckade inloggningar

Konfigurationen sker via /etc/rsyslog.conf och eventuella filer i /etc/rsyslog.d/.

#### **AIDE – Advanced Intrusion Detection Environment**

AIDE är ett kraftfullt verktyg för **filintegritetskontroll** – alltså att hålla koll på om något på filsystemet ändras, till exempel:

* Om ett systembinär modifieras
* Om en konfigurationsfil skrivs om
* Om någon lagt till en fil i /bin eller /etc

Det fungerar som ett "före-efter"-system. Först skapar man en databas över hur systemet *ska* se ut. Sen jämför man regelbundet systemets aktuella tillstånd med denna databas.

Exempel:

sudo aideinit # Initiera databas  
sudo aide.wrapper --check # Kontrollera om något förändrats

Det är viktigt att den ursprungliga databasen sparas på ett säkert ställe – annars kan en angripare även manipulera den.

**Tips:** Schemalägg AIDE via cron eller systemd timers, så att kontroller sker automatiskt varje dag.

#### **Varför är detta viktigt?**

* Loggar ger spårbarhet – vem gjorde vad, när?
* Loggar används vid incidentutredningar.
* Ett förändrat system (utan din vetskap) är ett starkt tecken på intrång.
* Automatiserad övervakning sparar tid och ökar säkerheten.

### **9.8.1 Kontrollfrågor**

1. Vad är journald och vilken typ av loggar hanterar det?
2. Vad gör kommandot journalctl -u ssh?
3. Nämn två skillnader mellan journald och rsyslog.
4. Var sparas vanligen klassiska loggfiler i Linux?
5. Vad används /var/log/auth.log till?
6. Vad är AIDE och vilket säkerhetssyfte fyller det?
7. Vad händer vid ett AIDE-check om en systemfil har ändrats?
8. Varför är det viktigt att spara AIDEs databas på ett säkert ställe?
9. Hur kan AIDE automatiseras för daglig kontroll?
10. Hur kan loggar användas vid säkerhetsincidenter?

### **9.8.2 Fördjupningslänkar**

* Red Hat – Introduction to journalctl  
   <https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system_administrators_guide/sect-basic_system_administration-working_with_system_log_files>
* Ubuntu – Log Files Explained  
   <https://help.ubuntu.com/community/LinuxLogFiles>
* rsyslog – Officiell dokumentation  
   <https://www.rsyslog.com/doc/>
* AIDE Project  
   <https://aide.github.io/>
* Loggning i Linux – Comparitech  
   <https://www.comparitech.com/net-admin/linux-logging/>

### **9.8.3 Egna anteckningar**

(Lämna plats för egna reflektioner och kompletterande information här.)

## **9.9** **Root, sudo och användarbehörighet**

I Linux är säkerheten starkt knuten till användarbehörigheter. Den mest kraftfulla användaren är root, som har obegränsad tillgång till hela systemet. Att förstå hur root, sudo, och användarbehörigheter fungerar är en grundpelare i att säkra ett Linuxsystem.

### **Root – superanvändaren**

Root-användaren är den administrativa användaren i Linux. Den har full tillgång till alla filer, processer och kommandon. Inloggning som root är sällan en bra idé, eftersom:

* Alla kommandon körs utan begränsningar – ett misstag kan bli katastrofalt.
* Root-inloggning loggas ofta inte separat, vilket försvårar spårning.
* Om en angripare får root-access är systemet helt komprometterat.

Därför är det bättre att använda sudo – ett verktyg som tillfälligt ger administrativ behörighet till vanliga användare.

### **sudo – tillfällig superkraft**

Med sudo kan en vanlig användare köra specifika kommandon med root-rättigheter. Exempel:

sudo apt update

Första gången (eller efter ett tag) krävs lösenord, vilket loggas i systemet. Detta gör det möjligt att:

* Säkerställa vem som körde vad
* Minimera risken för misstag
* Begränsa rättigheter till specifika kommandon

Konfigurationen av sudo sker i /etc/sudoers, och bör alltid redigeras med visudo för att undvika syntaxfel:

sudo visudo

Där kan man t.ex. sätta att en användare endast får köra vissa kommandon som root:

student ALL=(ALL) NOPASSWD: /usr/sbin/service apache2 restart

### **Användar- och grupphantering**

Varje användare i Linux har ett UID (User ID), och varje grupp har ett GID (Group ID). Fil- och katalogbehörigheter baseras på tre kategorier:

|  |  |  |
| --- | --- | --- |
| **Behörighet** | **Symbol** | **Beskrivning** |
| r | read | Läsa filer / lista kataloger |
| w | write | Skriva till filer / ändra kataloginnehåll |
| x | execute | Köra filer / gå in i kataloger |

Exempel från ls -l:

-rwxr-xr-- 1 robin admin 4200 jul 18 10:00 script.sh

Här betyder det:

* **rwx** = ägaren (robin) får läsa, skriva och köra
* **r-x** = gruppen (admin) får läsa och köra
* **r--** = andra får bara läsa

### **Byta ägare och rättigheter**

* chown – byt ägare:

sudo chown robin:admin fil.txt

* chmod – ändra rättigheter (t.ex. 755):

chmod 755 script.sh

* usermod – lägg till användare i grupp:

sudo usermod -aG sudo student

### **Sudoers-fällor att undvika**

* Låt inte alla användare få full sudo-access utan eftertanke.
* Använd NOPASSWD endast om det är absolut nödvändigt.
* Håll reda på vem som kan köra vad via loggar (/var/log/auth.log).

Tips: Ett vanligt säkerhetsråd är att *inte* logga in som root, utan alltid använda sudo. På många system (som Ubuntu) är root-användaren t.o.m. avaktiverad som standard.

### **9.9.1 Kontrollfrågor**

1. Vad är root-användarens roll i Linux?
2. Varför bör man undvika att arbeta direkt som root?
3. Vad gör kommandot sudo, och varför används det istället för root-inloggning?
4. Vad gör visudo, och varför bör man använda det istället för att direkt redigera /etc/sudoers?
5. Vad innebär rättigheterna r, w och x för filer och kataloger?
6. Vad gör kommandot chown, och vad betyder syntaxen robin:admin?
7. Vad är skillnaden mellan chmod 755 och chmod 700?
8. Hur lägger man till en användare i en grupp, t.ex. sudo-gruppen?
9. Vad är syftet med att begränsa vilka kommandon en användare får köra med sudo?
10. Var kan man hitta loggar över användning av sudo?

### **9.9.2 Fördjupningslänkar**

* Ubuntu – Users and Groups:  
   <https://ubuntu.com/server/docs/users-and-groups>
* DigitalOcean – How To Use Sudo on Ubuntu:  
   <https://www.digitalocean.com/community/tutorials/how-to-use-sudo-on-ubuntu>
* Linuxize – Linux File Permissions Explained:  
   <https://linuxize.com/post/linux-file-permissions/>
* Red Hat – Introduction to the sudo Command:  
   <https://www.redhat.com/sysadmin/sudo-command>
* Ubuntu Manpages – visudo:  
   <https://manpages.ubuntu.com/manpages/focal/en/man8/visudo.8.html>

### **9.9.3 Egna anteckningar**

*(Lämna plats för egna reflektioner, exempel och kommandon)*

## **9.10** **Loggning och övervakning i Linux**

Loggning är en av de viktigaste delarna i ett säkert Linux-system. Utan loggar vet du inte vad som har hänt – eller vad som händer just nu. Loggar kan visa allt från inloggningsförsök till programkrascher, ändrade filer, tjänstestarter, nätverksproblem och attacker. Övervakning handlar om att i realtid hålla koll på systemets hälsa, resursförbrukning och potentiella hot.

### **Viktiga loggsystem**

#### **journald (systemd journal)**

De flesta moderna Linux-system (inklusive Ubuntu) använder systemd som init-system, vilket inkluderar journald som loggtjänst.

* Visar loggar med: journalctl
* Exempel: journalctl -xe för senaste händelser med extra info
* Loggar i binärt format – inte direkt läsbara med cat

#### **rsyslog (traditionellt loggsystem)**

Klassiskt loggsystem som skriver till textfiler i /var/log/, t.ex.:

* /var/log/auth.log – inloggningar, sudo
* /var/log/syslog – systemmeddelanden
* /var/log/kern.log – kärnloggar
* /var/log/dmesg – hårdvarumeddelanden från boot

De flesta Ubuntu-system använder både rsyslog och journald.

### **Övervakningsverktyg**

#### **top / htop**

Visar realtidsinformation om CPU-, RAM- och processanvändning.

* top – inbyggt, konsolbaserat
* htop – mer visuellt och användarvänligt (kan installeras med sudo apt install htop)

#### **uptime och load average**

Visar systemets drifttid och belastning:

uptime

#### **vmstat, iostat, netstat (för mer avancerad analys)**

Ger information om minnesanvändning, I/O och nätverkstrafik.

#### **fail2ban – skydd mot brute-force**

Övervakar loggar (t.ex. /var/log/auth.log) och blockerar IP-adresser som misslyckas med inloggningar upprepade gånger.

Installera med:

sudo apt install fail2ban

#### **AIDE – Advanced Intrusion Detection Environment**

Skapar en databas över filsystemets tillstånd (storlek, rättigheter, innehåll) och kan sedan jämföra om något har ändrats (t.ex. attacker som ändrar konfigfiler eller skadlig kod som lägger till bakdörrar).

Installera:

sudo apt install aide

Initiera databas:

sudo aideinit

Verifiera ändringar:

sudo aide --check

Tips: Loggning är inte bara för forensik. Bra loggar = snabbare felsökning och upptäckt av problem innan de blir allvarliga.

### **9.10.1 Kontrollfrågor**

1. Vad är syftet med loggning i ett Linux-system?
2. Vilken kommandorad används för att visa systemd-journalen?
3. Vad är skillnaden mellan rsyslog och journald?
4. Nämn tre vanliga loggfiler i /var/log/.
5. Vad gör top respektive htop?
6. Hur kan du kontrollera hur länge ett system har varit igång?
7. Vad är fail2ban, och hur fungerar det?
8. Vad är AIDE och hur används det?
9. Hur kontrollerar man om AIDE har upptäckt några ändringar?
10. Varför är övervakning viktig även i system som fungerar "som de ska"?

### **9.10.2 Fördjupningslänkar**

* Ubuntu – Log Files Explained  
   <https://help.ubuntu.com/community/LinuxLogFiles>
* journalctl manpage  
   <https://man7.org/linux/man-pages/man1/journalctl.1.html>
* DigitalOcean – Fail2Ban Tutorial  
   <https://www.digitalocean.com/community/tutorials/how-to-protect-ssh-with-fail2ban-on-ubuntu-20-04>
* AIDE – Officiell webbplats  
   <https://aide.github.io/>
* Ubuntu – Monitor System Performance  
   <https://ubuntu.com/server/docs/monitoring>

### **9.10.3 Egna anteckningar**

*(Här kan du skriva ner egna exempel, kommandon du testat eller viktiga lärdomar om loggning och övervakning.)*

## **9.11** **Härdning av tjänster (t.ex. SSH, brandvägg, TLS/SSL)**

Härdning innebär att minska attackytan för ett system genom att stänga av onödiga funktioner, säkra konfigurationer och begränsa åtkomst. Målet är att göra det så svårt som möjligt för en angripare att ta sig in – och så lätt som möjligt att upptäcka försök.

Här tittar vi närmare på några vanliga tjänster och hur du säkrar dem.

### **SSH – Secure Shell**

SSH används för att fjärransluta till Linux-system. Det är kraftfullt, men också ett vanligt mål för attacker.

**Rekommenderade inställningar:**

* Byt standardport från **22** till något annat (för att undvika skriptade attacker)

sudo nano /etc/ssh/sshd\_config  
# Port 2222 (t.ex.)

* Stäng av inloggning för root:

PermitRootLogin no

* Aktivera endast inloggning med SSH-nycklar:

PasswordAuthentication no

* Begränsa inloggning till specifika användare:

AllowUsers robin adminuser

Efter ändring:

sudo systemctl restart ssh

Tips: Använd fail2ban för att blockera upprepade felaktiga inloggningsförsök via SSH.

### **Brandvägg – ufw (Uncomplicated Firewall)**

En brandvägg begränsar nätverkstrafik in och ut från systemet. ufw är ett enkelt gränssnitt till iptables.

**Kom igång:**

sudo apt install ufw  
sudo ufw enable

**Vanliga kommandon:**

* Tillåt SSH (t.ex. port 22):

sudo ufw allow 22/tcp

* Blockera all inkommande trafik som inte är tillåten:

sudo ufw default deny incoming  
sudo ufw default allow outgoing

* Visa status:

sudo ufw status verbose

Tips: Lägg till regler FÖRE du aktiverar brandväggen, annars kan du låsa ute dig själv från SSH.

### **TLS/SSL – Krypterad kommunikation**

TLS/SSL används för att kryptera webbtrafik (https), e-post (smtps/imaps) m.m. Att använda korrekt konfigurerade certifikat är avgörande.

**Gratiscertifikat med Let's Encrypt:**

sudo apt install certbot python3-certbot-apache  
sudo certbot --apache

För Nginx, byt --apache mot --nginx.

**Fördelar med TLS/SSL:**

* Skyddar data från att avlyssnas
* Möjliggör identitetsverifiering via certifikat
* Ger tillgång till HTTPS (vilket är ett krav för moderna webbläsare)

**Tänk på:**

* Ställ in automatisk förnyelse:

sudo systemctl status certbot.timer

### **9.11.1** **Kontrollfrågor**

1. Vad innebär det att "härda" ett system?
2. Varför är det bra att byta port för SSH från 22?
3. Vad gör inställningen PermitRootLogin no?
4. Hur fungerar SSH-nycklar som autentisering?
5. Vad är ufw, och vad används det till?
6. Hur ser du vilka regler som är aktiva i ufw?
7. Vad innebär ufw default deny incoming?
8. Vad är TLS/SSL och vad används det till?
9. Hur skaffar man ett gratis SSL-certifikat i Ubuntu?
10. Varför är det viktigt att förnya TLS-certifikat?

### **9.11.2** **Fördjupningslänkar**

* SSH Hardening Guide (Ubuntu)  
   <https://ubuntu.com/server/docs/service-openssh>
* UFW Firewall Tutorial  
   <https://help.ubuntu.com/community/UFW>
* Let's Encrypt (gratis TLS-certifikat)  
   <https://letsencrypt.org/>
* Mozilla SSL Configuration Generator  
   <https://ssl-config.mozilla.org/>
* OWASP – TLS Best Practices  
   <https://owasp.org/www-project-cheat-sheets/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html>

### **9.11.3** **Egna anteckningar**

*(Här kan du skriva ner egna exempel på härdning du testat, vilka regler du satt upp i ufw, eller inställningar du gjort i sshd\_config.)*

## **9.12** **Backup och återställning**

Att göra regelbundna säkerhetskopior (backuper) är en av de viktigaste säkerhetsåtgärderna du kan ta – men också en av de mest bortglömda. I Linux är det extra viktigt eftersom många servrar är kritiska för verksamheten och inte har något grafiskt gränssnitt som påminner om att "säkerhetskopiera nu".

Backup handlar inte bara om att skydda sig mot hårddiskkrascher – utan även mot ransomware, oavsiktlig radering, konfigurationsfel och attacker.

### **Typer av backup**

|  |  |  |
| --- | --- | --- |
| **Typ** | **Förklaring** | **Exempel** |
| Full backup | All data kopieras | Hela /home eller hela /etc |
| Inkrementell | Endast ändringar sedan senaste backup | Tids- och utrymmeseffektiv |
| Differentierad | Ändringar sedan senaste full backup | En kompromiss mellan ovan |

### **Verktyg i Linux**

Här är några vanliga verktyg för backup i Linux:

* **rsync** – Kopierar filer och kataloger effektivt (lokalt eller över nätverk)
* **tar** – Används för att skapa arkivfiler (.tar/.gz)
* **cron** – Schemalägger backupjobb
* **Deja Dup** – Enkel GUI-lösning för Ubuntu Desktop
* **Bacula / Duplicity / BorgBackup** – Mer avancerade backup-lösningar

### **Exempel på enkel backup med rsync**

Säkerhetskopiera en hemkatalog till en extern disk:

rsync -avh --delete /home/robin /media/backupdisk/

Beskrivning:

* -a: Arkivläge (bevarar rättigheter, ägarskap)
* -v: Verbos (visa vad som görs)
* -h: Mänskligt läsbart format
* --delete: Tar bort filer på backupmål som inte längre finns på källan (för att spegla exakt)

### **Schemalägga backup med cron**

Redigera crontab:

crontab -e

Lägg till en rad för att köra backup varje natt kl 03:00:

0 3 \* \* \* rsync -avh /home/robin /media/backupdisk/

### **Återställning – vad gör du när det smäller?**

En backup är bara användbar om du vet hur du återställer den. Testa därför återställning regelbundet!

**Återställningsexempel:**

rsync -avh /media/backupdisk/robin /home/

Tips: Dokumentera alltid var backuper sparas, hur de återställs och vem som ansvarar för dem.

### **Offsite & molnbackup**

Om backuper endast sparas lokalt riskerar de att förstöras i brand, stöld eller översvämning.

**Lösning:**

* Synka till en annan fysisk plats
* Använd molnlagring (ex. rclone med Google Drive, S3, etc.)
* Kryptera känslig data innan molnuppladdning

### **Vanliga misstag**

* Backupen sparas på samma disk som originalet
* Backupen körs – men återställning testas aldrig
* Backupen innehåller bara data – men inte konfiguration
* Backupskriptet loggar inte något

Bästa praxis: Full backup + inkrementell + offsite + återställningstest

### **9.12.1** **Kontrollfrågor**

1. Vad är skillnaden mellan full och inkrementell backup?
2. Vilket verktyg i Linux används för effektiv filsynkronisering?
3. Hur används cron för backup?
4. Vad gör flaggan --delete i rsync?
5. Varför är det viktigt att även göra backup på konfigurationsfiler?
6. Vad innebär offsite-backup?
7. Vad är risken med att bara ha backup lokalt?
8. Varför är återställningstester viktiga?
9. Ge ett exempel på en molnlösning som kan användas med rclone.
10. Vad kan gå fel om du inte loggar dina backupjobb?

### **9.12.2** **Fördjupningslänkar**

* Ubuntu rsync guide  
   <https://help.ubuntu.com/community/rsync>
* Linux tar command guide  
   <https://linuxize.com/post/how-to-create-and-extract-archives-using-the-tar-command-in-linux/>
* Cron syntax explained  
   <https://crontab.guru/>
* BorgBackup (säkert och komprimerat backupverktyg)  
   <https://www.borgbackup.org/>
* Rclone (synka till molnet)  
   <https://rclone.org/>

### **9.12.3** **Egna anteckningar**

*(Här kan du skriva hur du konfigurerat dina egna backupscript, vilka kataloger du prioriterar, eller testa att återskapa ett gammalt tar-arkiv.)*

## **9.13** **Automatisering av säkerhetsuppgifter (cron, script, loggning)**

Att automatisera återkommande uppgifter i Linux är en av de största fördelarna med systemet – och det gäller särskilt när det handlar om säkerhet. Genom att schemalägga uppgifter som uppdateringar, loggrensning, säkerhetskopiering eller övervakning kan du minimera risken för den mänskliga faktorn och säkerställa att skyddet är aktivt dygnet runt.

### **cron – Linux inbyggda schemaläggare**

cron används för att köra kommandon vid återkommande tidpunkter.

**Struktur i crontab:**

\* \* \* \* \* kommando  
│ │ │ │ │  
│ │ │ │ └─ dag i veckan (0-6, söndag=0)  
│ │ │ └─── månad (1-12)  
│ │ └───── dag i månaden (1-31)  
│ └─────── timme (0-23)  
└───────── minut (0-59)

**Exempel:**  
 Kör ett script varje natt kl 02:00:

0 2 \* \* \* /home/admin/scripts/update-check.sh

Tips: Använd crontab -e för att redigera användarens schema.

### **Bash-script – Automatisering i praktiken**

Script låter dig skapa anpassade säkerhetsrutiner.

**Exempel: Automatiserat uppdateringsscript**

#!/bin/bash  
echo "Uppdatering startade: $(date)" >> /var/log/uppdatering.log  
apt update && apt upgrade -y >> /var/log/uppdatering.log 2>&1  
echo "Färdigt: $(date)" >> /var/log/uppdatering.log

Spara som /home/admin/scripts/update-check.sh och gör körbar:

chmod +x update-check.sh

Lägg till i cron:  
 0 3 \* \* \* /home/admin/scripts/update-check.sh

### **Loggning – Sätt att hålla koll på vad som händer**

I Linux loggas det mesta, särskilt sådant som rör systemet och säkerhet.

Viktiga loggfiler:

* /var/log/auth.log – inloggningar och sudo-kommandon
* /var/log/syslog – generell systemlogg
* /var/log/ufw.log – brandväggsloggar (om UFW används)
* /var/log/apt/history.log – installationshistorik

Visa senaste 20 rader i auth.log:

tail -n 20 /var/log/auth.log

Övervaka logg i realtid:

tail -f /var/log/syslog

### **Automatiserad övervakning (med t.ex. fail2ban)**

fail2ban är ett verktyg som automatiskt blockerar IP-adresser som upprepat försöker logga in felaktigt – t.ex. via SSH.

Installation:

sudo apt install fail2ban

Starta och aktivera:

sudo systemctl enable --now fail2ban

Logg:

sudo cat /var/log/fail2ban.log

fail2ban använder loggfiler som auth.log för att identifiera attacker.

### **Rekommenderade säkerhetsrutiner att automatisera**

|  |  |  |  |
| --- | --- | --- | --- |
| **Åtgärd** | **Verktyg** | **Frekvens** | **Exempel** |
| Systemuppdatering | apt + cron | Dagligen | update-check.sh |
| Backup | rsync/tar + cron | Dagligen/veckovis | backup.sh |
| Loggrensning | logrotate | Veckovis | /etc/logrotate.d/ |
| Säkerhetskontroll | script + cron | Veckovis | T.ex. kontrollera öppna portar |
| Antivirusgenomsökning | clamav + cron | Veckovis | clamscan -r /home |

### **9.13.1** **Kontrollfrågor**

1. Vad är cron och vad används det till?
2. Vad betyder 0 3 \* \* \* i crontab?
3. Hur gör man ett script körbart?
4. Vad gör kommandot tail -f /var/log/syslog?
5. Varför är loggning viktigt för säkerheten?
6. Vad gör fail2ban?
7. Var loggas SSH-inloggningar?
8. Vad är skillnaden mellan apt update och apt upgrade?
9. Varför är det bra att automatisera säkerhetsuppgifter?
10. Nämn tre säkerhetsuppgifter du kan schemalägga med cron.

### **9.13.2** **Fördjupningslänkar**

* Cron syntax och guide  
   <https://crontab.guru/>
* Linux Bash Scripting Tutorial  
   <https://ryanstutorials.net/bash-scripting-tutorial/>
* Fail2ban – officiell dokumentation  
   <https://www.fail2ban.org/wiki/index.php/Main_Page>
* Ubuntu Log files explained  
   <https://help.ubuntu.com/community/LinuxLogFiles>
* Automatiserad säkerhet i Linux  
   <https://www.tecmint.com/automate-linux-tasks-using-cron/>

### **9.13.3** **Egna anteckningar**

*(Här kan du skriva vilka säkerhetsuppgifter du har automatiserat, lägga in utdrag från din egen crontab, eller testa att skapa ett eget säkerhetsscript.)*

## **9.14 Säkerhetsverktyg i Linux**

Att skydda ett Linuxsystem kräver mer än bara bra konfigurationer – det kräver också rätt verktyg. I denna del samlar vi några av de mest använda säkerhetsverktygen i Linuxvärlden. Vissa har vi redan introducerat tidigare, men här får de en gemensam överblick tillsammans med nya alternativ som kompletterar bilden.

#### **Översikt: Verktyg för olika ändamål**

|  |  |  |
| --- | --- | --- |
| **Typ av verktyg** | **Exempel** | **Syfte** |
| Intrångsskydd (brute-force) | fail2ban | Blockerar IP-adresser efter upprepade misslyckade inloggningsförsök |
| Brandväggsadministration | ufw, nftables | Kontrollerar och filtrerar nätverkstrafik |
| Rotskalsdetektering | chkrootkit, rkhunter | Söker efter rootkits och systemmanipulationer |
| Sårbarhetsanalys | Lynis | Gör en systemskanning och ger säkerhetsrekommendationer |
| Antivirusskydd | ClamAV | Letar efter virus, trojaner och skadlig kod |
| Integritetsövervakning | AIDE, Tripwire | Övervakar förändringar i filer och system |
| Övervakning & loggning | auditd, logwatch | Granskar aktiviteter och skapar rapporter |

#### **Exempel på verktyg**

##### **1. fail2ban – Skydd mot bruteforce**

Bevakar loggar (t.ex. SSH) och blockerar IP-adresser med för många inloggningsförsök.  
 Redan konfigurerat i många Debian-baserade system med /etc/fail2ban/jail.conf.

sudo apt install fail2ban  
sudo systemctl enable --now fail2ban

##### **2. ufw – Enkel brandväggshantering**

Wrapper för iptables/nftables. Bra för mindre miljöer eller snabb konfigurering.

sudo apt install ufw  
sudo ufw enable  
sudo ufw allow ssh

##### **3. chkrootkit och rkhunter – Rootkitsökning**

Söker efter tecken på att systemet har blivit komprometterat. De har olika signaturbaser så de kompletterar varandra.

sudo apt install chkrootkit rkhunter  
sudo chkrootkit  
sudo rkhunter --check

##### **4. Lynis – Sårbarhetsanalys**

Gör en säkerhetsgenomgång och listar potentiella risker i ett rapportformat.

sudo apt install lynis  
sudo lynis audit system

Lynis används ofta som en del av hårdvaru- och säkerhetsrevisioner.

##### **5. ClamAV – Antivirusscanner för Linux**

Bra för att analysera exempelvis e-postservrar eller användarkataloger. Inte realtid, men kan schemaläggas.

sudo apt install clamav  
sudo freshclam # uppdatera virussignaturer  
sudo clamscan -r /home

##### **6. AIDE – Övervakning av filsystemets integritet**

Jämför aktuella filer mot en sparad databas för att upptäcka oväntade förändringar.

sudo apt install aide  
sudo aideinit  
sudo cp /var/lib/aide/aide.db.new /var/lib/aide/aide.db  
sudo aide --check

##### **7. auditd – Systemgranskning på kernel-nivå**

Skapar detaljerade loggar över användaraktiviteter, ändringar i filer och program som körs.

sudo apt install auditd  
sudo systemctl start auditd

Loggar sparas i /var/log/audit/.

### **9.14.1 Kontrollfrågor**

1. Vad gör fail2ban och vilket typ av angrepp skyddar det mot?
2. Hur skiljer sig chkrootkit från rkhunter?
3. Vad är syftet med Lynis och hur används det?
4. Vilket verktyg använder du för att kontrollera integriteten hos filer i ett Linuxsystem?
5. Hur fungerar ClamAV och vad är det bra på?
6. Vad är skillnaden mellan realtidsskydd och schemalagda skanningar?
7. Vad är auditd och vad loggar det?
8. Varför kan det vara bra att använda både chkrootkit och rkhunter?
9. När är det lämpligt att använda AIDE eller Tripwire?
10. Hur kan dessa verktyg kombineras för att ge ett heltäckande skydd?

### **9.14.2 Fördjupningslänkar**

* CIS – "Linux Security Best Practices":  
   <https://www.cisecurity.org/benchmark/linux>
* Debian Wiki – Hardening and Security Tools:  
   <https://wiki.debian.org/Hardening>
* ClamAV – Official Documentation:  
   <https://docs.clamav.net/>
* RKHunter – Rootkit Hunter Project:  
   <http://rkhunter.sourceforge.net/>
* Lynis GitHub Repository:  
   <https://github.com/CISOfy/lynis>

### **9.14.3 Egna anteckningar**

(Lämna utrymme för reflektioner, kommandon du vill testa eller egna tips.)

## **9.15** **IT-säkerhet ur ett verksamhetsperspektiv**

### **9.15.1 Säkerhetspolicy och rutiner**

IT-säkerhet handlar inte bara om brandväggar, lösenord och antivirusprogram. För att säkerheten verkligen ska fungera i en organisation krävs tydliga riktlinjer, regler och ansvar. Här kommer säkerhetspolicyn in – ett dokument som fungerar som ett ramverk för hur en verksamhet hanterar sina resurser och skyddar sin information.

En **säkerhetspolicy** är ofta ett övergripande dokument som beskriver hur organisationen ser på säkerhet, vilka mål som finns, och vilka krav som ställs på medarbetare, system och processer. Den fungerar lite som ett säkerhetskontrakt: så här jobbar vi, det här förväntar vi oss av dig, och det här händer om något går fel.

Till policyn kopplas ofta **rutiner**, **instruktioner** och **riktlinjer**. Dessa går mer på detaljnivå och svarar på frågor som:

* Hur ofta ska vi ta backup?
* Hur skapar vi nya användare?
* Vad händer om någon tappar bort sin dator?

En vanlig missuppfattning är att det räcker att *ha* en policy – men det är först när policyn är **känd, förstådd och efterlevd** som den gör nytta. En policy som ligger i en mapp någonstans på intranätet och aldrig uppdateras är i praktiken värdelös.

#### **Exempel på vanliga policytyper:**

* **Lösenordspolicy** – regler för längd, komplexitet, giltighetstid
* **Användarpolicy** – vad får du göra (och inte göra) med din jobbmail eller jobbdator?
* **Backup-policy** – hur ofta tas backup, var lagras den, vem ansvarar?
* **Mobilpolicy** – hur hanteras bärbara enheter och BYOD (Bring Your Own Device)?

En viktig poäng är att policyn måste **anpassas till verksamheten**. Ett stort företag med många system och roller behöver andra regler än en mindre förening. Det finns ingen universallösning – säkerhetspolicyn måste spegla organisationens behov, risknivå och tekniska miljö.

Slutligen bör varje säkerhetspolicy följas upp: är den fortfarande relevant? Följs den? Och vad kan förbättras? En bra policy är ett levande dokument, inte ett dammigt Word-dokument från 2013.

#### **9.15.1.1 Kontrollfrågor**

1. Vad är syftet med en säkerhetspolicy?
2. Vad är skillnaden mellan en policy, en rutin och en instruktion?
3. Varför räcker det inte att bara ha en policy?
4. Vad innebär det att en policy måste vara känd och efterlevd?
5. Nämn tre exempel på vanliga typer av IT-säkerhetspolicys.
6. Vad är en lösenordspolicy till för?
7. Vad är en backup-policy och varför är den viktig?
8. Vad menas med BYOD och hur påverkar det säkerhetspolicyn?
9. Varför är det viktigt att anpassa säkerhetspolicyn till verksamheten?
10. Hur bör en organisation följa upp och förbättra sin säkerhetspolicy?

#### **9.15.1.2 Fördjupningslänkar**

1. MSB – Säkerhetspolicy – grunder och exempel  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/ledningssystem/sakerhetspolicy/>
2. DIGG – Informationssäkerhet för offentlig sektor  
    <https://www.digg.se/utveckling--innovation/informationssakerhet>
3. PTS – Informationssäkerhet och policyarbete  
    <https://www.pts.se/sv/privat/informationssakerhet/>
4. NIST – Security Policies and Implementation Issues (eng)  
    <https://csrc.nist.gov/publications/detail/sp/800-53/rev-5/final>
5. SANS Institute – Creating a Security Policy (eng)  
    <https://www.sans.org/security-resources/policies/>

#### **9.15.1.3 Egna anteckningar**

*(Här kan du anteckna egna reflektioner, exempel eller frågor kring säkerhetspolicys och rutiner.)*

### **9.15.2 Användarutbildning och medvetenhet**

När man pratar om IT-säkerhet är det lätt att fokusera på teknik – brandväggar, kryptering, antivirusprogram. Men en av de största riskerna i ett IT-system är inte teknik. Det är människor.

Användaren är ofta den svagaste länken i säkerhetskedjan. Det spelar ingen roll hur avancerad säkerhetslösning du har, om någon klickar på en phishing-länk, lämnar sitt lösenord på en post-it-lapp, eller tar med jobbdatorn på semester och glömmer den på tåget.

Därför är **utbildning och medvetenhet** en central del av säkerhetsarbetet. Syftet är inte att göra alla till IT-experter – utan att ge dem tillräcklig kunskap för att känna igen hot, förstå konsekvenser och agera ansvarsfullt i vardagen.

#### **Exempel på vad utbildning bör täcka:**

* Vad är ett starkt lösenord?
* Hur känner man igen phishing-mejl?
* Vad innebär social engineering?
* Hur hanteras USB-minnen och bärbara enheter?
* Varför är det viktigt att låsa datorn när man går därifrån?

Utbildningen ska vara **återkommande**. Det räcker inte att ha en timmes föreläsning en gång om året. Det bör finnas korta påminnelser, interna kampanjer, tester och gärna exempel från verkligheten. Målet är att skapa en **säkerhetskultur** – där säkerhet är något alla tänker på, hela tiden.

#### **Ett vanligt misstag:**

Att tro att användare *borde fatta*. Det gör de inte alltid – inte för att de är dumma, utan för att de inte fått rätt förutsättningar. Att skälla på någon för att de klickade på fel länk är som att bli arg på någon som inte kan cykla – det är bättre att lära dem cykla.

**Tips:** Låt gärna användarutbildning innehålla lite humor, storytelling och interaktivitet. Människor minns känslor och berättelser bättre än listor med regler.

#### **9.15.2.1 Kontrollfrågor**

1. Varför anses användaren ofta vara den svagaste länken i säkerhetskedjan?
2. Vad är målet med användarutbildning inom IT-säkerhet?
3. Nämn tre ämnen som bör ingå i en säkerhetsutbildning för användare.
4. Vad är social engineering?
5. Varför är det inte tillräckligt med en engångsutbildning i säkerhet?
6. Vad menas med säkerhetskultur?
7. Varför är det bättre att utbilda än att skälla?
8. Ge ett exempel på hur man kan öka säkerhetsmedvetenhet i vardagen.
9. Vad bör man göra om man misstänker att man klickat på något farligt?
10. Vad är poängen med att använda humor eller storytelling i utbildning?

#### **9.15.2.2 Fördjupningslänkar**

1. MSB – Säkerhetskultur i praktiken  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/sakerhetskultur/>
2. CERT-SE – Vanliga säkerhetshot mot användare  
    <https://www.cert.se/>
3. Phishing.org – What is Phishing? (eng)  
    <https://www.phishing.org/what-is-phishing>
4. KnowBe4 – Security Awareness Training (eng)  
    <https://www.knowbe4.com/security-awareness-training>
5. BleepingComputer – Security Tips for Users (eng)  
    <https://www.bleepingcomputer.com/>

#### **9.15.2.3 Egna anteckningar**

*(Här kan du skriva ner egna reflektioner, idéer för utbildning, eller exempel på incidenter som hade kunnat undvikas med rätt kunskap.)*

### **9.15.3** **Roller, ansvar och säkerhetsorganisation**

En säker IT-miljö uppstår inte av sig själv – den måste byggas, underhållas och kontrolleras. Därför krävs tydligt definierade **roller och ansvar**. Alla i en organisation – från vd till supportpersonal – har en roll att spela i säkerhetsarbetet.

### **Centrala roller inom IT-säkerhet:**

|  |  |
| --- | --- |
| **Roll** | **Ansvar** |
| **Systemadministratör** | Konfigurerar och underhåller system, ser till att uppdateringar sker, hanterar användare och rättigheter. |
| **Informationssäkerhetsansvarig (CISO)** | Övergripande ansvar för att organisationens information är skyddad. Tar fram policies, följer upp incidenter. |
| **IT-säkerhetsspecialist** | Analyserar hot, genomför tester, arbetar praktiskt med säkerhetsverktyg. |
| **Supportpersonal** | Första linjens försvar – möter användare, svarar på frågor, upptäcker ofta avvikelser först. |
| **Alla användare** | Ska följa regler, vara vaksamma och rapportera misstänkta händelser. |

En **tydlig ansvarsfördelning** minskar risken för att något glöms bort eller faller mellan stolarna. Det bör också finnas dokumentation som klargör:

* Vem får göra vad?
* Vem ansvarar för säkerhetskopiering?
* Vem får ändra brandväggsregler?

**Tips:** Roller och ansvar kan anpassas efter organisationens storlek – i ett mindre företag kan samma person ha flera roller, men det får aldrig råda oklarhet om *vem* som har *vilket ansvar*.

#### **9.15.3.1 Kontrollfrågor**

1. Varför är det viktigt med tydligt definierade roller i IT-säkerhetsarbetet?
2. Vad har en systemadministratör för ansvar inom säkerhet?
3. Vad gör en CISO?
4. Varför är supportpersonal viktiga i säkerhetskedjan?
5. Vad kan hända om ingen har ansvar för t.ex. säkerhetskopiering?
6. Ge ett exempel på hur ansvarsfördelning kan se ut i ett mindre företag.
7. Varför är det viktigt att dokumentera vem som får göra vad?
8. Vad är användarens ansvar i en säkerhetsorganisation?
9. Hur kan ansvarsfördelning bidra till bättre incidenthantering?
10. Kan en person ha flera roller? Vad krävs då för att det ska fungera?

#### **9.15.3.2 Fördjupningslänkar**

1. MSB – Ansvarsfördelning inom informationssäkerhet  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/styrning-och-ledning/roller-och-ansvar/>
2. CISO Role Explained – Techtarget (eng)  
    <https://www.techtarget.com/searchsecurity/definition/CISO>
3. SANS Institute – Security Roles and Responsibilities (eng)  
    <https://www.sans.org/white-papers/role-based-security-training/>
4. IT Governance – Building a Security Team (eng)  
    <https://www.itgovernance.co.uk/blog/how-to-build-an-information-security-team>
5. CERT-SE – Säkerhetsfunktioner i organisationer  
    <https://www.cert.se/>

#### **9.15.3.3 Egna anteckningar**

*(Skriv ner exempel från skolan, företagsmiljöer eller projekt du deltagit i. Har ni tydliga roller?)*

### **9.15.4** **Rutiner och beredskap vid incidenter**

Förr eller senare händer det – en incident. Det kan vara ett virus, ett konto som blivit kapat, en förlorad dator, eller en DDoS-attack. Frågan är inte *om* något händer, utan *hur väl förberedd man är när det gör det*.

Därför måste det finnas **rutiner och beredskap**. Att veta i förväg hur man ska agera sparar tid, minskar skada och kan förhindra spridning.

### **Några grundläggande rutiner:**

* **Incidentrapportering**: Vem ska man kontakta? Hur snabbt? Vad ska rapporteras?
* **Första åtgärder**: Koppla ur från nätverket? Informera andra?
* **Loggning**: Vad har hänt, när, och vem var inblandad?
* **Eskaleringsvägar**: När går ärendet från support till säkerhetsansvarig?

Det bör finnas en **incidentplan**, gärna som en del av en större säkerhetspolicy. I större organisationer har man ofta en **CSIRT** – Computer Security Incident Response Team – som hanterar allvarliga incidenter.

En bra incidenthantering innehåller också **lärdomar**. Efter varje incident bör man fråga:

* Vad gick fel?
* Vad kunde ha gjorts bättre?
* Hur kan vi förhindra att det händer igen?

💡 **Tips:** Det kan vara smart att göra en “brandövning” för IT – en simulerad incident där man tränar personalens reaktion och ser om rutinerna fungerar.

#### **9.15.4.1 Kontrollfrågor**

1. Vad menas med en IT-incident?
2. Varför är det viktigt att ha rutiner innan något händer?
3. Vad bör en incidentrapport innehålla?
4. Vad är det första man bör göra vid en allvarlig incident?
5. Vad är en CSIRT och vad gör de?
6. Vad menas med “eskalering” i incidenthantering?
7. Varför är loggning viktig under och efter en incident?
8. Vad bör man göra efter att en incident är löst?
9. Vad är syftet med att göra en “IT-brandövning”?
10. Ge exempel på en mindre och en allvarlig IT-incident.

#### **9.15.4.2 Fördjupningslänkar**

1. MSB – Hantering av IT-incidenter  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/incidenthantering/>
2. CERT-SE – Rapportera IT-incident  
    <https://www.cert.se/rapporteringsformularet/>
3. CISA – Incident Response Playbook (eng)  
    <https://www.cisa.gov/sites/default/files/publications/IR-Playbook.pdf>
4. ENISA – Guidelines for Incident Management (eng)  
    <https://www.enisa.europa.eu/publications/guidelines-for-incident-management>
5. NIST – Computer Security Incident Handling Guide (eng)  
    <https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final>

#### **9.15.4.3 Egna anteckningar**

*(Har du varit med om ett IT-problem? Hur löstes det? Vad lärde du dig?)*

### **9.15.5** **Säkerhetspolicy och dokumentation**

En säkerhetspolicy är inte bara ett dokument man skriver för att lägga i en pärm. Det är ett **levande styrdokument** som beskriver hur organisationen tänker kring säkerhet – vad som är tillåtet, förbjudet, rekommenderat och krävs.

### **En bra säkerhetspolicy innehåller:**

* **Syfte och mål**: Varför behövs denna policy?
* **Roller och ansvar**: Vem gör vad?
* **Regler för användning**: Vad gäller för lösenord, nätverk, fjärranslutning, e-post m.m.?
* **Incidenthantering**: Hur ska problem rapporteras?
* **Dokumentation och uppföljning**: Hur kontrolleras efterlevnad?

Alla policies ska vara **begripliga, tillgängliga och uppdaterade**. Ingen ska behöva gissa vad som gäller – det ska stå i klartext.

Det bör också finnas **loggar, rapporter och checklistor**. Inte för att “övervaka” i första hand, utan för att säkerställa att man gör det man sagt att man ska göra.

**Tips:** Gör säkerhetspolicyn synlig – prata om den, ta upp den på möten, och se till att nya användare får läsa och förstå den direkt vid anställning eller skolstart.

#### **9.15.5.1 Kontrollfrågor**

1. Vad är en säkerhetspolicy?
2. Varför är det viktigt att den är uppdaterad och tydlig?
3. Vad bör en säkerhetspolicy innehålla?
4. Hur används en policy i det dagliga arbetet?
5. Vad menas med att en policy ska vara ett “levande dokument”?
6. Varför är det viktigt att alla användare känner till policyn?
7. Vad innebär uppföljning och hur kan den göras?
8. Ge exempel på något som kan stå i en policy.
9. Hur kan dokumentation hjälpa vid en incident?
10. När ska en policy revideras?

#### **9.15.5.2 Fördjupningslänkar**

1. MSB – Exempel på säkerhetspolicy  
    <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/styrning-och-ledning/policys-och-rutiner/>
2. Internetstiftelsen – IT-policy mall  
    <https://internetstiftelsen.se/guide/sa-skriver-du-en-it-policy/>
3. NIST – Policy Frameworks (eng)  
    <https://csrc.nist.gov/publications/sp800>
4. GDPR och dokumentation  
    <https://www.imy.se/lagar--regler/dataskyddsforordningen/>
5. IT Governance – How to write an Information Security Policy (eng)  
    <https://www.itgovernance.co.uk/blog/how-to-write-an-information-security-policy>

#### **9.15.5.3 Egna anteckningar**

*(Finns det en policy där du är? Har du läst den? Vad skulle du vilja ändra eller lägga till?)*

### **9.15.6 Juridik och regelverk**

IT-säkerhet är inte bara en teknisk fråga – det är också en juridisk skyldighet. Organisationer måste förhålla sig till ett antal lagar och regelverk som styr hur information hanteras, skyddas och rapporteras. Att inte följa dessa kan få allvarliga konsekvenser, både ekonomiskt och juridiskt.

En av de mest centrala lagarna är **GDPR** – Dataskyddsförordningen – som styr hur personuppgifter får behandlas. Enligt GDPR måste organisationer ha tydliga syften med insamling av personuppgifter, de får bara lagras så länge som det är nödvändigt, och de måste skyddas från obehörig åtkomst. Dessutom har individen rättigheter – som att få ut sin data, få den rättad eller raderad.

En annan viktig lagstiftning är **NIS2-direktivet**, som är en uppföljare till det första NIS-direktivet. Det gäller särskilt för samhällsviktiga verksamheter som energi, transporter, sjukvård och digital infrastruktur. NIS2 ställer krav på både förebyggande säkerhetsåtgärder och incidentrapportering. Organisationer som omfattas måste rapportera allvarliga IT-incidenter till myndighet inom 24 timmar.

För verksamheter som hanterar känslig eller säkerhetsskyddad information – till exempel kommuner, statliga myndigheter eller företag med försvarsanknytning – gäller även **Säkerhetsskyddslagen**. Denna lag reglerar hur man skyddar information som kan vara viktig för Sveriges säkerhet, och omfattar bland annat behörighetskontroller, klassificering av information och säkerhetsprövningar.

En annan viktig juridisk aspekt är **ansvarsfördelningen**: vem är ansvarig för vad? I dataskyddssammanhang skiljer man på **personuppgiftsansvarig** (t.ex. en skola eller ett företag som samlar in data) och **personuppgiftsbiträde** (t.ex. en molntjänstleverantör som behandlar datan åt någon annan). Det är alltid den personuppgiftsansvarige som har det yttersta ansvaret för att reglerna följs – även om själva behandlingen sker hos någon annan.

Att bryta mot dessa regelverk kan få allvarliga konsekvenser. Inom ramen för GDPR kan till exempel sanktionsavgifter på flera miljoner kronor utdelas, även för mindre verksamheter. Utöver det kan bristande efterlevnad leda till förlorat förtroende, rättsprocesser eller utestängning från samarbeten och upphandlingar.

Att ha koll på juridiken är alltså inte en bonus – det är ett grundkrav för varje verksamhet som vill arbeta seriöst med IT-säkerhet.

#### 9.15.6.1 **Kontrollfrågor**

1. Vad är syftet med GDPR?
2. Vilka typer av organisationer omfattas av NIS2-direktivet?
3. Vad reglerar Säkerhetsskyddslagen?
4. Vad är skillnaden mellan en personuppgiftsansvarig och ett personuppgiftsbiträde?
5. Vilka konsekvenser kan det få om man bryter mot lagar inom IT-säkerhet?

#### 9.15.6.2 **Fördjupningslänkar**

* IMY – Dataskyddsförordningen (GDPR)  
   <https://www.imy.se/lagar--regler/dataskyddsforordningen/>
* EU – NIS2-direktivet (engelska)  
   <https://digital-strategy.ec.europa.eu/en/policies/nis2>
* Säkerhetspolisen – Säkerhetsskyddslagen  
   <https://www.sakerhetspolisen.se/verksamhet/sakerhetsskydd.html>
* MSB – Juridik och informationssäkerhet  
   <https://www.msb.se/sv/amnesomraden/informationssakerhet/ledningssystem/juridik-och-ansvar/>
* GDPR.eu – Guide till dataskyddsförordningen (engelska)  
   <https://gdpr.eu/>

#### 9.15.6.3 **Egna anteckningar** (Har du stött på situationer där juridiken kring IT var viktig? Kände du till rollerna enligt GDPR innan? Skriv ner exempel eller frågor du har.)

## **9.16** **Moderna hot och skydd**

IT-hot förändras ständigt. Nya tekniker, arbetssätt och angripare gör att säkerhetsarbetet måste vara proaktivt och ständigt uppdaterat. I detta avsnitt går vi igenom några av de mest aktuella hoten mot IT-miljöer – och hur man kan skydda sig mot dem.

### **9.16.1 Social engineering och phishing**

Det spelar ingen roll hur stark brandväggen är – om en användare frivilligt släpper in angriparen. Social engineering handlar om att lura människor snarare än att bryta sig igenom tekniska skydd.

Den vanligaste formen är **phishing** – falska mejl eller meddelanden som försöker lura mottagaren att klicka på en länk, lämna ifrån sig lösenord eller installera skadlig kod.

**Exempel på phishing:**  
 – Du får ett mejl från "Skatteverket" som ber dig logga in med BankID.  
 – Du får ett SMS från "PostNord" med en länk till en falsk spårningssida.  
 – Du blir uppringd av någon som påstår sig vara från IT-avdelningen och ber dig ge bort ditt lösenord.

Phishing kan vara massutskick, men också riktade attacker, t.ex. **spear phishing** (mot en viss person) eller **CEO-fraud** (där bedragaren utger sig för att vara chef och ber om en snabb överföring).

**Skydd:**  
 – Utbildning är nyckeln – användare måste veta hur angreppen ser ut.  
 – Använd e-postfilter, multifaktorsautentisering och varna för ovanliga språk eller avsändare.  
 – Tänk efter innan du klickar. Fråga: "Verkar detta rimligt?"

### **9.16.2 Ransomware – utpressningsprogram**

**Ransomware** är skadlig kod som krypterar filer på en dator eller ett nätverk. Därefter kräver angriparen en lösensumma (ofta i kryptovaluta) för att återställa filerna.

Ransomware kan komma via phishingmejl, sårbarheter i tjänster eller via oskyddade RDP-portar. Vissa varianter (s.k. *double extortion*) stjäl först filerna och hotar sedan att publicera dem om man inte betalar.

Exempel på kända ransomware: **WannaCry, REvil, LockBit**.

**Skydd:**  
 – Uppdatera programvara och operativsystem regelbundet  
 – Ta **offline-backuper** – krypterade filer kan inte återställas om även backuperna påverkas  
 – Använd minst MFA och starka lösenord  
 – Öva på incidenthantering – vet ni vad ni ska göra om det händer?

### **9.16.3 Zero-day exploits**

En **zero-day** är en sårbarhet som är okänd för programvarans tillverkare – och som därför inte har någon patch. Den är alltså oskyddad från dag ett.

När sådana sårbarheter upptäcks av angripare, används de ofta i avancerade attacker – innan tillverkaren ens hunnit skapa ett skydd.

**Exempel:** Log4Shell (2021), en allvarlig zero-day i Java-loggning, drabbade miljontals system.

Eftersom zero-days inte kan blockeras via traditionella patchar krävs andra försvar:

**Skydd:**  
 – Minimera exponerad yta (öppna portar, onödiga tjänster)  
 – Använd IDS/IPS-system och loggning för att upptäcka konstig trafik  
 – Använd säkerhetsprinciper som **Least Privilege** – begränsa vad en användare/process kan göra  
 – Håll dig uppdaterad via säkerhetsflöden (t.ex. CERT-SE)

### **9.16.4 Honeypots – lura angriparen**

En **honeypot** är en fälla. Den ser ut som ett riktigt system – t.ex. en SSH-server eller webbserver – men är till för att bli attackerad. Syftet är att:

1. Locka angripare så att de avslöjar sina metoder
2. Hålla dem borta från riktiga system
3. Samla information om vilka hot som finns

**Tänk så här:** Istället för att jaga angriparen – låt dem komma till dig.

I skolmiljö eller labbar används ofta verktyg som **Cowrie** (SSH/Telnet-honeypot) eller **Dionaea** (för att fånga malware).

En honeypot måste isoleras från produktionsnätverket – annars riskerar den att bli en språngbräda.

### **9.16.5 Penetrationstester – simulerade attacker**

Ett **penetrationstest** (eller "pentest") är en kontrollerad attack där en etisk hackare försöker bryta sig in i ett system – på uppdrag av systemägaren. Målet är att hitta svagheter **innan** någon illvillig gör det.

Pentest kan vara manuella eller automatiserade och omfatta t.ex.:

– Sårbarhetsscanning  
 – Försök att utnyttja brister (exploits)  
 – Social engineering-tester  
 – Fysisk intrångstest

**Exempel på verktyg:**  
 – Kali Linux  
 – Metasploit  
 – Nmap  
 – Burp Suite

**Juridik:** Ett pentest får aldrig ske utan godkännande. Etiska hackare följer lagar, dokumenterar allt och lämnar rapport till uppdragsgivaren.

### **9.16.6 Säkerhet i molnet och vid distansarbete**

I takt med att fler arbetar hemifrån och system flyttas till molnet, har även hotbilden förändrats. Säkerhet måste nu fungera **utanför företagets väggar**.

Utmaningar i molnet:  
 – Oklar ansvarsfördelning (vem skyddar vad?)  
 – Dålig konfiguration (t.ex. öppna S3-buckets)  
 – Felaktig hantering av användarbehörigheter

Utmaningar vid distansarbete:  
 – Osäkra Wi-Fi-nät  
 – Privata enheter  
 – Brist på övervakning

**Skyddsåtgärder:**  
 – Använd **Zero Trust**-modellen: Ingen är "automatiskt betrodd"  
 – Se till att MFA används överallt  
 – Använd central identitetshantering  
 – VPN + Endpoint-skydd  
 – Loggning och övervakning även utanför kontorsnätet

Molntjänster kan vara säkra – men bara om de konfigureras rätt och om användarna förstår sitt ansvar.

### **9****.16.7 Kontrollfrågor**

1. Vad är skillnaden mellan phishing och spear phishing?
2. Hur fungerar ransomware, och varför är det farligt?
3. Vad är en zero-day-sårbarhet?
4. Varför är honeypots användbara inom IT-säkerhet?
5. Vad är syftet med ett penetrationstest?
6. Vilka verktyg kan användas vid ett pentest?
7. Vad innebär "Zero Trust"-modellen?
8. Nämn två risker med dåligt konfigurerade molntjänster.
9. Hur kan man skydda sig vid distansarbete?
10. Varför är MFA extra viktigt vid molnanvändning?

### **9.16.8** **Fördjupningslänkar**

– CERT-SE – Aktuella säkerhetshot  
 [https://www.cert.se](https://www.cert.se/) – KnowBe4 – Träning mot phishing  
 [https://www.knowbe4.com](https://www.knowbe4.com/) – NoMoreRansom – Hjälp mot ransomware  
 [https://www.nomoreransom.org](https://www.nomoreransom.org/) – OWASP – Honeypot-projekt  
 <https://owasp.org/www-project-honeypot/> – NIST – Zero Trust Architecture  
 <https://www.nist.gov/publications/zero-trust-architecture>

### **9.16.9** **Egna anteckningar**

(Lämna plats för egna reflektioner, case eller exempel.)

## **9****.17 Verktyg, övervakning och loggning**

Att ha brandväggar, antivirus och uppdateringar är viktigt – men det räcker inte. För att kunna upptäcka angrepp, analysera intrång och förstå vad som händer i ett system behövs **övervakning och loggning**. Detta kapitel går igenom verktygen och tankesätten bakom ett effektivt övervakningssystem.

### **9.17.1 Varför loggning är avgörande**

Loggfiler är systemets minne. De registrerar allt från inloggningar och systemfel till nätverkstrafik och misstänkta aktiviteter. Om något går fel – en server kraschar, ett konto kapas eller en ransomware-attack inträffar – är det loggarna som hjälper oss att förstå vad som hänt.

Utan loggar blir felsökning ofta en gissningslek.

Exempel på viktiga loggfiler i olika miljöer:

* **Windows:** Event Viewer (t.ex. Security, System, Application)
* **Linux:** /var/log/auth.log, /var/log/syslog, /var/log/apache2/\*
* **Nätverk:** brandväggsloggar, routrar, accesspunkter
* **Applikationer:** databasservrar, webbservrar, e-postservrar

**Tips:** Att logga är inte nog – du måste också **samla in**, **analysera** och **spara** loggar på ett säkert sätt.

### **9.17.2 Centrala loggverktyg – Syslog, journald, Event Viewer**

I moderna IT-miljöer är det vanligt att loggar från många olika enheter samlas på ett ställe. Detta kallas **central logghantering**, och gör att man lättare kan upptäcka mönster och hot.

#### **Linux**

* **Syslog**: Ett äldre men fortfarande vanligt system som skriver loggar till /var/log/. Verktyg som rsyslog eller syslog-ng används ofta för att samla in och skicka loggar.
* **journald**: Nyare loggsystem i systemd-baserade distributioner (t.ex. Ubuntu). Kommandot journalctl används för att visa loggar.

#### **Windows**

* **Event Viewer**: Inbyggt verktyg där systemet grupperar loggar i t.ex. System, Security, och Application.
* För central hantering används ofta **Event Forwarding** eller verktyg som **Graylog**, **Wazuh** eller **Elastic Stack (ELK)**.

Loggformaten skiljer sig, men budskapet är detsamma: “Vad hände, när hände det – och varför?”

### **9.17.3 Övervakningsverktyg – Wazuh, Zabbix, Nagios**

Loggning är passivt – det skriver ner vad som händer. **Övervakning** är mer aktivt – den säger till om något händer i realtid.

Exempel på verktyg:

* **Wazuh**: Open source XDR/EDR och SIEM-lösning. Samlar in loggar, upptäcker hot och varnar.
* **Zabbix**: Kraftfullt övervakningssystem som mäter allt från CPU-användning till nätverkstrafik.
* **Nagios**: Klassiker inom serverövervakning. Kan användas för att hålla koll på driftstatus, nätverksresurser, m.m.

De här verktygen kan:

* Skicka e-post eller SMS vid fel
* Rita grafer över trender (t.ex. CPU-belastning)
* Identifiera attacker och ovanlig aktivitet
* Dokumentera SLA och tillgänglighet

Använd flera nivåer: Loggar + övervakning + larm = bäst skydd.

### **9.17.4 SIEM – Samlad analys av säkerhetsloggar**

**SIEM** står för **Security Information and Event Management**. Det är en lösning för att:

* Samla in loggar från många system
* Korrelera händelser (”logga A + logga B = attack”)
* Upptäcka ovanlig aktivitet
* Ge översikt i realtid

Ett SIEM-system fungerar som säkerhetens kontrollrum. Du ser attacker medan de händer, och kan agera direkt.

Exempel på SIEM-verktyg:

* **Wazuh** – inkluderar SIEM-funktionalitet
* **Graylog** – fokuserar på logganalys
* **Splunk** – kommersiellt kraftpaket
* **ELK Stack (Elasticsearch, Logstash, Kibana)** – flexibel och populär i open source-världen

SIEM är särskilt viktigt i större nätverk, skolor, företag och organisationer där många användare och system interagerar.

### **9.17.5 Att tänka på vid övervakning – etik och lagstiftning**

Med stora möjligheter kommer också ansvar. Att övervaka användare, logga trafik och lagra data innebär att man måste:

* Följa **GDPR** och andra lagar
* Informera användare om att loggning sker
* Inte spara mer än nödvändigt
* Skydda loggarna mot manipulation

Etik handlar också om **tillit**. Om elever, anställda eller användare vet att de övervakas, måste det ske med transparens och goda skäl.

En grundregel: Logga för att skydda systemet – inte för att spionera på människor.

### **9.17.6** **Kontrollfrågor**

1. Varför är loggfiler viktiga vid IT-säkerhetsarbete?
2. Vad är skillnaden mellan loggning och övervakning?
3. Vad gör ett verktyg som Wazuh?
4. Vad står SIEM för och vad används det till?
5. Nämn två loggverktyg i Linux och två i Windows.
6. Vad innebär central logghantering?
7. Vad är fördelen med att ha övervakning i realtid?
8. Hur kan GDPR påverka hur man hanterar loggar?
9. Vad är syftet med korrelation i ett SIEM-system?
10. Varför är det viktigt att informera användare om övervakning?

### **9.17.7** **Fördjupningslänkar**

– Wazuh – Officiell webbplats  
 [https://wazuh.com](https://wazuh.com/)

– Graylog – Logghantering  
 [https://www.graylog.org](https://www.graylog.org/)

– Zabbix – Övervakning  
 [https://www.zabbix.com](https://www.zabbix.com/)

– Splunk – SIEM-lösning  
 [https://www.splunk.com](https://www.splunk.com/)

– MSB – Loggning och incidenthantering  
 <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/tekniska-atgarder/loggning-och-incidenthantering/>

### **9.17.8** **Egna anteckningar**

(Lämna plats för reflektioner, egna verktygstips eller exempel.)

## **9.18** **Säkerhet i skarpa miljöer och incidenthantering**

Att förstå teorin bakom IT-säkerhet är en sak – men i verkligheten är miljöer ofta röriga, användarna slarviga och hoten ständigt föränderliga. Det är i dessa **skarpa miljöer** som säkerhet sätts på prov. Här måste man inte bara förebygga attacker – utan också **upptäcka**, **begränsa skadan** och **återställa systemen** efter incidenter.

### **9.18.1 Vad menas med ”skarp miljö”?**

En ”skarp” miljö innebär att systemet används i drift, med riktiga användare, data och affärsprocesser. Det kan vara ett företag, en skola, ett sjukhus – vilket som helst system där driftstopp eller dataintrång får **verkliga konsekvenser**.

Exempel:

* En skola där elevregister läcks
* Ett företag som får sin webshop nedsläckt av ransomware
* En kommun som tappar tillgång till sina e-postservrar

Säkerhetskraven är ofta högre i skarpa miljöer än i utvecklings- eller testmiljöer. Här krävs:

* Backupstrategier
* Rutiner för uppdatering
* Övervakning
* Incidentplaner

### **9.18.2 Vad gör man när det händer något?**

Det spelar ingen roll hur bra skydd du har – **incidenter inträffar ändå**. Det kan vara:

* Skadlig kod
* Obehöriga inloggningar
* System som slutar svara
* Dataintrång

Då är frågan: Vad gör du?

De flesta företag (och skolor) bör ha en **incidenthanteringsplan**. Den består ofta av fem steg:

1. **Identifiera** – Vad har hänt?
2. **Meddela** – Informera rätt personer (IT, chef, juridik)
3. **Isolera** – Begränsa skadan (koppla bort nätverk, stäng ner system)
4. **Analysera** – Undersök hur det hände
5. **Återställ** – Få igång systemen igen, med bättre skydd

Tips: Ha kontaktvägar och ansvarsfördelning klara i förväg. I en kris är det inte läge att börja gissa vem som gör vad.

### **9.18.3 Att kommunicera i kris – vem säger vad till vem?**

När något går fel är **kommunikationen** ofta lika viktig som tekniken. Otydlig eller sen information kan leda till:

* Panik bland användare
* Dåligt rykte i media
* Bristande förtroende hos kunder, elever eller personal

En bra kommunikationsplan innehåller:

* **Vem** ansvarar för att informera?
* **Vilka** ska få information? (interna, externa)
* **Vad** ska sägas – och vad ska inte sägas?
* **Hur** ska det sägas? (e-post, möte, pressmeddelande?)

Exempel från skola:  
Om en elev har laddat ner ransomware till en dator i datasalen – vem informerar lärarna? Rektor? Eleverna? Vårdnadshavarna?

### **9.18.4 Dokumentation, bevis och återställning**

När incidenten hanterats är jobbet inte över. Nu börjar det viktiga **efterarbetet**:

* Dokumentera allt: Tider, IP-adresser, konton, åtgärder
* Spara loggar och filer (kan behövas som **bevis**)
* Analysera vad som gick fel – tekniskt och organisatoriskt
* Förbättra rutiner, uppdatera system och träna personal

I vissa fall kan man behöva kontakta:

* Datainspektionen (vid personuppgiftsläckor)
* Försäkringsbolag
* Polisen (vid brott)

Om du inte kan återställa systemen – var finns backupen? Och **hur snabbt** kan du återställa den?

### **9.18.5 Vanliga misstag vid incidenthantering**

1. **Ingen övning** – Många har en plan på papper, men ingen vet vad de ska göra i praktiken.
2. **Ingen loggning** – Utan loggar finns inget att analysera.
3. **För sent agerande** – Många attacker kan stoppas om man reagerar direkt.
4. **Kommunikationskaos** – "IT säger en sak, lärarna en annan – vem har rätt?"
5. **Ingen analys efteråt** – Problemen kommer tillbaka om man inte lär sig något.

Gör gärna en övning per år: "Vad gör vi om någon installerar ransomware?" – och låt olika roller spela igenom scenariot.

### **9.18.6** **Kontrollfrågor**

1. Vad innebär en "skarp miljö"?
2. Vad är de fem stegen i en typisk incidenthanteringsplan?
3. Varför är det viktigt att ha en kommunikationsplan vid incidenter?
4. Vad bör dokumenteras efter en säkerhetsincident?
5. Nämn två externa parter som kan behöva kontaktas vid ett intrång.
6. Varför är övning viktig för incidenthantering?
7. Vad är ett vanligt misstag vid incidenter?
8. Vad innebär det att isolera ett system?
9. Ge exempel på vad som kan ingå i en återställning.
10. Hur kan man använda en incident som lärotillfälle?

### **9.18.7** **Fördjupningslänkar**

– MSB – Hantera informationssäkerhetsincidenter  
 <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/organisatoriska-atgarder/hantera-incidenter/>

– CERT-SE – Nationell incidenthantering  
 <https://www.cert.se/>

– Datainspektionen – Anmäl personuppgiftsincident  
 <https://www.imy.se/privatperson/dataskydd/detta-galler/incidenter/>

– NIST – Computer Security Incident Handling Guide (PDF)  
 <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf>

– European Union Agency for Cybersecurity – Incident Response  
 <https://www.enisa.europa.eu/topics/csirt-cert-services>

### **9.18.8** **Egna anteckningar**

(Plats för reflektioner, rutiner ni har på skolan, eller saker du vill testa i praktiken.)

## **9****.19 Social engineering, phishing och mänskliga misstag**

De flesta säkerhetsproblem i IT-världen orsakas inte av supersmarta hackare som utnyttjar avancerade kodbuggar. De orsakas av… **människor**. Vi klickar på fel länk, luras att lämna ifrån oss information, eller glömmer uppdatera system i tid. Det är här **social engineering** kommer in.

Social engineering handlar om att utnyttja människors beteenden, vanor och tillit – inte att hacka datorer, utan att manipulera personer.

### **9.19.1 Vad är social engineering?**

Social engineering är en form av psykologisk manipulation. Syftet är att få någon att göra något de inte borde – t.ex. klicka på en länk, lämna ut ett lösenord eller installera skadlig kod.

Det bygger ofta på:

* Stress
* Auktoritet (”Jag ringer från IT”)
* Nyfikenhet (”Se bifogade dokument”)
* Rädsla (”Ditt konto stängs om du inte loggar in nu!”)

Exempel: Någon ringer till receptionen och låtsas vara tekniker. De ber om fjärråtkomst till en dator för att "installera en uppdatering". I själva verket tar de kontroll över systemet.

### **9.19.2 Vanliga sociala attacker**

Här är några vanliga metoder som används för att lura människor:

|  |  |  |
| --- | --- | --- |
| **Attacktyp** | **Beskrivning** | **Exempel** |
| Phishing | E-post som lurar dig att klicka på falsk länk | ”Du har ett paket att hämta – klicka här” |
| Spear phishing | Målriktad phishing, ofta riktad mot enskild person | E-post till rektorn från ”ekonomiavdelningen” |
| Vishing | Bedrägerisamtal via telefon | ”Banken ringer – vi behöver verifiera ditt konto” |
| Smishing | Phishing via SMS | ”Din kod är 4567 – klicka här för att logga in” |
| Pretexting | En bluffidentitet används för att bygga förtroende | En ”kollega” som ber dig dela lösenord |
| Tailgating | Någon följer efter dig in genom en säkerhetsdörr | ”Jag glömde mitt passerkort, kan du släppa in mig?” |

### **9.19.3 Varför fungerar det så ofta?**

Vi människor är ofta den svagaste länken i säkerhetskedjan – inte för att vi är dumma, utan för att vi:

* Vill hjälpa till
* Tänker inte alltid efter
* Har ont om tid
* Litar på andra
* Är ovana vid digitala hot

En välformulerad bluff kan lura både nybörjare och erfarna användare.

Faktum: Enligt flera studier är över 90 % av alla cybersäkerhetsincidenter relaterade till **mänskligt beteende**.

### **9.19.4 Träning och medvetenhet**

Att förhindra social engineering handlar mer om **kultur** än teknik.

Det viktigaste skyddet är att människor:

* Vet att hoten finns
* Lär sig känna igen varningssignaler
* Vågar säga nej
* Vet vem de ska kontakta vid tveksamheter

Bra metoder för skolor och arbetsplatser:

* Falska phishing-test (skicka egna bluffmejl som träning)
* Genomgångar av riktiga exempel
* Rollspel (”Hur reagerar du om någon ringer från IT?”)
* Synliga rutiner för att rapportera incidenter

### **9.19.5 Att skapa en stark säkerhetskultur**

Säkerhet börjar inte i brandväggen – den börjar med hur vi **tänker**.

Ett låst serverrum skyddar inte om någon släpper in en främling "för att de ser ut att jobba här".

En bra säkerhetskultur kännetecknas av:

* Låg tröskel för att rapportera misstänkta händelser
* Tydlighet: Vad gäller och vad gäller inte?
* Uppmuntran istället för skam (”Bra att du rapporterade”)
* Att säkerhet tas upp ofta – inte bara vid kris

### **9.19.6** **Kontrollfrågor**

1. Vad är social engineering?
2. Ge exempel på tre vanliga metoder inom social engineering.
3. Vad skiljer spear phishing från vanlig phishing?
4. Vad menas med vishing?
5. Varför fungerar social engineering ofta så bra?
6. Hur kan man träna människor att stå emot bluffar?
7. Vad är pretexting?
8. Vad menas med att ha en "säkerhetskultur"?
9. Varför är det viktigt att rapportera misstänkta händelser?
10. Ge ett exempel på en enkel men effektiv säkerhetsåtgärd.

### **9.19.7** **Fördjupningslänkar**

– CSO Online – What is Social Engineering?  
 <https://www.csoonline.com/article/2124681/what-is-social-engineering.html>

– KnowBe4 – Social Engineering Explained  
 <https://www.knowbe4.com/social-engineering>

– Phishing.org – What is Phishing?  
 <https://www.phishing.org/what-is-phishing>

– MSB – Phishing och bluffmejl  
 <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/organisatoriska-atgarder/bluffmejl/>

– Naked Security by Sophos – Real-life social engineering cases  
 <https://nakedsecurity.sophos.com/tag/social-engineering/>

### **9.19.8** **Egna anteckningar**

(Plats för egna reflektioner, exempel från skolan eller egna erfarenheter.)

## **9****.20 Loggning, övervakning och revision**

I takt med att hotbilden mot IT-system växer, blir det allt viktigare att inte bara skydda systemen – utan också att kunna **upptäcka, spåra och analysera incidenter när de inträffar**. Här spelar **loggning, övervakning och revision** en avgörande roll.

Att "ha loggning" är inte detsamma som att "använda loggarna". Många företag sparar automatiskt loggfiler men saknar rutiner för att läsa av dem, analysera dem eller agera på det som upptäcks. Detta kapitel fördjupar sig i **hur och varför loggning bör integreras i säkerhetsarbetet**.

### **9.20.1 Varför loggning är en säkerhetsfråga**

Loggning handlar om att skapa **spårbarhet**. När något går fel, eller när en attack upptäcks i efterhand, är loggfiler ofta det enda sättet att ta reda på **vad som hände, hur det gick till, och vem som var inblandad**.

Exempel:

* Inloggningsförsök (både lyckade och misslyckade)
* Filsystemsåtkomst
* Programinstallationer
* Systemuppdateringar
* Brandväggsregler som ändrats
* Trafik som blockeras eller släpps igenom

Loggar blir därmed en **"svart låda"** för IT-system, ungefär som i ett flygplan. Ju mer detaljerad loggning – desto bättre analysmöjlighet vid en incident.

### **9.20.2 Vad som bör loggas**

Att logga "allt" kan snabbt bli opraktiskt, dels p.g.a. lagringskostnader, men också för att det kan försvåra analysen. Fokus bör ligga på **viktiga händelser och mönster** som är relevanta för säkerhet:

* Inloggningar (särskilt felaktiga)
* Ändringar i användarbehörigheter
* Konfigurationsändringar
* Start/stopp av tjänster
* Brandväggsändringar
* Systemfel och kraschloggar
* Anrop till känsliga API:er eller databaser
* Åtkomst till känsliga filer

I domänmiljöer bör Active Directory-aktiviteter (Group Policy-ändringar, kontoskapande etc.) loggas noga.

För webbtjänster är det viktigt att logga:

* IP-adress
* Användaragent
* Sessions-ID
* Begärd URL
* Statuskod (200, 403, 404 osv.)

Men det är också viktigt att **inte logga för mycket persondata**, vilket vi kommer till nedan.

### **9.20.3 Övervakning i realtid**

Det räcker inte alltid att spara loggar – ibland krävs **aktiv övervakning i realtid** för att upptäcka hot innan de blir till attacker.

Verktyg som kan användas:

* **Wazuh** – kombinerar logginsamling, regler och incidentrespons
* **Fail2ban** – reagerar på mönster i loggar (t.ex. många felinloggningar)
* **SIEM-system** (Security Information and Event Management) – exempelvis Splunk, ELK eller Microsoft Sentinel
* **IDS/IPS-system** – ofta integrerade med loggning och övervakning

Övervakning i realtid är särskilt viktigt i miljöer med:

* Exponering mot internet
* Fjärrinloggning via SSH eller RDP
* Känslig data (t.ex. personuppgifter eller affärshemligheter)

### **9.20.4 Revision och efteranalys**

Revision är processen att **granska loggar i efterhand**, antingen i syfte att förstå en inträffad incident eller för att kontrollera efterlevnad av policyer och lagar.

Det kan handla om att:

* Kontrollera vem som ändrade i en databas
* Spåra när ett konto skapades och av vem
* Granska hur många gånger någon försökt logga in med fel lösenord
* Kontrollera vilka IP-adresser som haft åtkomst till ett system

Revision är också viktigt för att visa att man **följer lagar och regler**, t.ex. inom GDPR, ISO 27001 eller NIS2.

### **9.20.5 Juridiska aspekter på loggning**

Att logga innebär ofta att man behandlar **personuppgifter** – särskilt om IP-adresser, användarnamn eller aktivitet sparas. Därför är det viktigt att:

* Informera användare om loggning sker
* Begränsa loggtiden (t.ex. radera efter 30 dagar om inte längre krävs)
* Undvika att logga t.ex. lösenord, kreditkortsnummer eller andra känsliga uppgifter
* Skydda loggfiler mot obehörig åtkomst

En loggfil är alltså inte bara en teknisk resurs – den är också ett **register enligt dataskyddslagstiftningen**.

### **9.20.6 Vanliga misstag**

**Att inte kontrollera att loggning verkligen är aktiverad.**  
 Många system kräver att loggning manuellt slås på.

**Att spara loggar lokalt utan backup.**  
 Vid t.ex. ransomware-angrepp är lokala loggar ofta bland det första som raderas.

**Att inte läsa loggar förrän något går fel.**  
 Loggar ska analyseras **förebyggande**, inte bara i efterhand.

**Att inte tänka på integriteten.**  
 För mycket detaljer i loggar kan utgöra ett riskmoment i sig.

### **9.20.7** **Kontrollfrågor**

1. Varför är loggning en viktig del av IT-säkerhet?
2. Ge tre exempel på vad som bör loggas i ett serversystem.
3. Vad menas med "övervakning i realtid"?
4. Vilka verktyg kan användas för att upptäcka mönster i loggfiler?
5. Vad är ett SIEM-system?
6. Hur används revision i säkerhetsarbetet?
7. Vilken roll spelar loggning i efterlevnad av GDPR?
8. Vad är en risk med att logga för mycket information?
9. Ge exempel på en juridisk åtgärd man bör vidta vid loggning.
10. Varför är det ett problem att spara loggar endast lokalt?

### **9.20.8** **Fördjupningslänkar**

* Wazuh – Open Source Security Platform  
   <https://wazuh.com/>
* Logghantering och GDPR (Integritetsskyddsmyndigheten)  
   <https://www.imy.se/verksamhet/it-och-sakerhet/loggar-och-personuppgifter/>
* Introduction to SIEM – Microsoft  
   <https://learn.microsoft.com/en-us/azure/sentinel/overview>
* Graylog – Open source log management  
   <https://www.graylog.org/>
* EU GDPR – What is considered personal data?  
   <https://gdpr.eu/eu-gdpr-personal-data/>

### **9.20.9** **Egna anteckningar**

(Lämna plats för egna reflektioner och anteckningar här.)

## **9****.21 Backup och återställning**

Att ha en **bra backup** är inte bara en teknisk åtgärd – det är en **försäkring mot dataförlust, ransomware, hårddiskhaverier, mänskliga misstag och katastrofer**. Trots det är backup ofta något som prioriteras först *efter* att något gått fel. I detta kapitel går vi igenom varför backup är livsviktigt, vilka metoder som finns, hur man testar återställning och vanliga misstag att undvika.

### **9.21.1 Varför backup?**

Alla som någon gång har tappat bort viktiga filer vet hur frustrerande det kan vara. Men när det handlar om en hel servermiljö, kritisk verksamhetsdata eller en databas med kundinformation – då kan förlusten bli förödande.

Några vanliga orsaker till dataförlust:

* Ransomware (filer krypteras av skadlig kod)
* Oavsiktlig radering
* Haveri i hårdvara (t.ex. SSD-dödsfall)
* Brand, översvämning eller elfel
* Sabotage eller stöld
* Felaktiga uppdateringar

En säkerhetskopia är inte bara till för återställning efter katastrof – det är också ett **verktyg för att ångra misstag**, återställa tidigare versioner av filer, och möjliggöra experimentering utan rädsla.

### **9.21.2 Typer av backup**

Det finns flera typer av backup – varje med fördelar och nackdelar beroende på miljö och behov:

**Full backup**  
 En komplett kopia av allt som ska säkerhetskopieras. Långsammare, men enkel att återställa.

**Inkrementell backup**  
Säkerhetskopierar endast det som ändrats sedan **senaste backupen** (inkrementell eller full). Snabb och platsbesparande – men kräver fler steg vid återställning.

**Differentiell backup**  
Säkerhetskopierar allt som ändrats sedan **senaste fullständiga backup**. En kompromiss mellan hastighet och återställningskomplexitet.

**Snapshot**  
En ögonblicksbild av ett system – ofta använd inom virtualisering (t.ex. Proxmox eller Hyper-V). Bra för snabb återställning men bör inte ersätta traditionell backup.

**Molnbaserad backup**  
Data skickas till en extern molntjänst (exempel: Backblaze, Wasabi, Azure, AWS). Bra vid fysiska katastrofer men kräver stabil uppkoppling och dataskyddsavtal.

**Offline-backup**  
 Backup lagras på fristående media (ex. USB-diskar) som sedan kopplas bort från nätverket. Extra skydd mot ransomware.

### **9.21.3 3-2-1-regeln**

En tumregel för säker backup är:

**3 kopior** av data  
**2 olika typer av lagringsmedia**  
**1 kopia offsite** (t.ex. i moln eller annan byggnad)

Exempel:

* Original på server
* Kopia till lokal NAS
* Molnkopia till Backblaze

Denna metod skyddar mot både tekniska, fysiska och mänskliga fel.

### **9.21.4 Återställning – ofta glömd men avgörande**

Att kunna **återställa** är viktigare än att kunna ta backup. Det är lätt att tro att en backup fungerar – tills man behöver den. Därför bör **återställningstest göras regelbundet**.

Bra vanor:

* Dokumentera hur återställning går till (steg-för-steg)
* Testa återställning varje kvartal eller halvår
* Återställ både filer och hela system (inkl. databaser)
* Kontrollera integriteten hos backupfiler (t.ex. med checksummor)

Återställningen bör även testas i **tid** – om det tar 10 timmar att återställa ett system i drift, kan konsekvenserna bli dyra.

### **9.21.5 Kryptering och integritet**

Backup är också ett säkerhetsproblem. Om en backup innehåller känslig data (t.ex. personuppgifter, lösenord, medicinsk information) och hamnar i fel händer, kan det bli en katastrof.

Skyddsåtgärder:

* Kryptera backupfiler (t.ex. med VeraCrypt eller inbyggd lösning)
* Skydda backupservrar med brandväggar och stark autentisering
* Kontrollera loggar – så att ingen obehörig läser eller laddar ner backup

En komprometterad backup är värre än ingen backup alls – då ger den angriparen tillgång till hela ditt system.

### **9.21.6 Vanliga misstag**

**Bara en backup** – Om den skadas eller blir stulen finns inget skydd kvar.  
**Backup på samma disk/server** – Om den kraschar är allt borta.  
**Ingen testad återställning** – Det fungerar bara i teorin.  
**Okrypterade backupfiler** – Risk för dataläckor.  
**Manuella backuprutiner** – Människor glömmer. Automatisera!

### **9.21.7** **Kontrollfrågor**

1. Varför är backup viktigt för säkerhet?
2. Vad är skillnaden mellan en inkrementell och en differentiell backup?
3. Förklara 3-2-1-regeln för backup.
4. Varför räcker det inte att bara ta backup – varför måste man testa återställning?
5. Vad är en snapshot och när kan det användas?
6. Vilka risker finns med att inte kryptera sina säkerhetskopior?
7. Vad menas med offline-backup?
8. Hur kan molnbaserad backup vara både en fördel och en risk?
9. Vilken backupstrategi skulle du föreslå för ett mindre företag?
10. Vad är de vanligaste misstagen vid backuphantering?

### **9.21.8** **Fördjupningslänkar**

* Veeam – What is 3-2-1 Backup?  
   <https://www.veeam.com/blog/3-2-1-backup-rule.html>
* CrashPlan for Small Business  
   <https://www.crashplan.com/en-us/>
* Backblaze Blog – Real World Backup Strategies  
   <https://www.backblaze.com/blog/the-3-2-1-backup-strategy/>
* Linux Handbook – rsync command explained  
   <https://linuxhandbook.com/rsync-command/>
* MSB – Råd för backup och återställning  
   <https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/tekniska-atgarder/sakerhetskopiering/>

### **9.21.9** **Egna anteckningar**

(Lämna plats för egna reflektioner och anteckningar här.)

## **9****.22 Säkerhetskopiering i molnet**

I takt med att allt mer data lagras digitalt har behovet av säkerhetskopiering blivit viktigare än någonsin. Molnbackup har vuxit fram som ett kraftfullt alternativ till traditionell lokal backup. Det ger tillgång till backup oavsett plats – men medför också nya risker och krav. Det här kapitlet går igenom hur molnbackup fungerar, vad man bör tänka på och vanliga fallgropar.

### **9.22.1 Vad innebär molnbackup?**

Molnbackup innebär att man säkerhetskopierar sin data till en server på internet – ett så kallat moln. Det kan vara en dedikerad tjänst som Backblaze eller ett bredare ekosystem som Google Drive eller Microsoft OneDrive. Syftet är detsamma som vid lokal backup: att skydda data vid förlust, skada, stöld eller attack.

**Fördelar med molnbackup:**

* Åtkomst från olika platser och enheter
* Automatisk synkronisering och versionering
* Skydd mot lokala katastrofer (t.ex. brand, stöld)
* Skalbarhet – du kan växa med mängden data

**Nackdelar:**

* Kräver stabil internetanslutning
* Kostar pengar vid större volymer
* Integritetsrisk – data lagras hos tredje part
* Mindre kontroll än vid lokal backup

Att förstå var och hur datan lagras är avgörande – särskilt ur integritets- och GDPR-perspektiv.

### **9.22.2 Olika typer av molnbackup**

Alla molnbackuper fungerar inte likadant. Här är några vanliga modeller:

* **Molnet som primär backup**  
   Hela säkerhetskopieringen sker i molnet, t.ex. med tjänster som Backblaze, iCloud eller iDrive. Bra för hemanvändare eller småföretag.
* **Hybridbackup**  
   Data sparas både lokalt (t.ex. på extern hårddisk) och i molnet. Ger det bästa av två världar: snabb återställning lokalt + extern säkerhet.
* **SaaS-backup**  
   Backup av molnbaserade tjänster som Microsoft 365, Google Workspace eller Dropbox. Viktigt att förstå att Microsoft/Google inte automatiskt ansvarar för all din data – du är själv ansvarig för backup.
* **Backup i molninfrastruktur (IaaS/PaaS)**  
   Företag som kör servrar i Azure, AWS eller Google Cloud behöver ofta externa backup-lösningar. Inbyggda funktioner räcker sällan för full återställning.

### **9.22.3 Kryptering och integritet i molnbackup**

Säkerhetskopiering är inte säkert utan kryptering. Det finns två huvudmodeller:

* **Klientsidans kryptering**  
   Datan krypteras på användarens dator innan den skickas upp. Endast användaren har nyckeln – högsta säkerhetsnivå, men kräver att du själv hanterar nycklarna säkert.
* **Serversidans kryptering**  
   Datan skickas okrypterad och krypteras av molnleverantören. Enklare, men leverantören har tillgång till innehållet.

**Vem har nyckeln?**

* Om du själv har nyckeln (end-to-end encryption): bättre säkerhet.
* Om leverantören har nyckeln: större risk för intrång, dataläckor eller statlig insyn.

**Rättsliga krav att tänka på:**

* **GDPR**: Du ansvarar för hur persondata behandlas – även i molnet.
* **Schrems II**: Personuppgifter i moln utanför EU kräver särskilda skyddsåtgärder.
* **Dataplacering**: Kontrollera om datan lagras inom EU eller i tredje land.

### **9.22.4 Återställning från molnet – utmaningar och tips**

Att skapa en backup är inte svårt – det är återställningen som avslöjar om systemet fungerar. Flera utmaningar uppstår först när det verkligen gäller.

**Vanliga problem:**

* **Nedladdningshastighet:** Återställning av flera hundra GB kan ta timmar eller dagar.
* **Bandbreddsgränser:** Vissa molntjänster stryper trafiken vid hög användning.
* **Filversionsproblem:** Råkade du radera en fil för tre veckor sedan? Kanske finns den inte kvar längre.
* **Krypterade backuper:** Om du har tappat bort nyckeln är datan oåtkomlig – för alltid.

**Tips:**

* Testa återställning regelbundet (även om det bara är en liten mapp).
* Dokumentera exakt hur återställningen går till.
* Överväg leverantörer som erbjuder **"data shuttle"** – fysisk leverans av backup på hårddisk.

### **9.22.5 Exempel på populära molnbackup-tjänster**

|  |  |  |
| --- | --- | --- |
| **Tjänst** | **Fördelar** | **Nackdelar** |
| **Backblaze** | Obegränsad backup, enkel prissättning | Begränsad konfiguration |
| **iDrive** | Både moln och lokal backup | Kan vara lite rörigt gränssnitt |
| **CrashPlan** | Fokuserar på företag, bra versionshantering | Ej för personligt bruk längre |
| **Wasabi** | Billigt lagringspris, ingen egressavgift | Kräver manuell hantering |
| **Google Workspace Vault** | Integrerad med Google, bra sökfunktion | Inte backup i traditionell mening |

För privatpersoner kan även iCloud och OneDrive fungera som backup – men de har ofta begränsad kontroll och automatisering jämfört med dedikerade tjänster.

### **9.22.6 Fallgropar och rekommendationer**

**Molnet är inte magi.** Många tror att bara för att data ligger i molnet så är den säker – men det stämmer inte. Du måste själv:

* Säkerställa att backup faktiskt sker (och inte bara en synkning!)
* Verifiera versionshantering – hur länge sparas gamla versioner?
* Se till att rätt mappar/filer inkluderas (ibland måste de väljas manuellt)
* Testa återställning regelbundet

**Vanliga misstag:**

* Synkronisering förväxlas med backup – om du raderar en fil i Dropbox så raderas den överallt
* Retention är för kort – gamla raderade filer går inte att återställa
* Kryptering glöms bort – data lagras i klartext hos leverantören

**Rekommendation:** Använd en hybridlösning (lokalt + moln), välj en tjänst med stark kryptering och dokumentera både backup- och återställningsprocess.

### **9.22.7 Kontrollfrågor**

1. Vad är skillnaden mellan molnbackup och lokal backup?
2. Nämn två fördelar och två nackdelar med molnbackup.
3. Vad menas med SaaS-backup och varför behövs det?
4. Vad är skillnaden mellan klientsidans och serversidans kryptering?
5. Vilka rättsliga krav måste beaktas vid lagring av data utanför EU?
6. Varför är det viktigt att testa återställning regelbundet?
7. Vad menas med hybridbackup?
8. Nämn tre exempel på molnbackup-tjänster.
9. Vad är problemet med att bara använda synkronisering som backup?
10. Vad innebär begreppet ”retention” i samband med backup?

### **9.22.8 Fördjupningslänkar**

* **Backblaze – How Backup Works**  
   <https://www.backblaze.com/backup-software.html>
* **Wasabi – Cloud Storage Overview**  
   <https://wasabi.com/cloud-storage/>
* **GDPR och molntjänster (IMY)**  
   <https://www.imy.se/verksamhet/tillsyn-av-molntjanster/>
* **CrashPlan for Business**  
   <https://www.crashplan.com/en-us/business/>
* **Veeam – SaaS Backup for Microsoft 365**  
   <https://www.veeam.com/backup-microsoft-office-365.html>

### **9.22.9 Egna anteckningar**

(Lämna plats för reflektioner, exempel från undervisningen eller egna molnerfarenheter.)

## 9.23 Etisk hacking och hackingkultur

I dagens digitala värld är det viktigt att förstå både hur attacker går till och varför de sker. Men det är minst lika viktigt att förstå skillnaden mellan att utföra en attack – och att förstå den för att kunna försvara sig. Det är här begreppet *etisk hacking* kommer in i bilden.

En etisk hackare (även kallad *white hat*) använder sina kunskaper om cybersäkerhet för att upptäcka sårbarheter innan någon illasinnad hinner utnyttja dem. Detta sker med tillstånd och i syfte att skydda. Etisk hacking är därför inte bara acceptabel – den är en central del av dagens IT-säkerhetsarbete.

Men för att förstå försvar måste man förstå angrepp. I detta kapitel går vi därför igenom vanliga angreppstyper, hotaktörer, verktyg, och hur den etiska hackaren arbetar.

### **Hattfärger – olika typer av hackers**

Inom säkerhetsvärlden används färger för att beskriva olika roller och avsikter:

* **Vit hatt** – Etiska hackers. Arbetar för att hitta och rapportera säkerhetshål innan någon annan utnyttjar dem. Ofta anställda inom IT-säkerhet.
* **Svart hatt** – Illasinnade hackers. Bryter sig in i system för personlig vinning eller förstörelse.
* **Grå hatt** – Ligger mittemellan. Upptäcker sårbarheter utan tillstånd, men rapporterar dem ofta – ibland efter att ha demonstrerat sin poäng.
* **Grön hatt** – Nybörjare i hackingvärlden. Ofta nyfikna, med ambitionen att lära sig, men utan stor erfarenhet.
* **Röd hatt** – En ovanligare beteckning. Används ibland för att beskriva aggressiva "hack-back"-aktörer som inte bara försvarar sig utan går till motattack. Också ett litet skämt, eftersom Red Hat är ett Linux-företag.

Att förstå hacking handlar alltså inte bara om teknik, utan även om avsikt, etik och lagstiftning.

### **Operativsystem för penetrationstester**

Professionella penetrationstestare använder särskilda operativsystem som är fullpackade med verktyg för att testa säkerheten i system:

* **Kali Linux** – Det mest kända OS:et för penetrationstester. Innehåller hundratals förinstallerade verktyg för nätverksscanning, analys, brute force, social engineering och mer.
* **Parrot Security OS** – Ett alternativ till Kali som fokuserar mer på anonymitet, forensik och prestanda. Också Debian-baserat.
* **BackBox** – Ett lättviktsalternativ till Kali, baserat på Ubuntu.
* **Tails** – Inte för hacking direkt, men används av aktivister och journalister för att surfa anonymt och säkert via Tor.

Dessa system används i utbildningssyfte, på testmiljöer – aldrig mot verkliga system utan tillstånd.

### **Verktyg för etisk hacking – exempel och funktion**

* **Wireshark** – Nätverkssniffer för att analysera trafik. Vanligt för att avslöja klartextlösenord och förstå nätverksprotokoll.
* **Nmap** – Skannar nätverk efter öppna portar och tjänster.
* **Metasploit Framework** – Ramverk för att utveckla och köra exploits mot sårbara system.
* **Hydra** – Brute force-verktyg som testar lösenord mot olika tjänster.
* **John the Ripper / Hashcat** – Verktyg för att knäcka hashade lösenord.
* **Burp Suite** – Verktyg för att testa webbapplikationers säkerhet (XSS, SQLi m.m.).

### **Syftet med etisk hacking**

Målet med etisk hacking är inte att förstöra – utan att förbättra. Genom att tänka som en angripare kan man identifiera svagheter i system innan de utnyttjas av någon med onda avsikter. Etisk hacking används bland annat för:

* Penetrationstester
* Sårbarhetsbedömningar
* Utbildning och träning
* Säkerhetsgranskningar
* Bug bounty-program

Men det viktigaste verktyget är alltid **etik och ansvar**.

### **Vanliga attacker**

* **Man-in-the-middle (MITM)** – Avlyssning och manipulering av trafik mellan två parter.
* **Man-on-the-side** – Passiv avlyssning och snabba svar innan legitima servrar hinner svara.
* **Maid attack** – Fysisk tillgång till datorer utnyttjas (t.ex. hotellrum).
* **Drive-by attack** – Webbsidor infekterar besökare utan att de klickar.
* **Cyber Kill Chain** – Struktur för att förstå steg i en attack: Recon → Weaponize → Deliver → Exploit → Install → C2 → Act.
* **Spoofing** – Falsk identitet (e-post, IP, DNS etc).
* **Squatting** – Domännamn som liknar kända sajter, ofta för phishing.
* **Zero-day exploits** – Okända sårbarheter som utnyttjas innan de patchas.
* **Privilege escalation** – Att få högre rättigheter än man ska ha.
* **Pivoting** – Att använda en komprometterad maskin för att ta sig vidare i ett nätverk.
* **Backdoors** – Dolda åtkomstvägar skapade av utvecklare eller angripare.

### **Sätt att få tag på information**

Utöver social engineering och tekniska attacker finns olika sätt att få tag på lösenord och autentiseringsuppgifter:

* **Brute force** – Testar alla möjliga kombinationer.
* **Dictionary attack** – Testar vanliga lösenord från en ordlista.
* **Rainbow tables** – Förgenererade hash-tabeller för att knäcka lösenord snabbare.
* **Credential stuffing** – Testar läckta lösenord från andra sajter.
* **Keylogging** – Loggar tangenttryckningar.
* **Sniffing** – Avlyssnar oskyddad trafik.
* **MITM / Man-on-the-side** – Kan användas för att få tag på inloggningar.
* **Shoulder surfing** – Tittar över någons axel.

Alla dessa metoder är olika varianter av informationsinsamling – och måste förstås för att byggas skydd mot dem.

### **9.23.1 Kontrollfrågor**

1. Vad är skillnaden mellan en vit hatt och en svart hatt?
2. Vad används Kali Linux till?
3. Hur fungerar ett brute force-angrepp?
4. Vad är en drive-by-attack?
5. Hur fungerar pivoting?
6. Vad gör Metasploit?
7. Vad är en zero-day-sårbarhet?
8. Vad innebär spoofing?
9. Ge exempel på hur en hacker kan använda ett keylogger.
10. Vad är syftet med bug bounty-program?

### **9.23.2 Fördjupningslänkar**

* OWASP – Cyber Kill Chain Explained  
  <https://owasp.org/www-community/attack-methods/Cyber_Kill_Chain>
* Cloudflare – What is a Man-in-the-Middle Attack?  
  <https://www.cloudflare.com/learning/ddos/glossary/man-in-the-middle-attack/>
* HackerOne – Bug Bounty Programs  
  <https://www.hackerone.com/bug-bounty-programs>
* Comparitech – What is Typosquatting?  
  <https://www.comparitech.com/blog/information-security/typosquatting/>
* IBM – What is privilege escalation?  
  <https://www.ibm.com/topics/privilege-escalation>

### **9.23.3 Egna anteckningar**

(Lämna plats för egna reflektioner och fördjupningar här.)

## 9.24 Grundläggande säkerhetsprinciper

Att förstå IT-säkerhet handlar inte bara om att förstå hur attacker går till – utan också hur man skapar en kultur och struktur för försvar. Det finns ett antal grundläggande principer som ofta återkommer i all säkerhetsdesign, oavsett om det handlar om att säkra en webbserver, ett Wi-Fi-nätverk eller en hel myndighets IT-system.

### **9.24.1 Least Privilege (minsta privilegium)**

Principen om minsta privilegium handlar om att varje användare, program eller process endast ska ha de rättigheter som absolut behövs för att utföra sin uppgift. Inget mer.

**Exempel:**

* En IT-tekniker behöver kanske tillfälligt administratörsåtkomst för att installera en drivrutin – men bör sedan arbeta med ett vanligt användarkonto.
* Ett backup-program bör endast ha läsrättigheter till källfiler – inte rätt att ändra eller ta bort dem.

Syftet är att minimera skada vid intrång. Om en angripare lyckas ta över ett konto eller ett program, ska skadan begränsas av det faktum att kontot/programmet inte hade full behörighet.

### **9.24.2 Defense in Depth (försvar i flera lager)**

Defense in depth handlar om att aldrig förlita sig på ett enda skydd. Istället skapar man flera lager av försvar som samverkar.

Tänk dig en borg: den har murar, vallgravar, vakter, fällor och utrymningsvägar. Samma tänk gäller i IT-säkerhet:

* Krypterade hårddiskar
* Brandväggar
* Antivirus
* Intrusion Detection Systems (IDS)
* Tvåfaktorsautentisering
* Backup

Om ett lager brister ska det finnas fler som tar vid. Detta tänkande är centralt i alla professionella säkerhetsmiljöer.

### **9.24.3 Security by Design**

Security by Design betyder att säkerhet inte ska läggas till i efterhand – den ska vara inbyggd från början. Programvara, system och rutiner ska designas med säkerhet som en naturlig del av arkitekturen.

**Exempel:**

* En webbapplikation ska från start validera all input för att undvika t.ex. SQL-injection.
* Ett operativsystem ska vara säkert i standardkonfigurationen, inte först efter timmars justering.

Denna princip har blivit allt viktigare i takt med DevSecOps-modellen, där utveckling, säkerhet och drift samverkar.

### **9.24.4 Fail Secure / Fail Safe**

Vad händer om ett system kraschar? Ska det bli helt öppet – eller helt stängt?

* **Fail Secure:** Systemet går i lås. Ingen kommer in. Prioriterar skydd före tillgänglighet.
* **Fail Safe:** Systemet tillåter viss tillgång. Prioriterar tillgänglighet före skydd.

Båda har sin plats. Ett brandskyddssystem ska vara "fail safe" (alla dörrar öppnas vid brand). Ett bankvalv ska vara "fail secure".

IT-säkerhet behöver båda perspektiven beroende på systemets syfte.

### **9.24.5 Separation of Duties**

Denna princip handlar om att dela upp ansvar mellan olika personer eller roller för att minska risken för missbruk eller fel.

**Exempel:**

* En person får konfigurera backup.
* En annan får återställa data.

Ingen ska kunna styra hela kedjan själv, för då kan den personen exempelvis ta bort loggar, dölja spår eller förstöra bevis.

Separation of Duties är vanligt inom ekonomi, revision och nu även inom IT.

### **9.24.6 Kontrollfrågor**

1. Vad innebär principen om minsta privilegium?
2. Ge ett exempel på Defense in Depth i praktiken.
3. Vad menas med Security by Design?
4. Vad är skillnaden mellan "fail secure" och "fail safe"?
5. Varför är Separation of Duties viktig inom IT-säkerhet?
6. Hur kan principen om minsta privilegium förhindra större skador vid ett intrång?
7. Hur fungerar Defense in Depth i en vanlig hemrouter?
8. Vad är DevSecOps?
9. Vilka risker finns med att inte bygga in säkerhet från start?
10. Vad kan hända om en person har ensam kontroll över alla säkerhetsfunktioner?

### **9.24.7 Fördjupningslänkar**

* National Cybersecurity Alliance – Least Privilege Explained <https://staysafeonline.org/resources/what-is-the-principle-of-least-privilege/>
* OWASP – Defense in Depth <https://owasp.org/www-community/Defense_in_depth>
* IBM – Security by Design <https://www.ibm.com/topics/security-by-design>
* Microsoft Learn – Fail Safe vs Fail Secure <https://learn.microsoft.com/en-us/azure/architecture/framework/security/design-principles>
* SANS Institute – Separation of Duties <https://www.sans.org/blog/separation-of-duties/>

### **9.24.8 Egna anteckningar**

(Lämna utrymme för reflektion, minnesanteckningar eller egna exempel.)

## 9.25 IT-säkerhet på arbetsplatsen – framtid och ansvar

IT-säkerhet är inte längre en fråga för bara IT-avdelningen. I dagens digitala samhälle är varje individ – oavsett yrkesroll – en del av försvarslinjen. Från att klicka på länkar i e-post till att hantera känsliga kunduppgifter på ett korrekt sätt: var och en har ett ansvar.

I det här sista kapitlet ska vi sätta all kunskap i kontext. Hur omsätter man säkerhetsprinciper i vardagen? Vad betyder det att vara en ansvarsfull användare – och en framtida IT-tekniker? Och vad väntar runt hörnet när det gäller hot och teknik?

### **9.25.1 Praktisk tillämpning av säkerhet**

Teori i all ära – men vad innebär det i praktiken?

**Lösenordshantering**  
 Använd starka, unika lösenord – helst med lösenordshanterare. Dela aldrig lösenord, inte ens med kollegor.  
Många intrång sker fortfarande på grund av svaga eller återanvända lösenord.

**E-postetikett och phishingmedvetenhet**  
 Granska avsändare, håll muspekaren över länkar innan du klickar, rapportera misstänkt e-post.

**Inloggning och sessioner**  
 Lås datorn när du lämnar arbetsplatsen – även om det bara är för en fika. Logga ut från fjärrsessioner och stäng ned system du inte använder.

**Uppdateringar och patchar**  
 Säkerhetshål lagas hela tiden – men fungerar bara om uppdateringar faktiskt installeras. Detta gäller allt från operativsystem till appar och tillägg i webbläsaren.

### **9.25.2 Det personliga ansvaret**

I många fall är det mänskliga misstag – inte tekniska brister – som leder till dataintrång. Därför måste **säkerhetskultur** vara levande.

Alla som arbetar med IT eller digitala system har ett ansvar:

* Att följa policys och rutiner
* Att rapportera avvikelser och misstänkta händelser
* Att inte klicka “bara för att det går”
* Att säga ifrån vid dålig praxis

Ett exempel: Om du ser att en kollega använder en post-it med lösenord – säg till! Det är inte att vara besvärlig – det är att vara proffsig.

### **9.25.3 Vanliga misstag – och hur man undviker dem**

Några klassiker som fortfarande dyker upp:

* **Samma lösenord överallt**
* **Dela konto för att det är "lättare"**
* **Inaktiva konton som aldrig tas bort**
* **USB-stickor som tappas bort utan kryptering**
* **Slumpmässig användning av molntjänster utan IT-avdelningens vetskap (s.k. Shadow IT)**

Tips: Se alltid till att det finns en balans mellan **användarvänlighet och säkerhet** – inte ett motsatsförhållande.

### **9.25.4 Etik, lagstiftning och professionellt ansvar**

IT-säkerhet är inte bara teknik – det är också lag och moral. I arbetslivet gäller:

* **GDPR** – du får inte behandla personuppgifter hur som helst.
* **Säkerhetslagen** – vissa verksamheter är samhällsviktiga och omfattas av särskilda krav.
* **Offentlighetsprincipen** – i myndigheter är mycket dokumentation tillgänglig för allmänheten.
* **Sekretessavtal** – många arbetsplatser kräver att du skriver under sekretessklausuler.

Men även där lagen är tyst – finns **etiken**. Vad är rimligt? Vad är rätt?  
En duktig tekniker ställer frågan: *"Bara för att jag kan – betyder det att jag bör?"*

### **9.25.5 Framtidens säkerhet – nya hot och möjligheter**

IT-säkerhet står aldrig stilla. Här är några områden att hålla ögonen på:

**AI (Artificiell Intelligens)**  
AI används både för att upptäcka attacker snabbare – men också av angripare för att skapa mer avancerade phishingmejl eller bryta mönster.

**IoT (Internet of Things)**  
Från kaffemaskiner till industrimaskiner – alla är uppkopplade, men få är säkrade.

**Kvantdatorer**  
Inte ett hot i dag – men kan i framtiden knäcka dagens krypteringsalgoritmer. Post-quantum-krypto är redan under utveckling.

**Smarta städer**  
 När trafikljus, elnät och vattenförsörjning blir digitala – måste säkerheten följa med.

### **9.25.6 Riktiga incidenter – vad kan vi lära oss?**

Här är tre verkliga händelser att reflektera kring:

1. **Maersk och NotPetya (2017)**  
    Ett ransomwareangrepp lamslog logistikjätten Maersk. Orsak: En ouppdaterad programvara i ett kontor i Ukraina. Totalkostnad: Uppemot 10 miljarder kronor.
2. **Sony Pictures (2014)**  
    Angrepp från en nationell aktör (Nordkorea) efter en film. Stora mängder interna dokument läckte. Lärdom: Säkerhet är också geopolitik.
3. **MyFitnessPal (Under Armour, 2018)**  
    150 miljoner konton läckte. Användarnas e-postadresser, hashade lösenord och användarnamn kom ut. Lärdom: Även "oskyldiga" appar bär ansvar för dataskydd.

### **9.25.7 Sammanfattning – säkerhet är allas ansvar**

När eleverna lämnar skolbänken för att börja arbeta kommer de att ha en nyckelroll: som användare, tekniker, utvecklare – och försvarare av information.

Det räcker inte med brandväggar och antivirus. Det krävs människor som förstår, ifrågasätter och agerar.

Så fråga dig själv:  
*Vad tar du med dig från det här kapitlet? Från hela kursen?*

Ditt ansvar börjar här.

### **9.25.8 Kontrollfrågor**

1. Vad menas med “säkerhetskultur” på en arbetsplats?
2. Vad är ett exempel på Shadow IT och varför är det ett problem?
3. Vilka lagar påverkar dig som IT-ansvarig i en organisation?
4. Vad är risken med att använda samma lösenord överallt?
5. Vad är skillnaden mellan GDPR och Säkerhetslagen?
6. Hur kan AI användas både för och emot IT-säkerhet?
7. Vad innebär begreppet “etiskt ansvar” i IT-sammanhang?
8. Vad lärde vi oss av Maersk-attacken?
9. Hur påverkar IoT vår syn på säkerhet?
10. Varför är det viktigt att även “småsaker” som USB-stickor hanteras säkert?

### **9.25.9 Fördjupningslänkar**

MSB – Informationssäkerhet i praktiken  
 <https://www.informationssakerhet.se/>

European Union Agency for Cybersecurity (ENISA)  
 <https://www.enisa.europa.eu/>

Cybersecurity & Infrastructure Security Agency (CISA) – Tips för arbetsplatser  
 <https://www.cisa.gov/secure-our-world>

CNIL (Frankrike) – GDPR Explained  
 <https://www.cnil.fr/en/home>

OWASP – Future Threats  
 <https://owasp.org/www-project-top-ten/>

### **9.25.10 Egna anteckningar**

(Lämna plats för reflektion, tankar och exempel från egna erfarenheter.)