

# DATORTEKNIK



## 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
1.1 Moderkortet.....	7
1.2 Processorn (CPU).....	13
1.3 RAM (Arbetsminne).....	22
1.4 Långtidslagring.....	29
1.5 PSU (Nättaggregat).....	37
1.6 GPU (Grafikkort).....	42
1.7 Portar och bussar.....	49
2 Uppstart.....	59
2.1 Introduktion.....	59
2.2 Översikt av uppstartssekvensen.....	59
2.3 Strömförsörjning och CPU-start.....	60
2.4 BIOS.....	61
2.5 Det tekniska bakom BIOS.....	61
2.6 UEFI.....	62
2.7 Secure Boot.....	63
2.8 Bootloaders.....	64
2.8.4 Uppstartssekvenser - exempel.....	65
2.8.5 Vanliga problem.....	65
2.8.6 Fördjupning.....	66
3 Virtualisering.....	70
3.1 Vad innebär virtualisering.....	70
3.2 Varför använder man virtualisering.....	70
3.3 Bare-metal hypervisorer.....	70
3.4 Hosted hypervisorer.....	72

3.5 Koppling mot molnet.....	73
3.6 Containrar (kort introduktion).....	74
3.7 Fördjupande länkar.....	75
3.8 Kontrollfrågor.....	76
3.9 Egna anteckningar.....	77
4 Lagring.....	78
4.1 Vad är lagring?.....	78
4.2 Disk – den fysiska enheten.....	78
4.3 Partition och partitionstabell.....	78
4.4 Volym och filsystem.....	79
4.5 Enhet i Windows – enhetsbokstav.....	80
4.6 Sammanfattning.....	81
4.7 Fördjupningslänkar.....	82
4.8 Diskussions- och kontrollfrågor.....	83
4.9 Egna anteckningar.....	84
4.10 RAID och LVM – Redundans och flexibel lagring.....	85
5 Windows.....	94
5.1 Historia.....	94
5.2 Uppbyggnad.....	102
5.3 Fördelar och nackdelar med Windows som operativsystem.....	107
5.4 Installation av Windows 11 i fysisk maskin/VM.....	114
5.5 Konfiguration och hantering av Windows.....	120
6 Windows Server.....	130
6.1 Installation av Windows Server.....	130
6.2 Workgroup, Domain och Active Directory.....	132
6.2.1 Vad är en Workgroup?.....	132

7. Linux – Historia, filosofi och ekosystem.....	138
7.1 Vad är Linux?.....	138
7.1.2 Linuxkärnans grenar (distributioner).....	140
7.2 Installation av Ubuntu Desktop.....	147
7.4 Installation av Ubuntu Server och Webmin.....	165
7.5 Linux grundläggande kommandon.....	172
8 OSI-modellen.....	193
8.1 Uppkomst – Bakgrunden till OSI.....	193
8.2 Syfte – Varför finns OSI-modellen?.....	194
8.3 Användningsområde – Hur används OSI-modellen?.....	194
8.4 Diskussions- och kontrollfrågor.....	197
8.5 Fördjupningslänkar.....	198
8.6 Egna anteckningar.....	199
9 Introduktion till IT-säkerhet.....	200
9.1 Vad är IT-säkerhet?.....	200
9.2 Skydd på olika nivåer.....	207
9.3 Trådlös säkerhet.....	215
9.4 Hot, risker och sårbarheter.....	222
9.5 Säkerhet i Windowsmiljö.....	227
9.6 Säkerhet i Linuxmiljö.....	246
9.7 Linux OS-härdning (Operating System Hardening).....	252
9.8 Linux paketuppdatering och patchhantering.....	258
9.9 Linux systemövervakning och loggning (journald, rsyslog, AIDE). ....	264
9.10 Linux root, sudo och användarbehörighet.....	270
9.11 Linux loggning och övervakning.....	276
9.12 Linux härdning av tjänster (t.ex. SSH, brandvägg, TLS/SSL).....	282

9.13 Linux backup och återställning.....	288
9.14 Linux automatisering av säkerhetsuppgifter (cron, script, loggning) .....	294
9.15 Linux Säkerhetsverktyg.....	300
9.16 Jämförelse: Säkerhet i Windows och Linux.....	306
9.17 IT-säkerhet ur ett verksamhetsperspektiv.....	314
9.18 Moderna hot och skydd.....	340
9.19 Verktyg, övervakning och loggning.....	347
9.20 Säkerhet i skarpa miljöer och incidenthantering.....	353
9.21 Social engineering, phishing och mänskliga misstag.....	361
9.22 Loggning, övervakning och revision.....	366
9.23 Backup och återställning.....	371
9.24 Säkerhetskopiering i molnet.....	377
9.25 Etisk hacking och hackingkultur.....	383
9.26 Grundläggande säkerhetsprinciper.....	391
9.27 IT-säkerhet på arbetsplatsen – framtid och ansvar.....	397

# 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), lagringenheter (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 kommuniceras. 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, resetknapp 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.



Standard-ATX



Micro-ATX



Mini-ITX



Pico-ITX  
Nano-ITX

## 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

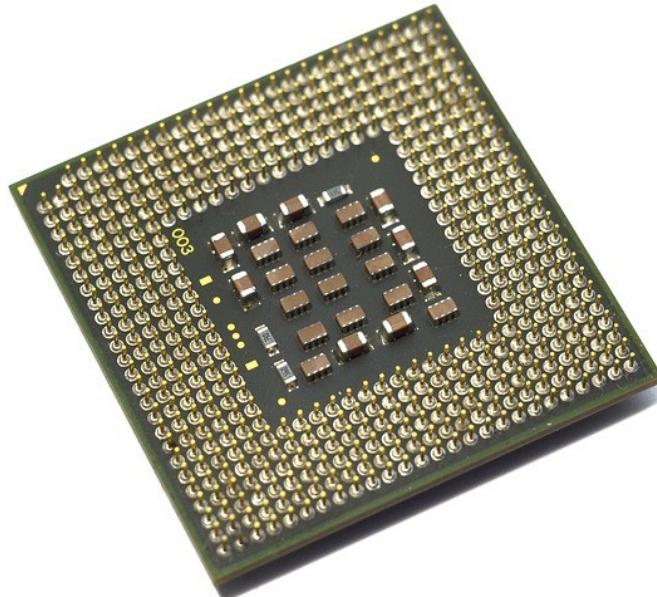
*(Här kan du som elev skriva egna anteckningar om moderkort, frågor eller reflektioner)*

## 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 attporta 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 åren 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 häpnäst (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 synkronisera 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 klockfrekvensen 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 ledar 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 kylnings

En CPU installeras i moderkortets sockel. Det är avgörande att välja en processor som passar moderkortets sockettyp. 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 kylnings 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](https://www.youtube.com/watch?v=cNN_tTXABUA)
- Wikipedia – Central Processing Unit  
[https://en.wikipedia.org/wiki/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 längsammare 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 korrigeras fel – lite dyrare och långsammare, men säkrare.

### 1.3.7 Skillnaden mellan RAM och lagring

RAM är **volatilt minne** – innehållet försätts i 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](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 försinner 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 sparats 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](https://en.wikipedia.org/wiki/Hard_disk_drive)
- Wikipedia – Solid-state drive  
[https://en.wikipedia.org/wiki/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 värmeförlust.

#### 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 entusiastbygggen eller servrar.
- **Titanium**: 94-96 % - toppklass, dyrt men extremt effektivt.

En bättre certifiering innebär mindre elförbrukning och lägre värmeförlust.

### 1.5.4 Modulärt eller icke-modulärt

#### **Icke-modulärt PSU:**

- Alla kablar sitter fast.
- Billigare, men kan bli stökgilt 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\\_\(computer\)](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 parallel 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 grafikprestandan kunde man tidigare koppla ihop flera grafikkort i samma dator. Detta gjordes med tekniker som SLI (Scalable Link Interface) från Nvidia och CrossFire från AMD. Lösningen gick ut på att korten delade på renderingsarbetet – till exempel genom att växelvis rita varannan bildruta. Det krävde dock att spelens varianter anpassade för tekniken, vilket inte alltid var fallet. Även om detta kunde ge prestandafördelar i vissa fall, fanns flera begränsningar: högt strömförbrukning, varierande resultat i olika spel, och ett beroende av fungerande drivrutiner. Med tiden har stödet för dessa lösningar försunnit helt, och idag satsar man i stället på kraftfulla enskilda grafikkort.

## 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 upscaling-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)

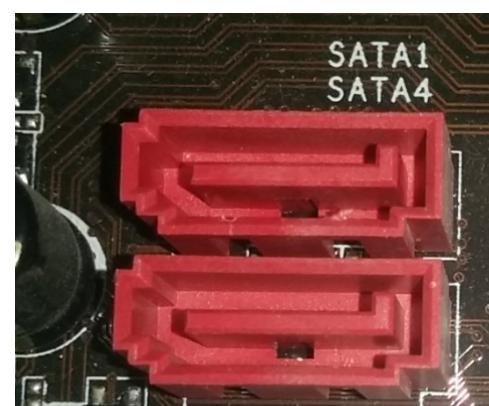
Standardanslutningen för moderna expansionskort.

- 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 försvisser 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 nätaggregater.

## 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.
  - o USB 2.0: ~5 m max för full hastighet.
  - o 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](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

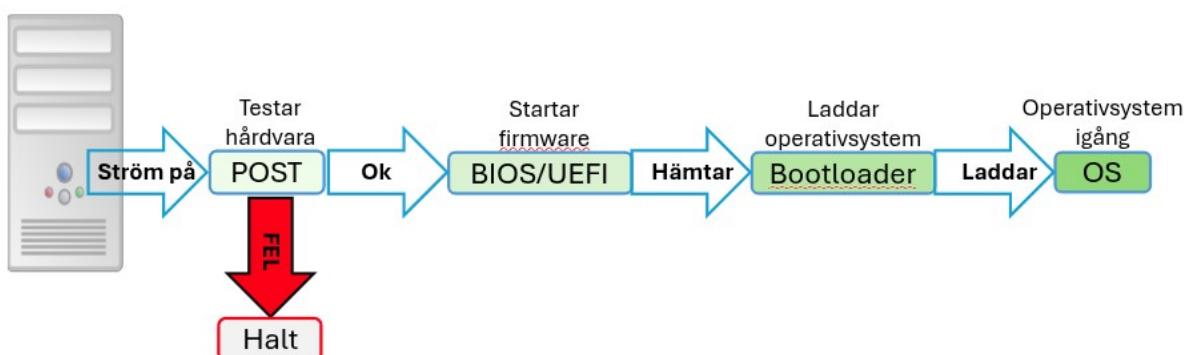
När vi slår på en dator startar en noga planerad kedja av steg som till slut laddar operativsystemet i minnet. Den här processen kallas **uppstарт** eller **boot** och är avgörande för att en dator överhuvudtaget ska fungera.

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):** 0xFFFF0

Adderat som fysisk adress:

CS base: 0xFFFF0000

IP: 0x0000FFFF0

-----  
= 0xFFFFFFFF0

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 överläta 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 – Basic Input/Output System – är den grundläggande programvara som körs direkt när datorn startas. Traditionellt är BIOS lagrat i antingen ROM (Read Only Memory) eller Flashminne på moderkortet, vilket gör det möjligt att köras oavsett om hårddisken fungerar eller inte. BIOS innehåller så kallad bootstrapkod som alltid är tillgänglig, och som har till

uppgift att starta upp datorn och initiera de komponenter som behövs för att ladda operativsystemet.

En viktig funktion i BIOS är att hantera bootordningen – alltså vilken enhet som datorn försöker starta från först. Det kan till exempel vara en intern hårddisk (HDD eller SSD), ett USB-minne, en CD/DVD eller nätverksuppsättning (PXE-boot). Denna bootordning kan ändras av användaren via BIOS-inställningarna.

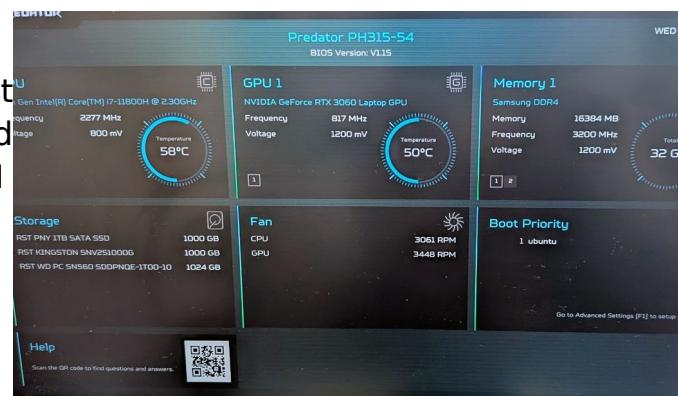
Traditionellt har BIOS använt partitionstabellen MBR (Master Boot Record) för att hantera diskar. MBR har dock vissa begränsningar – framför allt klarar den bara diskar upp till 2 terabyte i storlek. Själva bootinformationen i MBR ligger i de första 512 byten (bytes) av hårddisken, där även en liten del av bootloadern sparas. Denna kod ansvarar för att starta operativsystemets mer avancerade startprocedur.

## 2.6 UEFI

### 2.6.1 Vad är UEFI?

UEFI står för Unified Extensible Firmware Interface och är en modern ersättare till det traditionella BIOS.

Precis som BIOS är UEFI lagrat i ett flashminne på moderkortet, men det erbjuder en betydligt mer avancerad miljö. Till skillnad från BIOS har UEFI ett grafiskt gränssnitt med stöd för mus, vilket gör det mer användarvänligt. Det är också tekniskt mer kapabelt – till exempel kan det hantera stora hårddiskar över 2 terabyte genom att använda GPT (GUID Partition Table) i stället för MBR. UEFI har dessutom inbyggt nätverksstöd direkt i firmware och möjligheten att ladda och köra drivrutiner redan innan operativsystemet startas.



## 2.6.2 Historia och utveckling

UEFI är en vidareutveckling av Intel EFI (Extensible Firmware Interface) som togs fram under 2000-talets första decennium. I början av 2010-talet blev UEFI en industristandard och började ersätta BIOS i allt fler datorer. Syftet med UEFI var att lösa många av de begränsningar som BIOS led av. Med UEFI blev det möjligt att använda större diskar tack vare GPT, moderna drivrutiner kunde laddas in tidigare i uppstartsprocessen, och firmwaredesignen blev modulär och mer flexibel. Dessutom bidrog UEFI till snabbare uppstartstider, bland annat genom förbättrad POST (Power-On Self Test) och optimerad kod.

## 2.6.3 Varför är UEFI bättre?

Det finns flera skäl till varför UEFI anses vara överlägset det äldre BIOS-systemet. För det första har det fullt stöd för hårddiskar större än 2 terabyte, vilket är en nödvändighet i dagens lagringsintensiva miljöer. Det erbjuder även säkerhetsfunktioner som Secure Boot, som kontrollerar att endast godkända operativsystem och drivrutiner får starta, vilket minskar risken för rootkits och andra attacker. UEFI har bättre möjligheter för konfiguration, och tack vare GUI och musstöd är det enklare för användare att navigera i inställningarna. Slutligen ger UEFI generellt sett snabbare POST och systemstart jämfört med BIOS, vilket gör det både effektivare och säkrare.

## 2.7 Secure Boot

### 2.7.1 Vad är Secure Boot?

Secure Boot är en säkerhetsfunktion som ingår i UEFI-standarden. Dess huvudsakliga uppgift är att kontrollera att operativsystemets bootloader är kryptografiskt signerad med ett godkänt certifikat. Genom att verifiera signaturen innan något laddas in i minnet säkerställer Secure Boot att endast legitim och auktoriserad kod körs under uppstartsprocessen. Om något försök görs att starta ett modifierat eller obehörigt system – exempelvis ett rootkit – stoppas processen omedelbart.

## 2.7.2 Fördelar

En av de största fördelarna med Secure Boot är att det skyddar systemet mot avancerade attacker som rootkits och bootkits – alltså skadlig kod som försöker gömma sig redan innan operativsystemet är igång. Secure Boot skapar en så kallad säker kedja av förtroende, från firmware till operativsystemets kärna. Den förhindrar att bootloadern manipuleras eller ersätts av obehörig kod, vilket gör att endast pålitlig programvara kan startas.

## 2.7.3 Nackdelar

Trots sina säkerhetsfördelar har Secure Boot vissa nackdelar. Eftersom det kräver att operativsystemet är kryptografiskt signerat kan det förhindra installation av alternativa operativsystem som inte är officiellt signerade – exempelvis vissa Linux-distributioner eller specialsystem. Det kan även krävas att användaren själv hanterar certifikat om man vill lägga till undantag, och i vissa fall behöver Secure Boot inaktiveras manuellt i firmwareinställningarna för att installationen ska fungera.

# 2.8 Bootloaders

## 2.8.1 Vad är en bootloader?

En bootloader är ett litet men kritiskt program som laddas in av datorns firmware – BIOS eller UEFI – direkt efter att den första systemkontrollen (POST) har genomförts. Bootloaderns uppgift är att initiera hårddisken, lokalisera och ladda in operativsystemets kärna (kernel) i RAM-minnet och därefter lämna över kontrollen till operativsystemet. Det är alltså bootloadern som är länken mellan hårdvaran och själva operativsystemet, och utan den skulle systemet aldrig kunna startas upp. Exempel på bootloaders inkluderar GRUB (Linux), Windows Boot Manager, och systemd-boot.

## 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

### 2.8.4 Uppstartssekvenser – exempel

#### *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

#### *macOS*

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

#### *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

### 2.8.5 Vanliga problem

- "No bootable device found" – fel boot order
- Korrupt bootloader – behöver repareras
- Secure Boot-blockering av osignerade OS

- Felaktiga partitionstabeller (MBR/GPT-konflikter)

## 2.8.6 Fördjupning

### 2.8.7 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.

### 2.8.8 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

### 2.8.9 Secure Boot

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

## 2.8.10 Fördjupningslänkar

1. [https://en.wikipedia.org/wiki/Power-on\\_self-test](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>

## 2.8.11 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?

## 2.8.12 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

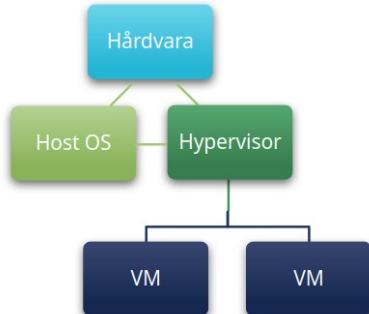
En bare-metal hypervisor (även kallad typ 1-hypervisor) är en virtualiseringsplattform som installeras direkt på serversns hårdvara – alltså utan att först installera ett vanligt operativsystem. Den fungerar i praktiken som maskinens "operativsystem", men med syftet att köra och hantera virtuella maskiner (VMs).

Det finns två huvudsakliga varianter av bare-metal hypervisorer:

### *Ren bare-metal (klassisk typ 1)*

Detta är en fristående hypervisor som bootar direkt från hårddisken, USB eller nätverk. Den har ett mycket slimmat gränssnitt och är optimerad för prestanda och stabilitet. Exempel: VMware ESXi och Xen.

Typ 1



### *Hybridmodell (minimal OS-komponent)*

Vissa lösningar räknas också som typ 1-hypvisorer trots att de använder ett specialanpassat underliggande operativsystem. Detta OS är dock endast till för hantering och påverkar inte VM:arnas resurser i samma grad som ett traditionellt operativsystem. Exempel: Proxmox VE (som använder KVM på Debian) och Microsoft Hyper-V i Windows Server (med sin "parent partition").

### 3.3.1 Fördelar och nackdelar

#### Fördelar:

- Hög prestanda: Direkt åtkomst till hårdvara ger minimal overhead.
- Säkerhet: Mindre attackyta än ett fullskaligt OS.
- Stabilitet: Utvecklade för dygnet runt-drift i produktionsmiljö.
- Skalbarhet: Bra stöd för stora VM-miljöer med många noder.

#### Nackdelar:

- Komplex installation: Kräver ibland särskilda verktyg och gränssnitt.
- Mindre användarvänligt: Få GUI-funktioner direkt på värden (många kräver fjärrstyrning).
- Överkvalificerat för små miljöer: Inte alltid idealiskt för personligt bruk eller testmiljöer på klientdatorer.

### 3.3.2 Exempel på plattformar

- VMware ESXi
- Microsoft Hyper-V (Server-installation)
- KVM (Linux-baserad)
- Proxmox

## 3.4 Hosted hypervisorer

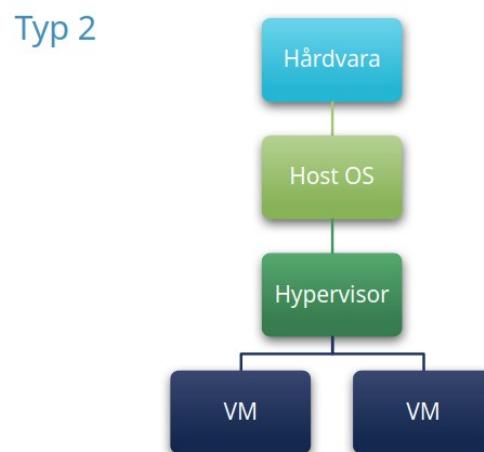
En hosted hypervisor (även kallad typ 2-hypervisor) installeras som ett vanligt program i ett befintligt operativsystem, till exempel Windows, macOS eller Linux. Det innebär att hypervisor körs ovanpå ett redan installerat system, vilket gör den enkel att komma igång med och särskilt användbar i labbmiljöer, utbildning och utveckling.

Till skillnad från bare-metal-hypervisorer (typ 1) kräver en hosted hypervisor ingen särskild serverhårdvara, utan kan köras direkt på en vanlig dator. Det gör det också möjligt att köra ett större urval av operativsystem – även sådana som inte alltid stöds officiellt av typ 1-hypervisorer. Ett exempel är att köra macOS i VirtualBox, vilket är tekniskt möjligt men bryter mot Apples licensvillkor om det inte sker på Apple-hårdvara.

Ett annat exempel på virtuell miljö som många känner igen är BlueStacks – en plattform som låter användare köra Android-appar på Windows. BlueStacks bygger på virtualiseringsteknik liknande en typ 2-hypervisor, även om den är mer specialiserad mot just Android. För användaren fungerar det ungefär som en vanlig app, men under ytan används tekniker som påminner om både emulering och virtualisering.

### 3.4.1 Fördelar och nackdelar

#### Fördelar:



- Enkel att installera och använda.
- Bra för labb, utveckling och utbildning.
- Kan köras på en vanlig dator utan krav på särskild serverhårdvara.
- Ger flexibilitet att köra olika operativsystem – även ovanliga eller experimentella.

#### Nackdelar:

- Något sämre prestanda jämfört med bare-metal-hypervisors.
- Värdoperativsystemet måste fungera korrekt – kraschar det, kraschar alla virtuella maskiner.
- Dubbla mjukvarulager (värd-OS + hypervisor) kan ge fler attackytter och ökad komplexitet.

### 3.4.2 Exempel på plattformar

- Oracle VirtualBox
- VMware Workstation / Fusion
- Parallels (för macOS)
- BlueStacks (Androidmiljö i Windows – inte en traditionell hypervisor, men bygger på liknande teknik)

## 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. En av fördelarna med detta är "skalbarhet". Du kan när som helst anpassa sin datorpark så att den kan bli större eller mindre utefter behov. Generellt

blir det dyrare i längden än att köpa själv, men ett bra komplement och en enkel lösning då du inte behöver stå med varesig hårdvara eller underhåll.

### 3.6 Containerar (kort introduktion)

Containrar är ett annat sätt att paketiera 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.

Ofta när ett program har testkörts, kanske i skolan, så säger eleven: "Men det fungerade ju i min dator". Med en container så kan du paketera "din dator" och på så sätt är du garanterad att det fungerar. Det är därför man säger att "containrar gör att det fungerar överallt - inte bara hos dig".

## 3.7 Fordjupande 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](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

### 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 partitionstabbellerna är:

- **MBR (Master Boot Record)**
  - o Äldre standard
  - o Max 4 primära partitioner
  - o Max 2 TB per disk

- o Sårbar: all information om diskens struktur lagras i en liten MBR-del – blir den förstörd kan allt gå förlorat
- **GPT (GUID Partition Table)**
  - o Nyare standard
  - o Nästan obegränsat antal partitioner
  - o Stöd för mycket större diskar
  - o Inbyggd redundans (lagrar flera kopior av partitionstabellen)
- **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 sparar. En partition är ett avgränsat område på en disk, medan en volym är en formaterad partition med ett specifikt filsystem – det är volymen som operativsystemet använder för att läsa och skriva data.

Vanliga filsystem:

- **FAT32**
  - o Stöd på nästan alla enheter
  - o Max filstorlek: 4 GB
  - o Bra för kompatibilitet, men begränsningar för stora filer
- **exFAT**
  - o Nyare variant
  - o Inga praktiska filstorleksgränser för vanliga behov
  - o Bra för USB-stickor mellan Windows och Mac
- **NTFS**
  - o Windows-standard
  - o Stöd för stora filer
  - o Rättigheter och journalföring

- **ext4**
  - o Vanligt i Linux
  - o Stabilt, snabbt, bra för servrar
- **APFS**
  - o Apples filsystem för macOS
  - o Modern funktionalitet: snapshot, kryptering

Tänk att du har en garderob. Stommen (skalet) motsvaras av disken. För att du skall kunna använda garderoben så måste du ha hyllor eller en stång. Hur du anpassar layouten i din garderob blir som en eller flera partitioner.

I din garderob har du även insatser i dina lådor eller korgar. Dessa motsvaras av volymen eller volymerna. Men volymerna kan ha begränsningar. Om lådan har många små fack (Fat 32) så passar inte stora saker som en 6Gb filt, men med större fack går det bättre.

Samtidigt så finns det fördelar och nackdelar med olika filsystem. Fat 32 till exempel är kompatibelt med nästan allt som kan läsa USB, till skillnad från ext4 som i stort sett bara kan läsas av Linux-system. För att kunna läsa det i Windows krävs att du har tilläggsprogram.

Windows 11 kräver exempelvis att du använder NTFS då rättighetshantering m.m. finns i filsystemet, till skillnad från Linux som har det i kärnan.

## 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 Sammanfattnings

- **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 Fordjupningslänkar

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. Förlara 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. Nämн 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% försättsvinna** 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

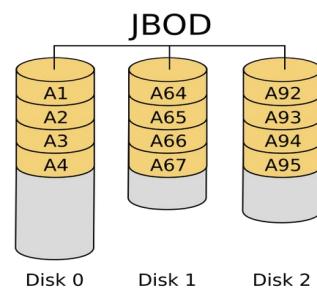
- **Hårdvaru-RAID** sker i en RAID-kontroller som sitter på moderkortet eller som separat kort. Operativsystemet ser hela RAID-arrayen som en vanlig disk.
- **Mjukvaru-RAID** sker i operativsystemet (t.ex. i Linux via mdadm). Det är billigare, mer flexibelt, men kan belasta CPU:n mer.

## 4.10.3 RAID-nivåer

### JBOD

JBOD står för Just a Bunch Of Disks, och det är precis vad det är. Till skillnad från en RAID så får du inte högre överföringshastigheter eller redundans.

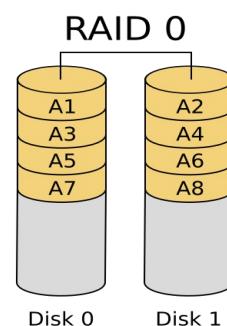
Däremot får du en länkad volym som kan nyttja alla hårddiskars utrymme även om de är olika stora



### RAID 0

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

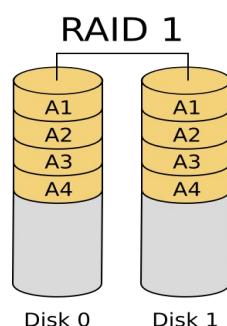
- Effektiv lagring:  $n * X * 0,85$
- Fördelar: Snabbt
- Nackdelar: Ingen säkerhet



### RAID 1

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

- Effektiv lagring:  $(n / 2) * X * 0,85$

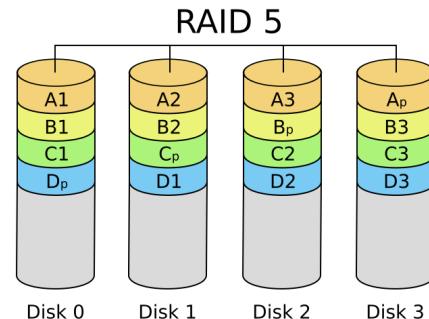


- Fördelar: Säkerhet
- Nackdelar: Hälften av utrymmet försvinner

### RAID 5

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

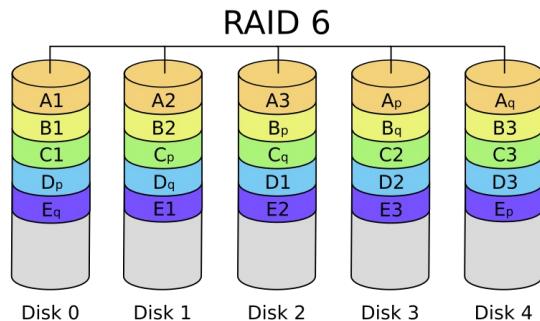
- Effektiv lagring:  $(n - 1) * X * 0,85$
- Fördelar: Bra balans mellan prestanda och säkerhet
- Nackdelar: Kan vara segt vid återuppbryggnad



### RAID 6

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

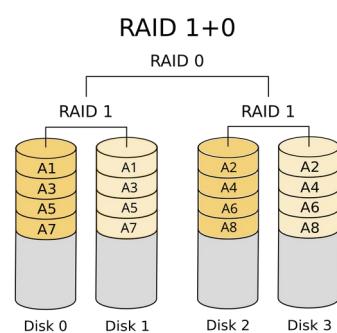
- Effektiv lagring:  $(n - 2) * X * 0,85$
- Fördelar: Mer säkerhet
- Nackdelar: Mindre lagringsyta, lägre skrivprestanda



### 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-arrayer 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.

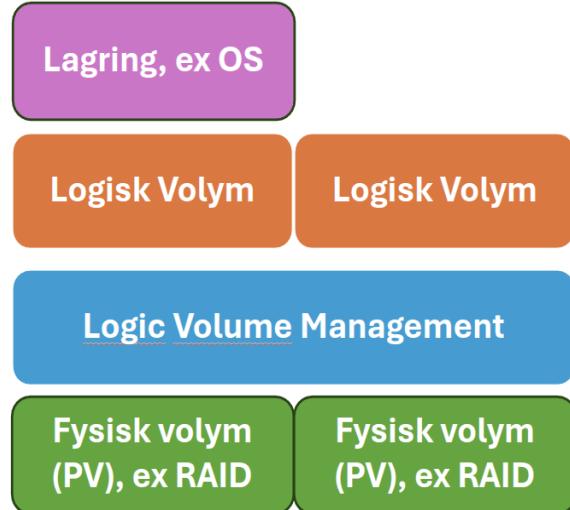
LVM används ofta tillsammans med RAID eller ZFS för att ge ännu större flexibilitet och kontroll över lagringen.

### Fördelar:

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

### Nackdelar:

- Lite mer komplext
- Mjukvarubaserat
- Windowsstöd saknas. Windows kan installeras på en LV men kan inte hantera LVM



## 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](https://openzfs.org/wiki/Main_Page)
5. <https://wiki.archlinux.org/title/LVM>

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

## 4.10.9 Egna anteckningar

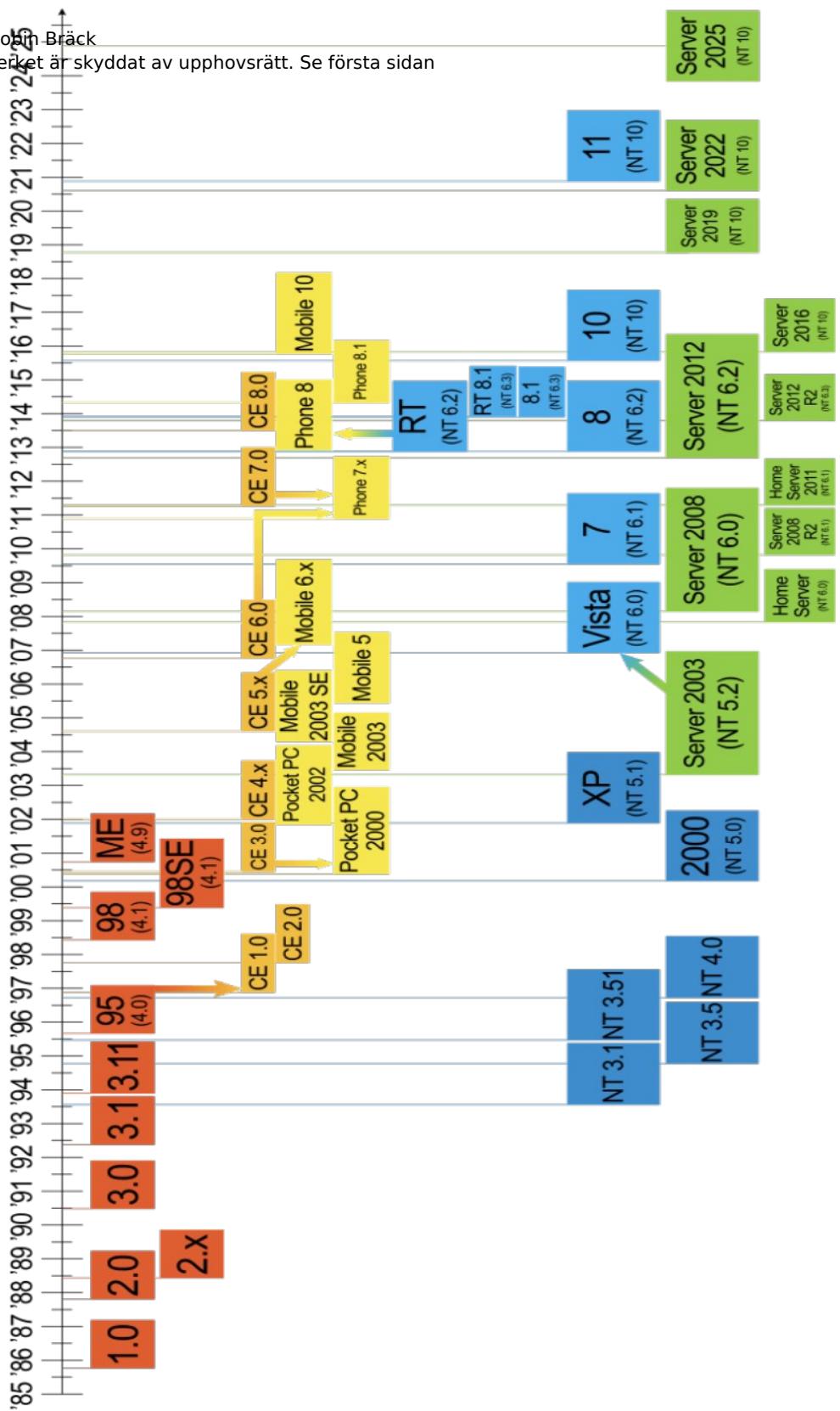
...

## 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.

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan



### 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.

➡ Däremellan släpptes också Windows ME (Millennium Edition) – en version som ofta kritiseras för sin instabilitet och snabbt ersattes av XP. Detta var dock det första konsument-OS som inte var beroende av MS-DOS för uppstart

#### ✓ Windows NT-linjen

- Parallelt 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 mikrokärnarkitektur.
- 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).

 Windows 11 är också det första operativsystemet från Microsoft som kräver TPM 2.0 och Secure Boot, vilket gjort att vissa äldre datorer inte går att uppgradera utan speciallösningar.

- 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.
- Det är i grunden 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](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.
  - Journalskriftning (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](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](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.
  - a. 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>
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:
  - a. Tryck **Shift + F10** för att öppna kommandoprompt.
  - b. Skriv: start ms-cxh:localonly
  - c. 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:
  - o *C:\Users\Namn\Documents*
  - o *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 administrörsrättigheter.

 **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 + andra 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?**
  - o Inställningar → Appar → Standardappar.

- **Vad gör den?**

- o Styr vilket program som öppnar olika filtyper, t.ex. webbsidor eller bilder.

## **Diskrensning (Disk Cleanup)**

- **Hur hittar du det?**

- o Sök i Startmenyn: *Diskrensning*.

- **Vad gör den?**

- o Tar bort tillfälliga filer och frigör utrymme på disken.

## **chkdsk**

- **Hur hittar du det?**

- o Kommandoprompt som administratör → skriv chkdsk.

- **Viktiga växlar:**

- o /f (fixar fel), /r (letar dåliga sektorer och återställer data).

- **Vad gör den?**

- o Kontrollerar och reparerar filsystemfel på disken.

## **Defragmentera / Optimize Drives**

- **Hur hittar du det?**

- o Sök i Startmenyn: *Defragmentera* eller *Optimize Drives*.

- **Vad gör den?**

- o Omorganiseringar filer på hårddisken för snabbare åtkomst (SSD använder trim).

## **Felsökning (Troubleshoot)**

- **Hur hittar du det?**

- o Inställningar → System → Felsökning.

- **Vad gör den?**

- o Guidade lösningar för vanliga problem (ljud, nätverk, uppdateringar).

## Kompatibilitetsläge (Compatibility settings)

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

## Tillförlitlighetshistorik (Reliability Monitor)

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

## Aktivitetshanteraren (Task Manager)

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

## Resursövervakaren (Resource Monitor)

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

## Msconfig / Systemkonfiguration

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

## Minnesdiagnostik

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

## Lokal säkerhetsprincip (Local Security Policy)

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

## Fjärrskrivbord (Remote Desktop)

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

## Registereditorn

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

## Enhetshanteraren (Device Manager)

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

## **Schemaläggaren (Task Scheduler)**

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

## **diskpart**

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

## **Utskriftshantering (Print Management)**

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

## **Hjälpmmedel (Accessibility)**

- **Hur hittar du det?**
  - o Inställningar → Hjälpmmedel.
- **Vad gör den?**
  - o 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
<b>Standard</b>	Ja/Nej	Vanligaste versionen, passar för de flesta servrar
<b>Datacenter</b>	Ja/Nej	För mycket stora organisationer, stöder fler VM:ar
<b>Core (utan GUI)</b>	Nej	Kommandoradsbaserad, används i större servermiljöer
<b>Desktop Experience</b>	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 administratörlösenord

 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 domänekontrollant
Skalbarhet	Liten miljö ( $\leq 10$ 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

 Tips: Även i en domänmiljö kan lokala användarkonton på klienterna förekomma. Det är vanligt att ha ett lokalt administratörskonto för felsökning, särskilt om nätverksanslutningen till domänekontrollanten inte fungerar.

## 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

## 6.2.6 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?

## 6.2.7 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/edition-s-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/>

## 6.2.8 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.

#### 7.1.1 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 nätwerk – 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.

### 7.1.2 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 **bilmmodell byggd kring motorn**, med olika inredning, reglage och målgrupper.

## 7.1.2 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.
- **Resurssnålt** - Linux fungerar ofta bra även på äldre datorer med begränsat RAM och lagring, till skillnad från nyare Windows-versioner som ställer högre hårdvarukrav.

## 7.1.3 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 känna 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 och programkompatibilitet inte alltid lika bra** - När det gäller spel har Steam och Proton gjort stora framsteg, och en

övervägande majoritet av titlarna på Steam fungerar numera i Linux. På programsidan finns verktyg som Wine, men det är sällan man kan köra nyare Windowsprogram helt smärtfritt.

- **Fler steg vid installation och underhåll** – Du förväntas ibland hantera saker manuellt, till exempel montering av diskar eller konfigurering av nätverk.

## 7.1.4 Fördjupningslänkar

1. Linux Foundation - <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>
5. Ubuntu - <https://ubuntu.com>
6. Debian - <https://www.debian.org>
7. Red Hat - <https://www.redhat.com>
8. Arch Linux - <https://www.archlinux.org>
9. Free Software Foundation - <https://www.fsf.org>
10. Steam Deck (Linux-baserad) - <https://www.steamdeck.com>

## 7.1.5 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?

## 7.1.6 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.

En spännande sak med många Linuxdistributioner är att de ofta kommer som en live-installation. Det innebär att du kan testa och köra operativsystemet direkt från ett USB-minne, utan att det påverkar ditt befintliga system.

Detta används också ofta av IT-personal för felsökning – eftersom man enkelt kan boota ett helt operativsystem från en USB-sticka och få tillgång till filsystem, nätverk och verktyg utan att behöva starta datorns installerade system.

#### *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 **signeras 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**.

### 7.2.1 Installation av Ubuntu Desktop i fysisk maskin/VM

Att kunna installera Ubuntu 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.

#### *7.2.1.1 Förberedelser*

##### Kontrollera systemkrav

- Processor: 64-bitars, 1 GHz eller snabbare
- RAM: Minst 2 GB (4 GB eller mer rekommenderas)
- Lagring: Minst 25 GB (helst 40+ GB)
- UEFI med Secure Boot (Ubuntu fungerar med detta aktiverat)

#### *7.2.1.2 Ladda ner Ubuntu Desktop ISO*

##### Steg för steg:

1. Gå till den officiella nedladdningssidan:  
<https://ubuntu.com/download/desktop>

2. Ladda ner den senaste LTS-versionen (t.ex. Ubuntu 24.04 LTS).
3. ISO-filen sparar lokalt på datorn.

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

#### *7.2.1.3 Skapa en startbar USB-sticka*

 Rekommenderade verktyg:

- Rufus (Windows): <https://rufus.ie>
- Balena Etcher (Windows/macOS/Linux): <https://etcher.balena.io>
- Startup Disk Creator (Ubuntu/Linux)

 Steg för steg:

1. Starta ditt valda verktyg.
2. Välj ISO-filen.
3. Välj din USB-enhet (minst 8 GB rekommenderas).
4. Klicka på "Start" eller "Flash".

#### *7.2.1.4 Ändra bootordning i BIOS/UEFI*

 Steg för steg:

1. Sätt in USB-stickan och starta om datorn.
2. Tryck rätt tangent för BIOS/UEFI (vanligtvis Del, F2, F10 eller Esc).
3. Leta upp Boot Order eller Boot Priority.
4. Välj USB-enheten som första alternativ.
5. Spara och avsluta.

#### *7.2.1.5 Installera Ubuntu Desktop*

 Steg för steg:

1. Välj språk och tangentbordslayout.
2. Anslut till Wi-Fi om du har trådlöst nätverk.
3. Välj installationsalternativ:
  - Normal installation – inkluderar webbläsare, kontorsprogram m.m.
  - Minimal installation – endast webbläsare och grundfunktioner

4. Markera:

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

5. Välj "Erase disk and install Ubuntu" (eller "Something else" för avancerade användare).

6. Skapa användare:

- Fullständigt namn
- Datornamn
- Användarnamn och lösenord

7. Klicka "Install" – installationen tar ca 10–20 minuter.

#### *7.2.1.6 Hantera MOK vid Secure Boot (valfritt)*

Om du installerat proprietära drivrutiner med Secure Boot aktivt:

1. Efter omstart visas blå MOK-meny.
2. Välj Enroll MOK.
3. Ange lösenordet du skapade under installationen.
4. Bekräfta och starta om.

#### *7.2.1.7 Kontrollera 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

#### *7.2.1.8 Installera Ubuntu Desktop i virtuell maskin*

Att installera Ubuntu i en VM (t.ex. VirtualBox eller VMware) är smidigt och säkert för labbmiljöer.

 Steg för steg:

1. Skapa en ny VM i VirtualBox eller liknande.
2. Välj typ: Linux och version: Ubuntu (64-bit).
3. Tilldela minst 2 GB RAM (helst 4+) och minst 25 GB disk.

4. Montera ISO-filen i VM-inställningarna.

5. Starta VM - Ubuntu-bootmeny visas.

6. Följ samma installationssteg som ovan.

 Tips: Installera "Guest Additions" (VirtualBox) eller "VMware Tools" efteråt för bättre integration (t.ex. skärmanpassning, delade urklipp).

## 7.2.2 Fördjupningslänkar

1. Ubuntu Desktop Download – <https://ubuntu.com/download/desktop>
2. Canonical – <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>

### 7.2.3 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?

## 7.2.4 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 **siffriform**, så kallad oktal notation:

Okta I	Rättigh et	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 Använtbara 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)

I bland 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, nätverksenhets)
- 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.

### 7.4.1 Installation av Ubuntu Server i fysisk maskin eller VM

Att installera Ubuntu Server är en grundläggande färdighet för den som vill jobba med servrar, nätverk och systemadministration. Här går vi igenom hur du installerar Ubuntu Server både på fysisk hårdvara och i en virtuell maskin – t.ex. i VirtualBox, Proxmox eller VMware.

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

##### 1. Ladda ner Ubuntu Server

Besök:

<https://ubuntu.com/download/server>

Ladda ner den senaste LTS-versionen (t.ex. Ubuntu Server 24.04 LTS).

##### 2. Skapa en startbar USB

Använd till exempel:

- Rufus (Windows) – <https://rufus.ie>
- Balena Etcher – <https://etcher.balena.io>
- Startup Disk Creator (på Ubuntu/Linux)

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

##### 3. Starta datorn från USB

Starta om datorn och välj att boota från USB-enheten.

Detta görs oftast via F12, F2, ESC eller DEL beroende på tillverkare.

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

Följ installationsguiden och välj rätt inställningar för språk och region.

#### *5. Nätverksinställningar*

Om du har en DHCP-server (t.ex. via Proxmox eller OPNsense) får servern IP-adress automatiskt.

Annars kan du välja att ställa in en statisk IP-adress direkt i installationen.

#### *6. Användare och lösenord*

Skapa ett användarnamn och välj ett starkt lösenord.

Du får möjlighet att installera OpenSSH-server – detta rekommenderas, så du kan ansluta till servern via SSH.

#### *7. Lagring och partitionering*

Välj hur disken ska användas:

- Use an entire disk – Ubuntu partitionerar automatiskt
- Custom storage layout – För dig som vill lägga upp egna partitioner (avancerat)

#### *8. Installera valfria tjänster*

Ubuntu Server erbjuder färdiga tjänstegrupper att installera direkt:

- OpenSSH
- Samba
- LAMP / LEMP
- Docker
- Print server
- ... och fler

 För skolsammanhang är det ofta bäst att börja utan tillval – installera endast det du behöver i efterhand.

#### *9. Starta installationen*

När alla inställningar är klara startar installationen.

Det tar vanligtvis 5-10 minuter. Servern startar om när det är klart.

### 7.4.2 Installera Ubuntu Server i virtuell maskin

Ubuntu Server kan installeras i en VM för labb och testning.

Använd till exempel VirtualBox, VMware eller Proxmox.

 Steg för steg:

## 1. Skapa ny virtuell maskin

- Typ: Linux
- Version: Ubuntu (64-bit)
- RAM: minst 1-2 GB (helst 4+)
- Disk: minst 8 GB (helst 20+ GB)

2. Montera ISO-filen i VM-inställningarna som startskiva.
3. Starta VM:n – installationsguiden körs automatiskt.
4. Följ samma installationssteg som ovan (språk, nätverk, användare, lagring).

## 7.4.3 Webmin – Grafiskt gränssnitt för serveradministration

Att arbeta i terminalen är kraftfullt, men kan kännas svårt i början. Webmin är ett webbaserat gränssnitt för att administrera Linux-servrar visuellt.

Med Webmin kan du hantera:

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

Det är ett utmärkt sätt att kombinera pedagogik med verlig systemadministration.

*Så här installerar du Webmin på Ubuntu Server*

Webmin har ett officiellt installationsscript:

```
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 din webbläsare kommer att ge en varning – det är normalt.

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se första sidan

Logga in med samma användarnamn och lösenord du skapade vid 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>
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.)*

## 7.5 Linux grundläggande kommandon

Tanken med denna handledning är att du på egen hand övar kommandon i Linuxterminal, så att du blir mer bekväm med dem. En grundläggande sak att ALLTID ha koll på är att Linuxterminalen är känslig för stora och små bokstäver. Det innebär att /download och /Download kommer den att läsa som två olika sökvägar. Det samma gäller kommandon m.m. Skulle du skriva en stor bokstav istället för en liten så kan fel sak hända, eller ingenting alls. Detta är viktigt att hålla koll på, framförallt när det gäller växlar där -g och -G betyder två olika saker i till exempel kommandot useradd (kolla genom att skriva useradd --help).

### 7.5.1 Grundläggande tips

I Bash finns mycket hjälp, om man bara vet var man skall leta. Det finns 3 olika sätt att få information:

#### **Kommandot whatis**

Testa att skriva whatis ls. Här får du en enkel förklaring på vad kommandot gör

#### **Kommandot man**

Testa att skriva man apt-get. Här får du den fullständiga manualen till kommandot. Mycket utförligt

#### **Växeln --help**

Testa att skriva rm --help. Här får du ett kortare utdrag ur manualen, som framförallt går igenom växlarna i programmet/funktionen

En av Linux grundläggande saker är att allt är en fil, en fil som kan redigeras på ett eller annat sätt. Detta är viktigt att minnas, främst då det handlar om rättigheter, eftersom det ger dig möjligheten att enkelt styra rättigheter på allt från enheter till program.

En annan sak att minnas är Linux sätt för att dölja en fil. Skrivs en punkt (.) före filnamnet så syns den inte, och är därmed dold. Du kan se den genom kommandon som ls -a (visa alla filer) eller att du väljer att visa alla filer grafiskt.

#### **Katalogstruktur:**

/bin/

Binary. Körbara kommandon för alla användare

/boot/

Filer för bootloader, även kärnan

/media

Usb enhet m.m. syns här. Monteras

/dev/

Devices, systemenheter

/etc/

Konfigurationsfiler

/home/

Hemkatalog

/root/

Inaktiverad admin

/lib/

Biblioteksfiler, typ dll-filer. Gemensamma programbibliotek

/mnt/

Mount

/sbin/

Securebin, krävs root

/srv/

Services

/sys/

Viktiga systemfiler

/usr/

Program och verktyg

/var/

Variabler, typ log, spool

## 7.5.2 Kommandon

### Kommandot ls

Öppna en terminal, titta var du befinner dig. Läser du från vänster till höger så börjar det med:

Användare@datornamn:sökväg\$

Dollartecknet på slutet visar att du är inloggad som användare. Hade det istället varit ett # så skulle du ha varit root. I vissa fall finns ingen sökväg utan endast ett ~(vågecken). Det indikerar din (inloggad användares) hemmapp. Om vi har en användare som heter batman så finns dennes hemkatalog i: /home/batman/. Det skrivs aldrig ut, utan istället skrivs ~, det samma gäller fortsättningsvis så batmans skrivbord har sökvägen ~/Desktop vilket betyder /home/batman/Desktop

Testa att skriva ls

Vad händer, och vad ser du

ls har två viktiga växlar att ha koll på, -a och -l, vad gör de olika (tips du kan även kolla --help)

Med hjälp av ls kan du lista innehållet i en mapp. Skriver du ingen sökväg efter så listar du automatiskt innehållet i den sökväg du befinner dig. Skriver du istället ls /etc/ så listas innehållet i /etc.

### *Kommandot cd*

Med kommandot cd ändrar du vilket mapp du befinner dig i. Det finns flera olika sätt att förflytta sig. Antingen så använder du Absoluta sökvägar, till exempel cd /etc/dpkg, eller med hjälp av Relativa sökvägar, då förflyttar du dig från mapp till mapp. Exempel:

cd /

cd /etc

cd /dpkg

Två sätt att komma till samma ställe

För att kliva ett steg bakåt, från dpkg till etc, skriver du cd .. . Vill du istället gå direkt till roten skriver du cd / och vill du komma till din hemkatalog skriver du cd ~.

### *Kommandot mkdir*

mkdir är kommandot som du använder för att skapa mappar. Till exempel mkdir test. Då kommer det att skapas en mapp i den sökväg som du befinner dig, med namnet test. Skriver du istället mkdir med en sökväg, till exempel mkdir /home/test kommer det att skapas en mapp med namnet test på den sökvägen.

mkdir är ett kommando med många växlar. En mycket praktisk sådan är -p. Den gör en hel katalogstruktur direkt. Exempel:

Du skall skapa mappen /home/ee20/elev/inlämningar

Mappen elev finns inte än, inte heller inlämningar. Skriver du mkdir /home/ee/elev/inlämningar kommer du få ett felmeddelande, men skriver du mkdir -p /home/ee/elev/inlämningar kommer hela strukturen att skapas

### **Kommandona mv, cp, rm och rmdir**

Kommandot mv är det du använder för att flytta en fil. Det används även för att byta namn på filer. Skriver du mv /home/user/fil /home/user/filen så kommer den fil du hänvisar till att byta namn. Om vi bryter ner kommandot så vad du faktiskt skriver är: mv (flytta) filen fil som ligger i /home/user/ till mappen /home/user och där skall den heta filen

Kommandot cp är för att kopiera en fil.

rm och rmdir är två olika kommandon för att radera. rmdir är det kommando som sällan används på grund av att det enbart tar bort tomma mappar.

### *Kommandot adduser*

Kommandot adduser skapar en fullständig användarprofil. Det vill säga att den skapar en användare med mappar, script och all information som behövs för att kunna logga in fysiskt vid datorn. Eventuella förändringar från standardmall sker med hjälp av växlar/flaggor. Så vill du att användaren skall ha förändringar från en standardmall måste du skriva in det. Exempel:

```
sudo adduser britney
```

Det kommer att skapa användaren britney, som kommer kunna logga in på datorn när hon sitter vi den

### *Kommandot useradd*

Kommandot useradd skapar endast ett skal av en användare. För att den skall kunna användas måste du använda flaggor/växlar. Detta är ett mycket smidigt sätt att skapa användare som bara har den profil som behövs. En användare som skall fjärransluta mot en server behöver ju ingen hemmapp om det är en filserver som hen ansluter till. Då räcker det med användarnamn och lösenord, samt grupp tillhörigheter. Exempel:

```
sudo useradd -M batman
```

Då skapas användaren batman, men växeln -M säger att det är en användare som inte skall ha någon hemmapp.

### Kommandot passwd

Det är kommandot för att skapa, eller ändra en användares lösenord. Exempel:

```
sudo passwd mentor
```

Därefter får du skriva in det nya lösenordet

### *Kommandona groupadd och addgroup*

Kommandot groupadd skapar en ny grupp, som du kan lägga till användare i (se ovan). Dessa grupper måste vara skapade innan du addrar en användare till dem. Exempel

```
sudo groupadd itelever
```

Skall du lägga till en användare till en grupp så använder du istället kommandot addgroup. Exempel:

```
sudo addgroup robin itelever
```

### *Kommandot usermod*

Kommandot används för att ändra befintlig användare, till exempel att lägga till fler grupper, flytta hemkatalog eller dylik. Detta är en förlängning av kommandona adduser/useradd. Exempel:

```
usermod -d /home/itelever/robin robin
```

Det kommer flytta användaren robins hemmapp till /home/itelever/robin

### *Kommandot chmod*

Chmod är det kommando som används för att styra rättigheter för en fil eller mapp. Det kan antingen göras med oktaler eller tecken. Oktaler bygger på en enkel kombination av siffror. Vill du ge rättigheter så kombinerar du bara de värdena som motsvarar rättigheten.

Read (läsa): 4

Write (skriva/ändra): 2

Execute (utföra/köra): 1

Inga rättigheter: 0

Rättigheter i Linux visas (oftast) i en 3-sifferkombination, exempelvis: 644. Den första siffran (6) berättar vad ägaren till dokumentet har för rättigheter, den andra (4) vilka rättigheter användare som har tillgång till gruppen och den sista (4) visar vilka rättigheter alla andra användare har till filen. Vill du ändra en fils rättigheter så adderar du siffrorna ovan, per position. Exempel:

```
sudo chmod 640 klasslista
```

I ovan exempel så kommer ägaren till filen klasslista att ha read och write (läsa och skriva), användare som har tillgång till gruppen har read (läsa) och övriga användare har ingen åtkomst.

### *Kommandot umask*

Umask är kommandot som styrde automatiska rättigheter som ges en fil eller mapp när den skapas. Alla filer och mappar har ursprungligen värdena 777 (för mappar) och 666 (för filer). Det innebär att den som skapar den har fullständiga rättigheter (7/6), de i samma grupp har samma (7/6) och alla andra i datorn har även de fulla rättigheter till filen (7/6). Det umask gör är att automatiskt ta bort rättigheter. Man kan säga att den fungerar omvänt mot värdet. Exempel:

Filen klasslista skapas. Då har den rättigheterna 666. Kommandot umask används hela tiden och har värdet 022. Resultatet av rättigheterna kan du se nedan:

777

-022

-----

755

Umask kan styras, men det gäller bara under tiden som Bash-fönstret är öppet. Stängs det eller ett nytt fönster öppnas så gäller bara förändringen i det fönster där umask ändrades. Exempel:

sudo umask 000

Ovan kommando kommer innebära att alla kommer ha fullständiga rättigheter till filen/mappen och alla andra filer/mappar som skapas. Förklaring:

777

-000

-----

777

### *Kommandot chown och chgrp*

chown och chgrp ändrar ägare respektive grupp tillhörighet till en fil eller mapp. Detta kan göras om ägandeskapet för en fil skall ändras eller om den skall göras tillgänglig för en annan grupp. Exempel

sudo chown klasslista rektor

### *Kommandot touch*

Touch är kommandot som har två syften. Dels så kan det användas för att skapa en tom fil. Exempel:

touch examen

Ovan kommando skapar filen examen.

Det kan också användas till att ”bumpa” en fil, det vill säga att uppdatera datum på den. Ett sätt att uppdatera senast sparad-datumet enkelt

### *Kommandona userdel och groupdel*

groupdel används för att ta bort en grupp ur systemet. Så länge det inte är någon användare som har den som primär grupp går gruppen att radera utan växlar. Exempel:

sudo groupdel itelever

userdel är kommandot som används för att radera en användare. Det är ett kommando som generellt används med växlar då kommandot ensamt bara tar bort användarnamnet men ingenting annat. En vanlig växel att använda är -r som tar bort användarens hemkatalog. Exempel:

```
sudo userdel -r grundskolan
```

### *Kommandot APT (och apt-get som ibland används på samma sätt)*

APT är namnet på pakethanteraren, funktionen som installerar program i Debiantsystem (Debian, Linuxdistribution som bland annat Ubuntu baseras på). APT har underfunktioner som är viktiga att kunna:

```
apt update - Laddar ner nya listor från repositories
```

```
apt upgrade - Uppdaterar alla program och operativsystemet
```

```
apt remove [programnamn] - Avinstallerar program
```

```
apt autoremove - Tar bort överflödiga programpaket
```

Exempel:

```
sudo apt upgrade
```

### *Kommandot nano*

Nano är namnet på den textredigeraren (textredigerare) som ligger i Bash som standard. Det är som en variant på Windows Notepad (Anteckningar). Exempel:

```
nano narvaro
```

Ovan kommando öppnar filen narvaro i en textredigerare

### *Kommandona getent, cat och grep*

Cat är funktionen för att skriva ut innehållet i en fil, direkt på skärmen. Getent gör ganska liknande men den söker enbart i vissa specifika databaser (ahosts, ahostsv4, ahostsv6, aliases, ethers (Ethernet adresser), group, gshadow, hosts, netgroup, networks, passwd, protocols, rpc, services, och shadow). Exempel:

```
getent group robin
```

Det kommer lista alla grupper som användaren robin tillhör

```
cat klasslistaitprogramet
```

Det kommer lista innehållet i filen klasslistaitprogramet. I fallet cat så går det att lista flera filer på en gång, det är bara att lista dem efter varandra.

Grep kommandot används för att söka efter en speciell sträng (ord) i en fil. När den hittar det så kommer hela den raden skrivas ut på skärmen. Exempel:

grep itprogrammet karlstad

Då kommer du att söka efter alla rader i dokumentet karlstad som innehåller itprogrammet. Även med detta kommando så kan du lista flera sökord på en gång.

### *Kommandona ifconfig och ip*

ifconfig används för att visa information om din nätverksanslutning. Det är Linux motsvarighet till ipconfig. Exempel

ifconfig eth0

Det kommer visa information om ethernetanslutning 0 (den första). Dock är kommandot “deprecated”, det vill säga att det inte rekommenderas längre. Istället rekommenderas att använda kommandot ip.

Ip kan användas till mycket mer, men framförallt används det på samma sätt som ifconfig. Exempel

ip address show eth0

Det kommer göra samma sak som ifconfig eth0.

På serversidan har kommandot ifconfig uteslutits helt och istället är det ip som används

### *Kommandona locate och find*

Locate är ett program som oftast behövs laddas ner (heter mlocate). När det väl är nedladdat behövs oftast databasen uppdateras. Det gör med kommandot sudo updatedb. När hela disken är indexerad (genomsökt) kan du söka snabbt och enkelt efter filnamn i datorn. Exempel:

locate .bashrc

Då kommer locate att leta i datorn efter filen .bashrc och ge dig sökväg(ar) tillbaka.

find är egentligen ett väldigt avancerat kommando, men används oftast bara för att leta filer. För att se hur kommandot används, kolla denna länk:

<https://www.geeksforgeeks.org/find-command-in-linux-with-examples/>

### *Kommandot su*

su betyder switch user. Detta gör det lätt att ”hoppa” mellan användare i terminalen. Du kan enkelt logga in som en annan användare. Exempel:

su rektor

Du går enkelt tillbaka till din användare med kommandot exit

### 7.5.3 Bash-scriptning

Grunderna i bashscriptning är enkla. Skapa en textfil. Inled textfilen med shebang (#!/bin/bash) för att peka på var kommandon finns

#### *Variabel*

En variabel i bash kan innehålla vad som helst. Du deklarerar (skapar) en variabel genom att döpa den, och ha en text efteråt

variabel=innehåll

för att "kalla på variabeln" sätter du ett dollartecken framför

\$variabel

#### *Funktion*

En funktion illustreras genom att det är en parentes runt kommandot. Om en variabel skall innehålla en funktion blir exemplet:

variabel=\$(whoami)

Kommandot echo

Kommando för att skriva ut ren text på skärmen

echo "En vacker rad text"

### 7.5.4 Kommandot whiptail

Whiptail är ett program, eller en funktion för att skapa dialogrutor i bash.

#### **Whiptail Infobox**

!/bin/bash

TERM=ansi whiptail --title "Viktig info" --infobox "Jag är bäst" 8 20

TERM=ansi är definition av terminal

whiptail kollar på programmet

-title är rubrik

- infobox definierar vad det är för typ av ruta
- 8 20 är rader och kolumner som rutan täcker

### *Whiptail messagebox*

```
#!/bin/bash

TERM=ansi whiptail --title "Inforuta" --msgbox "Jag är bäst och ni suger!" 8 78
```

En messagebox är som en infobox, men du måste bekräfta med Ok

### *Whiptail inputbox*

```
#!/bin/bash

valfrittvariabelnamn=$(whiptail --inputbox "What is your favorite?" 8 39
Skit_du_i_det --title "Inputbox" 3>&1 1>&2 2>&3)

# A trick to swap stdout and stderr.

exitstatus=$?
```

Inputbox ger dig möjlighet att mata in värde till en variabel, och återanvända det sedan

### *Whiptail yes/no-box*

```
#!/bin/bash

if whiptail --title "Example Dialog" --yesno "This is an example of a yes/no box." 8
78; then
    echo "User selected Yes, exit status was $?."
else
    echo "User selected No, exit status was $?."
fi
```

### *Whiptail passwordbox*

```
#!/bin/bash

password=$(whiptail --passwordbox "please enter your secret password" 8 78
--title "password dialog" 3>&1 1>&2 2>&3)

# A trick to swap stdout and stderr.
```

Gör samma sak som inputbox, men döljer input

### *Whiptail - Menyer*

#### *Vanlig meny*

```
#!/bin/bash

whiptail --title "Menu example" --menu "Choose an option" 25 78 16 \
"--<-- Back" "Return to the main menu." \
"Add User" "Add a user to the system." \
"Modify User" "Modify an existing user." \
>List Users" "List all users on the system." \
"Add Group" "Add a user group to the system." \
"Modify Group" "Modify a group and its list of members." \
>List Groups" "List all groups on the system."
```

Värdena som ges till --menu är:

```
--title ("Menu example")
-menu ("Choose an option")
Höjden på dialogrutan (25)
Bredden på dialogrutan (78)
Höjden på menyn (16)
```

Resten av värdena är en lista på menyval i formatet av "tags" och "items", där taggen är namnet på valet (som skickas till stderr) och items är beskrivningen i menyn

### Check list

```
#!/bin/bash

whiptail --title "Check list example" --checklist \
"Choose user's permissions" 20 78 4 \
"NET_OUTBOUND" "Allow connections to other hosts" ON \
"NET_INBOUND" "Allow connections from other hosts" OFF \
"LOCAL_MOUNT" "Allow mounting of local devices" OFF \
"REMOTE_MOUNT" "Allow mounting of remote devices" OFF
```

Om du vill ge möjligheter, som inte lämpar sig i en meny så kan du ha en checklista istället

### Radio list

```
#!/bin/bash

whiptail --title "Radio list example" --radiolist \
"Choose user's permissions" 20 78 4 \
"NET_OUTBOUND" "Allow connections to other hosts" ON \
"NET_INBOUND" "Allow connections from other hosts" OFF \
"LOCAL_MOUNT" "Allow mounting of local devices" OFF \
"REMOTE_MOUNT" "Allow mounting of remote devices" OFF
```

En radiolista ger användaren möjlighet att välja en sak från listan (med space) och brekrätta med enter

## Progress

```
#!/bin/bash

{
    for ((i = 0 ; i <= 100 ; i+=5)); do
        sleep 0.1
        echo $i
    done
} | whiptail --gauge "Please wait while we are sleeping..." 6 50 0
```

### *Slumpa genom en string*

För att slumpgenerera meddelanden eller dyl, kan du använda dig av en sträng. Den bygger i grund och botten på 3 olika variabler som samarbetar.

Kod:

```
variabel1=("1" "2")
#Skapa en variabel, inom citationsteknen kan du skriva olika saker, som kommer tolkas som en sträng
variabel2=$(( RANDOM % ${#variabel1[@]} ))
variabel3=${variabel1[$variabel2]}
```

## 7.5.5 Loopar

Loopar är grunden för all typ av programmering. Generellt kan man säga att, om du använder samma kommando fler än två gånger i rad för att utföra något, är det lämpligt att göra en loop.

De loopar vi jobbar med, framförallt är:

### *While-loop*

```
#!/bin/bash
counter=5
```

```
factorial=1
```

```
while (( counter > 0 )); do
    factorial=$(( factorial * counter ))
    ((counter -- ))
done
echo $factorial
```

En While-loop är en loop som upprepar saker fram tills påståendet är korrekt. I ovan fall så kommer den att upprepa funktionen tills dess att variabeln counter är större än noll

### *For-loop*

```
#!/bin/bash

string="0 1 2 3 4 5 6 7 8 9"
for i in $testvariabel;
do
    echo $i
done
```

En For-loop upprepar något tills dess att den fullgjort antal iterationer, upprepningar. I ovan exempel så använder den variabeln string, och upprepar (do) för varje värde i strängen.

### *If-sats*

```
#!/bin/bash

if [ "$(whoami)" == "root" ]; then
    echo "Du är root-användaren. Du har fullständiga rättigheter."
```

```
elif groups | grep -q "sudo"; then
    echo "Du är en sudo-användare. Du har avancerade rättigheter."
else
    echo "Du är en vanlig användare. Begränsade rättigheter."
fi
```

En if-sats i Bash används för att testa ett påstående. Det kan vara antingen ett aritmetiskt uttryck (beräkningar) eller ett logiskt villkor. För att Bash ska förstå skillnaden används:

`$(( ... ))` för aritmetiska beräkningar.

`[ ... ]` för logiska villkor.

Vanliga jämförelseoperatorer:

`<` : Mindre än

`<=` : Mindre än eller lika med

`==` : Lika med

`>` : Större än

`>=` : Större än eller lika med

`!` : Inte

Specifika Bash-operatorer för siffror:

`-lt` : Mindre än (Less than)

`-le` : Mindre än eller lika med (Less or equal)

`-eq` : Lika med (Equal)

`-gt` : Större än (Greater than)

`-ge` : Större än eller lika med (Greater or equal)

`-ne` : Inte lika med (Not equal)

If-satsen bygger på en rad där påstående testas

Exempel: if [ "\$(whoami)" == "root" ]; then

Raden under är vad som skall hända

elif motsvaras av Om inte, så om if inte stämmer så skall programmet göra följande:

Exempel: elif groups | grep -q "sudo"; then

Återigen så är raden under vad som skall hända

If-satser brukar avslutas med else. Om inget annat stämmer. I else-fallet används inga parenteser eller klamrar, då det inte testas något.

Exempel: else

Men återigen så är raden under vad som skall hända

Viktigt är att:

- Alla påståenden som skall testas, avslutas med ; (semikolon)
- Hela if-satsen avslutas med att stänga den. Då skriver du if baklänges (fi)

### *Skapa en funktion*

Om du vill skapa en funktion som skall köras i ditt script. Till exempel att du kallar på en for-loop

```
#!/bin/bash

create_users() {

    for user in "${user_list[@]}"; do
        sudo useradd -m $user
        echo "Användare $user skapad." | tee -a $log_file
    done
}
```

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

Då skriver du namnet följt av () och allt du vill ha i din funktion mellan {}

Sedan kan du kalla på funktionen, på samma sätt som vilket annat program som helst

```
option=$(whiptail --title "Meny" --menu "Välj ett alternativ:" 15 40 3 \  
1 "Skapa användare" \  
2 "Ping server" \  
3 "Avsluta" 3>&1 1>&2 2>&3)  
case $option in  
1)  
    create_users  
;;
```

## 7.5.6 Diskussions- och kontrollfrågor

1. Vad innebär det att Linux är skiftlägeskänsligt?
2. Vilka tre sätt kan du använda för att få hjälp med kommandon i Bash?
3. Vad gör kommandot whatis?
4. Vad är skillnaden mellan for- och while-loopar i Bash?
5. Vad används whiptail till?
6. Hur skapar man ett enkelt Bash-script?
7. Varför är det viktigt att förstå terminalen även om man använder GUI?
8. Vad händer om man använder en stor bokstav i ett kommando som är skiftlägeskänsligt?
9. Ge exempel på två kommandon för att navigera i filsystemet.
10. Vad betyder kommandot chmod +x filnamn.sh?

## 7.5.7 Fördjupande länkar

- <https://linuxjourney.com/>
- <https://www.gnu.org/software/bash/manual/>
- <https://ubuntu.com/tutorials/command-line-for-beginners>
- <https://tldp.org/LDP/Bash-Beginners-Guide/html/>
- <https://www.shellscript.sh/>

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

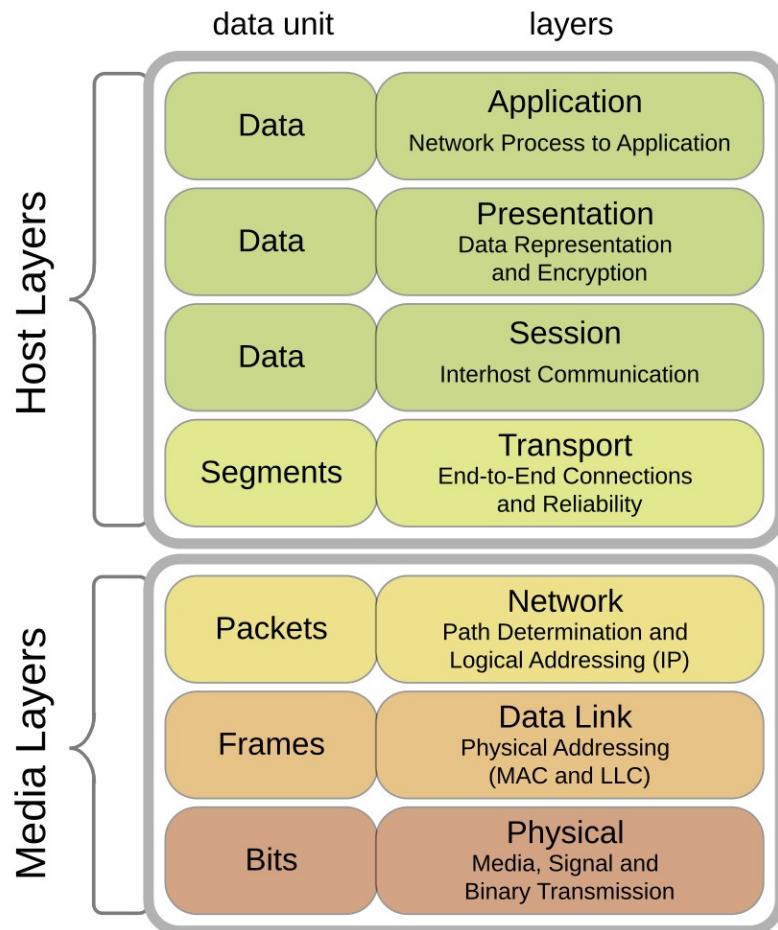
## 7.5.8 Egna anteckningar

## 8 OSI-modellen

När du skickar ett meddelande i Teams, spelar ett spel online eller surfar på en hemsida, så händer det faktiskt hundratals saker i bakgrunden. Men hur håller vi ordning på allt detta?

OSI-modellen är som ett kartläggningssystem – den delar upp kommunikationen i sju tydliga lager, där varje lager har en specifik roll. Från den fysiska kopparträden längst ner till den app du ser på skärmen högst upp.

Syftet är inte att varje lager alltid existerar exakt så här i verkligheten – utan att modellen ska hjälpa oss att förstå, felsöka och designa nätverk på ett strukturerat sätt.



### 8.1 Uppkomst – Bakgrund 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 en teoretisk modell, används den dagligen inom IT för att förstå, felsöka och strukturera nätverkskommunikation. Den hjälper både tekniker och utvecklare att prata samma språk när det gäller hur data färdas genom ett nätverk.

Exempel på användningsområden:

- Felsökning av nätverk: En tekniker kan systematiskt undersöka var problemet ligger – är det i applikationen (lager 7), i nätverket (lager 3) eller i själva kabeln (lager 1)?

- Kommunikation mellan IT-personal: Modellen fungerar som en gemensam karta som underlättar dialog mellan olika team, t.ex. mellan nätverkstekniker och mjukvaruutvecklare.
- Utbildning: OSI används som ett pedagogiskt verktyg för att visa hur komplex nätverkskommunikation egentligen är – men hur den också kan delas upp i hanterbara delar.
- Utveckling av protokoll och system: När nya tekniker eller lösningar utvecklas kan man specificera vilket lager de tillhör – t.ex. HTTPS (lager 7) eller TCP (lager 4).

### Tabellöversikt över OSI-lagren

Lager	Namn	Funktion	Exempel
7	Application	Det användaren ser och använder	HTTP, DNS, FTP, SMTP
6	Presentation	Översättning, kryptering, komprimering	TLS/SSL, JPEG, UTF-8
5	Session	Håller kommunikationen vid liv	NetBIOS, RPC
4	Transport	Levererar data, styr flöde och fel	TCP, UDP
3	Network	Hittar vägen mellan avsändare och mottagare	IP, ICMP
2	Data Link	Adressering i lokala nätverk	Ethernet, MAC-adress, ARP
1	Physical	Elektriska signaler, kablar, radio	Nätverkskabel, fiber, Wi-Fi, Bluetooth

 Tips: Tänk på OSI-modellen som en "nätverkslasagne" – varje lager bygger på det under, men har sin egen smak och funktion.

## Minnesregler

Ett klassiskt sätt att komma ihåg lagren (uppfirån och ner) är:  
"All People Seem To Need Data Processing"

Vill du gå nerifrån och upp? Då kan du använda:  
"Please Do Not Throw Sausage Pizza Away"

## Exempel: Hur en webbsida skickas genom OSI-modellen

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

- Lager 7 (Applikation): Webbläsaren (t.ex. Firefox eller Chrome) skapar en begäran om att hämta webbsidan.
- Lager 6 (Presentation): Webbsidan kan krypteras med TLS/SSL – och sedan avkodas på rätt sätt (t.ex. text i UTF-8, bilder i JPEG).
- Lager 5 (Session): En session upprättas mellan din dator och webbservern, så kommunikationen kan fortsätta utan att kopplas ner efter varje begäran.
- Lager 4 (Transport): Begäran delas upp i segment. TCP ser till att segmenten kommer fram i rätt ordning, utan förlust.
- Lager 3 (Nätverk): IP-protokollet lägger på avsändar- och mottagaradresser – som en poststämpel.
- Lager 2 (Datalänk): MAC-adresser används för att skicka paketet inom det lokala nätverket.
- Lager 1 (Fysiskt): Data omvandlas till elektriska signaler eller radiosignaler och skickas via kabel eller trådlöst.

När informationen når mottagaren går den uppåt genom samma lager – tills webbsidan visas i din webbläsare.

## 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>

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

## 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 kommunikation från obehörig åtkomst, skador eller störningar. Det gäller både tekniska lösningar och hur människor agerar. Målet är att skydda både filer och människor – oavsett om det gäller en dator hemma, ett företagsnätverk eller ett globalt molnsystem.

IT-säkerhet är inte bara något för experter. Det påverkar alla som använder teknik i vardagen. En elev som glömmer logga ut från en skolplattform, ett företag som drabbas av ransomware, eller en användare som klickar på en blufflänk – alla dessa scenarier handlar om IT-säkerhet.

Att arbeta med IT-säkerhet är lite som att låsa dörren, dra ur nyckeln och dessutom se till att ingen kan klättra in genom fönstret – samtidigt som man också litar på att ingen på insidan öppnar för främlingar.

#### 9.1.1 Informationssäkerhet vs IT-säkerhet

Det är lätt att blanda ihop begreppen informationssäkerhet och IT-säkerhet, men de är inte riktigt samma sak.

- Informationssäkerhet handlar om att skydda information – oavsett om den är digital, på papper eller i huvudet på en medarbetare.
- IT-säkerhet är en del av informationssäkerheten, men fokuserar på digital teknik: datorer, nätverk, program och system.

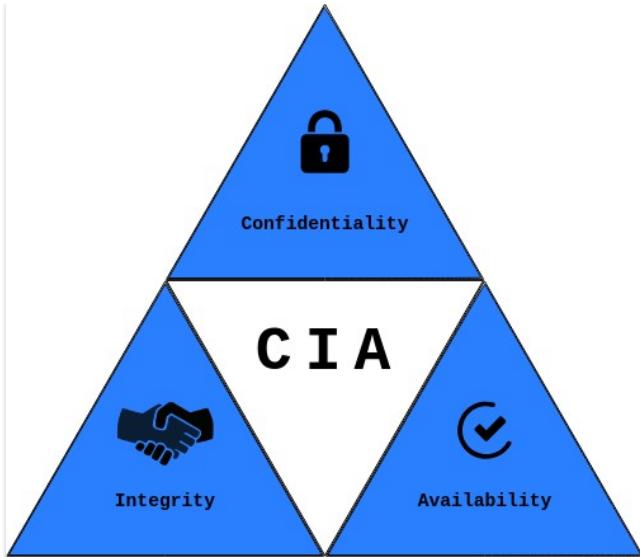
Exempel: Om ett USB-minne med känsliga filer glöms på bussen är det ett problem med informationssäkerheten. Om USB-minnet dessutom saknar kryptering och lösenordsskydd – då handlar det även om IT-säkerhet.

#### 9.1.2 CIA – Confidentiality, Integrity, Availability

Inom IT-säkerhet pratar man ofta om CIA-triadens. Den används som modell för att utvärdera hur säkert ett system är.

- Confidentiality (Sekretess): Endast behöriga personer ska ha tillgång till information. Kryptering och behörighetsstyrning är vanliga skydd.

- Integrity (Integritet):  
Informationen ska vara korrekt och inte ha ändrats utan tillåtelse.  
Kontrollsummering, versionshantering och logning används som skydd.
- Availability (Tillgänglighet):  
Informationen ska finnas tillgänglig när den behövs – även vid störningar. Det kan handla om backup, redundans, UPS eller skydd mot DDoS-attacker.



Alla tre delar behövs. En databas kan vara krypterad och korrekt, men om den ligger på en kraschad server utan backup är den fortfarande värdelös.

### 9.1.3 Säkerhet i praktiken

I praktiken innebär IT-säkerhet en kombination av teknik, beteende och rutiner.

- Tekniskt handlar det om att använda rätt program, system och inställningar.
- Beteendemässigt handlar det om att agera säkert – som att undvika att klicka på okända länkar, använda unika lösenord och logga ut från delade datorer.
- På organisationsnivå handlar det om att ha rutiner för uppdateringar, rättigheter, utbildning och återställning.

### 9.1.4 Säkerhetsnivåer – olika behov

Alla system behöver inte samma nivå av säkerhet.

- Ett hemdatorsystem kräver en viss grundläggande skyddsnivå.
- Ett system som hanterar personuppgifter, ekonomi eller känslig kommunikation kräver betydligt mer.

Exempel: En speldator i hemmet kanske inte behöver brandväggsloggning och multifaktorinloggning – men ett journalsystem på ett sjukhus kräver det som standard.

Säkerhet handlar alltid om balans mellan risk och nytta. För mycket säkerhet kan göra ett system svårt att använda – för lite gör det osäkert.

### 9.1.5 Tekniska skydd

De vanligaste tekniska skydden inom IT-säkerhet är:

- Antivirusprogram – skyddar mot känd skadlig kod.
- Brandväggar – kontrollerar vilken trafik som får passera.
- Kryptering – skyddar information från att läsas av obehöriga.
- Backup (säkerhetskopior) – gör att man kan återställa data vid krasch, virus eller misstag.
  - En säkerhetskopia bör alltid sparas på en annan fysisk plats eller i molnet – annars kan den gå förlorad vid brand, stöld eller krypteringsattack.
- Behörighetsstyrning – ser till att endast rätt användare får tillgång till rätt delar av ett system.

Tekniska skydd är grundläggande – men inte tillräckliga utan mänskligt ansvar.

### 9.1.6 Mänskliga faktorn

Många säkerhetsproblem uppstår inte på grund av teknik, utan för att människor gör misstag.

Det kan handla om slarv, okunskap eller stress – eller att någon helt enkelt luras.

Exempel på vanliga misstag:

- Att klicka på en länk i ett bluffmejl (phishing)
- Att använda samma lösenord överallt
- Att lämna en dator obevakad med inloggat konto

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se första sidan

Därför är användarutbildning och medvetenhet minst lika viktig som brandväggar och virusskydd. Tekniken kan bara skydda så långt användaren tillåter det.

## 9.1.7 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-triadén står för och ge ett exempel.
8. Vad innebär "I" i CIA-triadén?
9. Hur skyddar man tillgängligheten i ett IT-system?
10. Hur kan man uppnå balans mellan sekretess, integritet och tillgänglighet?

## 9.1.8 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/>

## 9.1.9 Egna anteckningar

(Lämna plats för elevernas egna reflektioner och noteringar här)

## 9.2 Skydd på olika nivåer

Fysisk säkerhet är en grundläggande men ofta förbiseedd 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. Det räcker alltså inte med tekniska skydd – om hårdvaran stjäls eller förstörs är informationen i fara.

### 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")

Skadliga aktörer kan använda automatiserade attacker (brute force) som testar miljontals lösenord per sekund. Ett starkt lösenord kan stå emot detta i åratals – ett svagt i bara några sekunder.

Ett vanligt sätt att förstå lösenordsstyrka är att räkna antalet möjliga kombinationer:

Teckenuppsättning	Antal möjliga tecken	Komb. vid 6 tecken
Endast siffror (0-9)	10	$10^6 = 1 \text{ miljon}$
Små bokstäver (a-z)	26	$26^6 \approx 308 \text{ miljoner}$
Små + stora bokstäver	52	$52^6 \approx 19 \text{ miljarder}$

Teckenuppsättning	Antal möjliga tecken	Komb. vid 6 tecken
Bokstäver + siffror + specialtecken	ca 96	$96^6 \approx 782$ miljarder

Och med 8 tecken?

- $96^8 \approx 7\ 213\ 895\ 789\ 838\ 336$  kombinationer – över 7 biljarder.

Detta är varför ett lösenord på bara 6 tecken kan knäckas på sekunder, medan ett starkt lösenord på 12–14 tecken kan ta flera tusen år att knäcka, även med kraftfull hårdvara.

 Tips: Ett säkert lösenord bör vara minst 12 tecken långt – gärna en lösenordsfras (t.ex. KorvMedBröd!2025) som är både stark och lätt att minnas. Undvik vanliga ord, återanvänd inte lösenord, och byt vid misstänkt läcka.

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

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

De tre autentiseringssfaktorerna:

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 unikt för dig	Fingeravtryck, ansikte, iris

För att räknas som MFA måste minst två olika typer av faktorer användas. Två lösenord är alltså inte MFA – det är bara två saker du vet.

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 + autentiseringssapp	Vad du vet	Vad du har	✓
Fingeravtryck + ansiktsigenkänning	Vem du är	Vem du är	✗

MFA skyddar konton även om lösenordet läcker – angriparen behöver också din mobil, din nyckel eller ditt finger.

## 9.2.2 Programvarusäkerhet – Antivirus och uppdateringar

Ett vanligt misstag är att tro att ett installerat antivirus automatiskt gör datorn "osårbar". Så är det inte.

Antivirus fungerar som ett digitalt vaktsystem – det letar efter och blockerar kända hot, men är bara så bra som dess senaste uppdatering. Ett antivirus som inte uppdateras ger bara en falsk trygghet.

Det är också viktigt att själva antivirusprogrammet uppdateras – även dessa kan innehålla sårbarheter.

🛡️ Exempel på vanliga antivirusprogram:

1. Bitdefender – Hög upptäcktsgrad, låg systempåverkan.
2. Avast – Gratis, men har kritiserats för datainsamling (se nedan).
3. Kaspersky – Tekniskt avancerat, används globalt.

4. ESET NOD32 – Snabbt och resurssnålt.
5. Windows Defender – Inbyggt i Windows 10/11 – numera ett starkt alternativ.

⚠️ Integritetsrisker:

Alla antivirusprogram samlar inte bara in virus – vissa samlar även in användardata. Ett exempel är Avast, som samlade in surfhistorik och andra uppgifter via dotterbolaget Jumpshot. Det visar att även säkerhetsprogram kan utgöra en risk.

💡 Tips: Läs alltid användarvillkor och sekretesspolicy. Fundera på hur företaget tjänar pengar – särskilt om programmet är ”gratis”.

### 9.2.3 Nätverkssäkerhet

Nätverkssäkerhet handlar om att skydda all data som rör sig mellan enheter – både i hemnätverk, företagsnätverk och över internet.

#### *Brandväggar – nätverkets väktare*

En brandvägg kontrollerar vilken trafik som får passera. Det finns två typer:

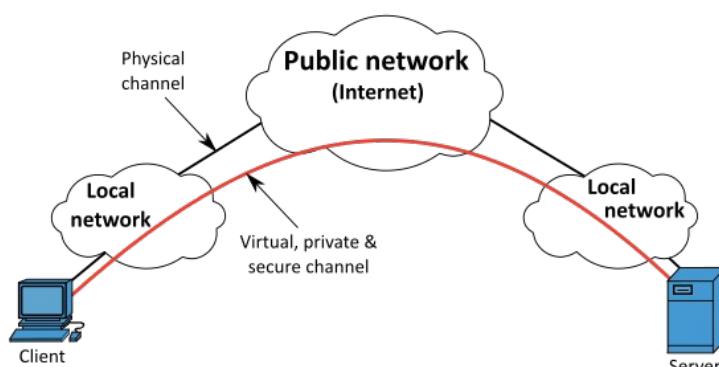
- Nätverksbrandväggar – skyddar hela nätverk (t.ex. i routrar eller OPNsense).
- Host-baserade brandväggar – skyddar enskilda datorer (t.ex. Windows Firewall).

Regler kan styra vilka portar, IP-adresser eller protokoll som är tillåtna. Ett vanligt misstag är att tillåta ”all trafik” – bekvämt men mycket farligt.

#### *VPN – Krypterad tunnel för säker anslutning*

Ett VPN (Virtual Private Network) krypterar trafiken mellan användare och nätverk. Det används för:

- Fjärrarbete
- Offentliga Wi-Fi-nätverk



- Sekretess (dölja IP-adress och plats)

Med VPN reser du i ett krypterat rör – ingen kan se in, inte ens din internetleverantör.

#### *IDS/IPS - Larmsystem i nätverket*

- IDS (Intrusion Detection System) övervakar nätwerkstrafik och larmar vid misstänkta händelser.
- IPS (Intrusion Prevention System) försöker aktivt stoppa attacken i realtid.

Används ofta i företag för att skydda mot t.ex. portscanning, brute-force-attacker eller exploit kits.

#### *Segmentering - Dela upp för att begränsa skada*

Att ha alla enheter i samma nätverk är riskabelt. Genom segmentering (t.ex. VLAN) kan man isolera trafik.

Exempel:

- Gästnätverk får endast tillgång till internet.
- IoT-enheter får inte prata med servrar.
- Administration isoleras från övriga användare.

Det bygger på principen om minsta möjliga åtkomst, och gör det mycket svårare för angripare att ta sig vidare i nätverket om de får in en fot

## 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](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 befina 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 1990-talet var WEP det första säkerhetsprotokollet. Namnet antydde att det skulle ge samma säkerhet som ett trådbundet nätverk. I praktiken var det dock allvarligt bristfällt.

Problem med WEP:

- Använde statisk krypteringsnyckel (samma nyckel hela tiden)
- Byggde på RC4 (Rivest Cipher 4) – en föråldrad och osäker krypteringsalgoritm
- Kunde knäckas på några minuter med fria verktyg (t.ex. aircrack-ng)

#### *WPA – Wi-Fi Protected Access (2003)*

WPA togs fram som en tillfällig lösning för att ersätta WEP innan en ny standard var helt färdig. Den förbättrade säkerheten genom dynamisk nyckelhantering och starkare kryptering.

- Använde TKIP (Temporal Key Integrity Protocol) – förbättrade nyckelrotation och integritetsskydd, men använde fortfarande RC4 i grunden.
- Byggde ofta på PSK (Pre-Shared Key), alltså ett gemensamt lösenord som skrivs in manuellt i router och klient.

#### *WPA2 – Wi-Fi Protected Access version 2 (2004)*

WPA2 ersatte WPA som fullständig standard. Den införde modern kryptering och blev snabbt branschstandard.

- Använde AES (Advanced Encryption Standard) – en robust blockkrypteringsalgoritm som även används inom statlig och militär kommunikation.
- Introducerade CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) – ett mer avancerat protokoll för att säkerställa både konfidentialitet och integritet.
- Kunde fortfarande använda WPA2-PSK, men med starkare nyckelhantering.

### *WPA3 – Wi-Fi Protected Access version 3 (2018)*

WPA3 förbättrar säkerheten ytterligare, framför allt för hemmabruk och öppna nätverk.

- Ersätter PSK med SAE (Simultaneous Authentication of Equals) – en metod som förhindrar offline brute-force-attacker.
- Har stöd för forward secrecy – vilket gör att äldre data inte kan dekrypteras i efterhand även om lösenordet läcker.
- Inkluderar Enhanced Open – förbättrad kryptering även i öppna nätverk (utan lösenord).
- Kräver nyare hårdvara och är inte alltid aktiverat som standard.

! OBS: Alla routrar har inte stöd för WPA3 – kontrollera inställningarna och uppdatera firmware vid behov.

### 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"

Skyddsåtgärd	Förklaring
Dölj eller byt SSID	Byt från standardnamn som "TP-Link_123" - det signalerar okunskap
MAC-filter (begränsad nyttja)	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
Uppdatera routerns firmware	Tillverkare släpper säkerhetsuppdateringar - håll mjukvaran aktuell

### 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.

"En Evil Twin-router kan även använda samma BSSID (MAC-adress) och starkare signal för att lura klienten att byta automatiskt."

#### **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. I ett nätverk utan WPA-skydd är all trafik sårbar.

"Även i krypterade nätverk kan viss information läcka – exempelvis vilka domäner som besöks, vilket kan användas för spårning."

### **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 Säkerhetsmedvetenhet**

Att använda rätt teknik är viktigt – men lika viktigt är att förstå riskerna. Många attacker bygger inte på avancerad teknik, utan på att användare är stressade, ouppmärksamma eller okunniga. Därför är säkerhetsmedvetenhet en avgörande del av trådlös säkerhet.

Informera användare om att:

- Undvika att ansluta till okända eller osäkra nätverk.
- Alltid dubbelkolla nätverksnamnet (SSID) innan anslutning – särskilt i offentliga miljöer.
- Logga ut från känsliga tjänster efter användning på publika nätverk.
- Använda VPN vid behov för att skydda sin trafik.

Eftersom trådlös trafik skickas genom luften räcker det med att vara inom räckhåll för att försöka avlyssna, störa eller lura användare. Genom en kombination av tekniska skydd (som WPA3, starka lösenord och segmentering) och medvetna användare skapas en betydligt säkrare miljö – både i skolan, på jobbet och hemma.

### 9.3.5 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.

### 9.3.6 Fördjupningslänkar

1. Wikipedia - WEP  
[https://en.wikipedia.org/wiki/Wired\\_Equivalent\\_Privacy](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/>

## 9.3.7 Egna anteckningar

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

## 9.4 Hot, risker och sårbarheter

Att prata om IT-säkerhet utan att förstå begreppen hot, risk och sårbarhet är som att prata om brandskydd utan att förstå eld. Dessa tre begrepp utgör grunden för nästan allt säkerhetsarbete – från att konfigurera brandväggar till att utbilda användare. I det här kapitlet går vi igenom vad orden betyder, hur de hänger ihop, och varför de är så centrala i både teori och praktik.

### 9.4.1 Vad är ett hot?

Ett hot är något som kan orsaka skada. Det är alltså en potentiell fara – något som, om det blir verklighet, påverkar systemet negativt. Hot kan vara både avsiktliga (t.ex. hackerangrepp) och oavsiktliga (t.ex. brand, elavbrott eller användarfel).

Exempel på hot:

- En obehörig användare försöker få tillgång till nätverket
- En anställd glömmer en okrypterad USB-sticka med känsliga filer
- En sårbarhet i ett operativsystem utnyttjas av skadlig kod

Hot är alltså inte skadan i sig – utan möjligheten till skada.

### 9.4.2 Vad är en sårbarhet?

En sårbarhet är en svaghet i ett system som kan utnyttjas av ett hot. Det kan vara tekniskt (t.ex. dålig kryptering), organisatoriskt (t.ex. brist på rutiner) eller mänskligt (t.ex. dålig lösenordshantering).

Exempel på sårbarheter:

- Öppna portar utan brandväggsregler
- Föråldrad programvara med kända buggar
- Användare som klickar på allt de får i inkorgen

Ju fler sårbarheter ett system har, desto större är chansen att hot blir verklighet.

### 9.4.3 Vad är en risk?

En risk är sannolikheten att ett hot utnyttjar en sårbarhet – och den potentiella konsekvensen av det. Risken är alltså en kombination av hur troligt det är att något händer, och hur illa det blir om det gör det.

Formel (förenklad):

$$\text{Risk} = \text{Hot} \times \text{Sårbarhet} \times \text{Konsekvens}$$

Exempel:

Om ett företag har en dåligt skyddad server (sårbarhet), och det finns aktiva attacker mot just den typen av server (hot), så är risken hög – särskilt om servern innehåller känslig data (stor konsekvens).

Riskanalys handlar om att identifiera dessa kombinationer och prioritera vad som måste åtgärdas först.



#### 9.4.4 Diskussions- och kontrollfrågor

1. Vad är skillnaden mellan ett hot och en sårbarhet?
2. Ge ett exempel på ett tekniskt hot och en organisatorisk sårbarhet.
3. Vad menas med att ett hot kan vara "oavsiktligt"?
4. Hur hänger hot, sårbarhet och risk ihop?
5. Varför är det viktigt att göra en riskanalys?
6. Hur kan mänskliga faktorer skapa sårbarheter i IT-system?
7. Beskriv en verklig situation där alla tre begreppen finns.
8. Vad är skillnaden mellan en låg och en hög risk?
9. Hur påverkas risken om sårbarheten tas bort?
10. Kan ett hot vara farligt även om det inte finns någon sårbarhet?  
Motivera.

## 9.4.5 Fördjupningslänkar

- MSB – Vad är risk och sårbarhet?  
<https://www.msb.se/sv/rad-och-kunskap/skydda-din-information/grundlaggande-sakerhet/>
- OWASP Top Ten (exempel på vanliga sårbarheter)  
<https://owasp.org/www-project-top-ten/>
- NIST Risk Management Framework  
<https://csrc.nist.gov/publications/detail/sp/800-37/rev-2/final>
- CERT-SE – Aktuella hot och sårbarheter  
<https://www.cert.se/>
- Wikipedia – Threat, Vulnerability and Risk  
[https://en.wikipedia.org/wiki/Threat\\_\(computer\)](https://en.wikipedia.org/wiki/Threat_(computer))

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

## 9.4.6 Egna anteckningar

## 9.5 Säkerhet i Windowsmiljö

### 9.5.1 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.

👉 **Tillägg:**

**TPM** (Trusted Platform Module) används för att lagra krypteringsnycklar på ett säkert sätt, t.ex. för BitLocker. Secure Boot kontrollerar att endast godkänd kod startas vid uppstart – vilket minskar risken för rootkits och boot-attacker.

**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.

## 9.5.2 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.

UAC är inte en funktion som appar själva styr – utan det är Windows som avgör när en förhöjd behörighet krävs. Det är alltså inte ett skydd i själva programmet, utan i operativsystemets kontrollmekanism.

### 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

Det går att helt stänga av UAC – men det rekommenderas inte, då det tar bort ett viktigt skydd mot skadlig kod.

## @@ 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.

### 9.5.3 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. Funktionen “Controlled Folder Access” är ett extra lager skydd mot ransomware. Den tillåter endast betrodda appar att ändra filer i utvalda mappar, vilket minimerar risken för att skadlig kod krypterar din data.
- **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 **försiktig surfande, starka lösenord och regelbundna uppdateringar**.

## 9.5.4 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. Med Group Policy kan man också centralt styra lösenordspolicy (t.ex. längd och komplexitet), blockera specifika program, aktivera brandväggsregler, begränsa åtkomst till USB-portar, eller automatiskt distribuera inställningar och program – allt utan att behöva gå till varje dator.

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.

## **Grppolicy Editor (gpedit.msc)**

Verktyget används lokalt på datorer för att visa och redigera policies. 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.

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

**NTFS (New Technology File System)** är det filsystemet som används i moderna Windows-versioner, och det innehåller stöd för avancerad **behörighetskонтroll**. 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?
<b>Läsa (Read)</b>	Kan visa innehåll, men inte ändra
<b>Skriva (Write)</b>	Kan skapa och ändra filer
<b>Ändra (Modify)</b>	Kan läsa, skriva, ändra och ta bort
<b>Fullständig behörighet</b>	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.
- Behörigheter kan sättas via fliken "Säkerhet" i Utforskaren, eller med kommandoradsverktyget icacls - vilket ger mer kontroll och används ofta i script eller GPO.

## 9.5.6 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.

BitLocker finns endast i Windows 10/11 Pro, Enterprise eller Education. I Home-versionen kan du använda tredjepartsverktyg – men de erbjuder inte alltid samma integration och säkerhetsnivå.

## 9.5.7 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
<b>Inkommande regler</b>	Bestämmer vad som får <i>komma in</i> till datorn från nätverket/internet
<b>Utgående regler</b>	Bestämmer vad som får <i>lämna</i> 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:**

<b>Nätverkstyp</b>	<b>Beskrivning</b>
<b>Privat</b>	Hemmanätverk – du litar på andra enheter
<b>Offentligt</b>	T.ex. kafé eller flygplats – mer restriktiv
<b>Domän</b>	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!
- Vissa tredjeparts-brandväggar (t.ex. från antivirusprogram) kan automatiskt stänga av Windows Firewall – detta bör alltid kontrolleras, särskilt om du byter säkerhetslösning.

## 9.5.8 Autentisering: Kerberos och NTLM

När du loggar in på en Windows-dator i ett nätverk (särskilt i en domän), används ett autentiseringssprotokoll 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 autentiseringssprotokoll.

- 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 domänskontrollant.

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). NTLM används också för lokal inloggning, i arbetsgruppsmiljöer, eller om du använder äldre

programvaror som inte stöder Kerberos. Därför är det viktigt att förstå dess risker, även om du arbetar i moderna miljöer.

### Tips:

- I företagssammanhang bör **Kerberos alltid användas som standard**
- NTLM bör **avaktiveras** i moderna miljöer om det är möjligt

## 9.5.9 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.

⚠️ Visste du? Microsofts säkerhetsrapporter visar att över 90 % av dagens cyberattacker börjar med phishing – alltså försök att lura användare via e-post, SMS eller fejkade webbsidor.

### Vad är malware?

Malware är ett samlingsnamn för olika typer av skadlig programvara:

Typ	Förklaring
<b>Virus</b>	Kopierar sig själv och infekterar andra filer
<b>Mask (worm)</b>	Sprider sig automatiskt via nätverk
<b>Trojan</b>	Låtsas vara något användbart, men innehåller skadlig kod
<b>Spyware</b>	Samlar in information i smyg
<b>Ransomware</b>	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 var 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!

## 9.5.10 Group Policy i domänmiljöer

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. I avsnitt 9.4.4 introducerades Group Policy. Här tittar vi närmare på hur det fungerar i domänmiljöer, där inställningar styrs 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 administrörsrättigheter
- USB-portar är avstängda på elevdatorer

### 9.5.11 PowerShell – kraftfullt men farligt

PowerShell är ett av de mest kraftfulla verktygen i Windows – och därför också ett av de mest riskfyllda om det missbrukas. En angripare som får tillgång till PowerShell kan göra nästan vad som helst: skapa användare, stänga av säkerhetssystem, ladda ner skadlig kod direkt i minnet utan att skriva till disk.

Varför är det ett säkerhetsproblem?

PowerShell är installerat som standard och körs ofta med höga rättigheter. Många attacker använder PowerShell för att undvika antivirus – eftersom skadlig kod kan köras direkt i kommandoraden utan att skapa misstänkta filer.

## Skyddsåtgärder:

- Execution Policy: Begränsa vilka skript som får köras. Exempel:
  - Restricted – inga skript får köras (standard för nya installationer)
  - RemoteSigned – lokala skript körs, nedladdade kräver signatur
  - AllSigned – alla skript måste vara digitalt signerade
- Group Policy-inställningar: Går att sätta centralt i domänmiljö.
- Loggning: Aktivera PowerShell Script Block Logging och Transcription för att se vad som körs.

Tips: Kombinera restriktioner med övervakning – det är inte ovanligt att angripare kör PowerShell-kommandon via t.ex. makron i Office-dokument.

### 9.5.12 AppLocker – styr vilka program som får köras

AppLocker är ett verktyg i Windows (tillgängligt i Pro/Enterprise/Education) som låter administratörer skapa regler för vilka program, script och installerare som får köras. Det fungerar som ett smart filter som stoppar obehörig kod – även om användaren har tillgång till själva filen.

Exempel på användning:

- Tillåt bara program signerade av Microsoft eller skolan
- Blockera .bat, .ps1 eller .exe från okända platser
- Låt bara appar från Program Files eller en särskild mapp köras

Fördelar:

- Kan stoppa ransomware och malware innan de körs
- Bra komplement till antivirus
- Styrs via Group Policy

Tips: Börja i Audit Mode (granskningssläge) för att se vilka regler som skulle blockera program – utan att de faktiskt blockeras. På så sätt kan du testa utan att riskera driftstopp.

### 9.5.13 Event Viewer – gräva i loggarna

Event Viewer (Loggboken) är ett inbyggt verktyg i Windows där du kan granska systemets loggar. Det är ovärderligt vid felsökning, säkerhetsgranskning och incidenthantering.

Viktiga loggar att känna till:

Loggtyp	Innehåll
System	Start/stopp av tjänster, drivrutinsfel
Applikation	Fel i program (t.ex. Word kraschar)
Security	Inloggningar, UAC-försök, behörighetsändringar
Windows Defender	Hot som upptäckts, realtidsskydd
PowerShell	Kommandon som körts (om Script Block Logging är på)

Användningsområden:

- Se om någon försökt logga in utan rätt lösenord
- Spåra när ett USB-minne kopplades in
- Identifiera när ett misstänkt program kördes
- Knyta händelser till användare, IP-adresser eller tidpunkter

Tips:

- Använd filtrering (t.ex. bara Visa Event ID 4625 = misslyckade inloggningar)
- Exportera loggar för granskning eller arkivering

- Kombinera med SIEM-system (t.ex. Wazuh, Splunk) för central analys

### 9.5.14 Användarkonton – olika typer och deras rättigheter

Windows har flera olika kontotypar som avgör vad en användare får – och inte får – göra i systemet. En tydlig förståelse för dessa är viktig både för säkerhet och administration.

Kontotyp	Rättigheter	Vanligt användningsområde
Administratör	Full kontroll över systemet	IT-tekniker, systemadministratörer
Standardkonto	Kan använda program, men ej installera nya	Vanliga användare, t.ex. elever
Gästkonto	Mycket begränsad åtkomst	Tillfälliga användare, testinloggning
Microsoft-konto	Onlinekonto kopplat till Microsoft-tjänster	Synkar inställningar, OneDrive, Office



#### Tips för säkerhet:

- Använd standardkonto för dagligt arbete – även om du är admin.
- Gästkonto är oftast avstängt i moderna versioner – håll det så om det inte behövs.
- Undvik att dela Microsoft-konton mellan flera användare – det kan leda till synkningsproblem och integritetsrisker.

"Kom ihåg: Säkerheten i Windows består inte av ett enda verktyg – utan av ett helt ekosystem. Från UAC till BitLocker, från Defender till Group Policy – varje komponent spelar sin roll."  
(Men bara om du vill ge kapitlet en mjuk landning.)

## 9.5.15 Kontrollfrågor – 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.)?

## 9.5.16 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>

## 9.5.17 Egna anteckningar

(Plats för elevens egna reflektioner, minnesregler eller fördjupningar.)

## 9.6 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 gissa lösenord, utan om att utnyttja dåliga inställningar – till exempel öppna portar, gamla programversioner eller bortglömda användarkonton. Det är därför Linux-säkerhet ofta handlar mer om hur du konfigurerar än vad du installerar.

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.

### 9.6.1 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. Detta gäller även tjänster – det är bättre att en webbserver kör som användaren www-data än som root, eftersom skadorna vid en kompromettering då blir begränsade.

#### **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.

#### **Exempel på lager:**

- Brandvägg (t.ex. ufw)
- Logganalysverktyg (t.ex. logwatch)
- Automatisk blockering av inloggningsförsök (t.ex. fail2ban)
- Diskkryptering (t.ex. LUKS)
- Systemgranskning (t.ex. lynis)

Inget enskilt skydd räcker – men tillsammans gör de systemet mycket säkrare.

### *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 längre används
- 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  
(lista användare med cut -d: -f1 /etc/passwd, inaktivera med sudo usermod --expiredate 1 användarnamn)
- Stäng onödig trafik med ufw:

sudo ufw status

sudo ufw deny ftp

### *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
- Tjänster kör med för höga rättigheter

Att härda systemet är en viktig del av driften – inte ett engångsjobb.

Tips: Installera lynis och kör sudo lynis audit system för en översiktlig säkerhetsgranskning.

### *Säkerhet är en process – inte en produkt*

Det går inte att installera ett "säkerhetsprogram" och sen luta sig tillbaka. Säkerheten måste underhållas och ses som en levande del av drift och administration.

#### **Åtgärder som bör vara rutin:**

- Patcha regelbundet – håll system och paket uppdaterade
- Bevaka sårbarheter (CVE) – t.ex. via ubuntu-security-notice eller <https://cve.mitre.org>
- Automatisera säkerhetsuppdateringar med unattended-upgrades eller apticron
- Testa miljön regelbundet med verktyg som lynis, rkhunter eller chkrootkit

## 9.6.2 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?

### 9.6.3 Fördjupningslänkar

- NIST – Least Privilege  
[https://csrc.nist.gov/glossary/term/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>

## 9.6.4 Egna anteckningar

(Lämna plats för egna reflektioner och fördjupningar här.)

## 9.7 Linux OS-härdning (Operating System Hardening)

Att "härdta" 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 – ett kommandoradsverktyg som skannar systemet och föreslår förbättringar baserat på säkerhetsprinciper
- CIS-CAT – ett Java-baserat verktyg från CIS som kontrollerar om systemet följer benchmarkrekommendationerna (compliance)

Tips: Lynis är särskilt användbart i undervisning eftersom det är lätt att köra och ger pedagogiska resultat.

### *AppArmor & SELinux – kontrollera vad processer får göra*

AppArmor (Application Armor) och SELinux (Security-Enhanced Linux) är så kallade Mandatory Access Control-system (MAC). De fungerar som extraskydd ovanpå vanliga filrättigheter (t.ex. chmod).

AppArmor:

- Lättare att förstå och administrera

- Har färdiga profiler för vanliga program (t.ex. apache2, mysql, ping)
- Status kollas med sudo aa-status

### **SELinux:**

- Mer avancerat men svårare att felsöka
- Vanligt i enterprismiljöer (t.ex. Red Hat, CentOS)
- Status: sestatus

Tips: I Ubuntu- och Debianmiljöer rekommenderas AppArmor för att det är mer användarvänligt och räcker gott för de flesta skolmiljöer.

### *Secure Boot - skydda redan vid uppstart*

Secure Boot är ett säkerhetsläge i UEFI (Unified Extensible Firmware Interface) som ser till att endast signerade kärnor och program laddas vid uppstart. Det förhindrar att angripare injicerar skadlig kod (t.ex. rootkits) innan operativsystemet ens har startat.

I Linux:

- Måste vara aktiverat i BIOS/UEFI
- Kräver ofta att kärnmoduler är signerade digitalt
- Vid installation av proprietära drivrutiner kan användaren behöva "enrolla" en nyckel via MOK (Machine Owner Key)

Tips: Secure Boot fungerar väl med Ubuntu LTS-versioner, men kan kräva manuell bekräftelse vid installation av tredjepartsdrivrutiner.

### *Kryptera diskar och partitioner*

Fysisk säkerhet är inte alltid nog – särskilt inte för bärbara enheter. Kryptering gör att datan blir oläsbar utan rätt nyckel, även om någon plockar ur hårddisken och försöker läsa den på en annan dator.

Vanliga alternativ:

- LUKS (Linux Unified Key Setup) – standard för full diskryptering i t.ex. Ubuntu. Aktiveras ofta under installation med alternativet "Kryptera hela disken". Används tillsammans med verktyget cryptsetup.

- eCryptfs – tidigare populärt för att kryptera hemmakataloger, men är idag mindre vanligt.
- ZFS Native Encryption – används i ZFS-baserade system och ger flexibel kryptering per dataset.

Tips: På bärbara datorer bör full diskryptering alltid användas. På stationära system räcker det ofta att kryptera viktiga kataloger som /home.

### *Partitionering som säkerhetsåtgärd*

Att dela upp disken i flera partitioner gör att olika delar av systemet är isolerade från varandra. Det minskar risken för att ett misstag – som att loggfiler fyller disken – påverkar hela systemet.

Rekommenderade separata partitioner:

- /home – användardata
- /var – loggar, cache, e-post, databaser m.m.
- /tmp – tillfälliga filer
- /boot – uppstartsdata (i vissa fall)

### **Vanliga mount-alternativ för ökad säkerhet:**

- noexec – tillåter inte körbara filer (bra för t.ex. /tmp)
- nosuid – förhindrar att SUID-program körs med höjda rättigheter
- nodev – tillåter inte skapande av enhetsfiler

### **Tips:**

Vill du snabbt se vilka mount-alternativ som används?

Kör:

mount | grep "^\/"

eller

findmnt

## 9.7.1 Kontrollfrågor

1. Vad menas med att "härdta" 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?

## 9.7.2 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](https://selinuxproject.org/page/Main_Page)
- Ubuntu LUKS Encryption –  
[https://help.ubuntu.com/community/Full\\_Disk\\_Encryption\\_Howto\\_2019](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](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/ch-mounting-file-systems)

### 9.7.3 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.)*

## 9.8 Linux 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 handlar alltså inte om avancerad "hacker-magi", utan snarare om att administratören inte har hunnit, orkat eller vetat att uppdatera i tid.

### *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 täppas till)

I ett större nätverk kan en sårbar server bli en inkörsport för angripare – även om det är en gammal testserver som glömts bort. Genom att hålla systemet uppdaterat stänger man dessa portar innan de ens hinner öppnas av en angripare.

### *Verktyg i Debian-baserade system*

I Debian och Ubuntu använder man främst följande kommandon för uppdatering:

Kommando	Förklaring
sudo apt update	Uppdaterar listan över tillgängliga paket från repositories
sudo apt upgrade	Installerar tillgängliga uppdateringar för installerade paket
sudo apt full-upgrade	Uppgraderar och tar även bort paket vid behov (beroenden)
apt list --upgradable	Visar vilka paket som har uppdateringar

Kommando	Förklaring
tillgängliga	

Man kan även använda apt-get, men apt är numera standard för interaktivt bruk.

Tips: Vill du endast uppgradera ett enskilt paket (t.ex. openssh-server), kan du använda:

`sudo apt install --only-upgrade openssh-server`

#### *Automatisk uppdatering med unattended-upgrades*

På servrar där man inte loggar in så ofta kan det vara bra att automatisera vissa uppdateringar – särskilt säkerhetspatchar.

Installation:

`sudo apt install unattended-upgrades`

Aktivera:

`sudo dpkg-reconfigure --priority=low unattended-upgrades`

Detta sätter upp automatisk installation av säkerhetsuppdateringar. Vill du justera vad som uppdateras, kan du redigera:

`/etc/apt/apt.conf.d/50unattended-upgrades`

Tips: Det går även att konfigurera loggning eller notifiering via e-post, så att du får veta när något uppdaterats.

#### *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 – perfekt för felsökning.
- Följ med i säkerhetsflöden för din distribution – t.ex. [Ubuntu Security Notices](#).
- Undvik att köra uppdateringar på produktion utan att först läsa vilka paket som påverkas.

- Kom ihåg: vissa uppdateringar (som kärnan) kräver omstart. Det kan kontrolleras med:

```
[ -f /var/run/reboot-required ] && echo "Omstart krävs"
```

Patchhantering handlar om att vara proaktiv – att stänga säkerhetshål innan någon försöker klättra in genom dem.

## 9.8.1 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?

## 9.8.2 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/>

### 9.8.3 Egna anteckningar

(Lämna plats för egna reflektioner och anteckningar här.)

## 9.9 Linux 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 annan systemnära information. Journald lagrar loggar binärt, vilket gör dem mer motståndskraftiga mot manipulation jämfört med vanliga textfiler.

 Vanliga kommandon:

bash

CopyEdit

```
journalctl          # Visa alla loggar
journalctl -b       # Loggar från senaste boot
journalctl -u ssh    # Endast SSH-tjänstens loggar
journalctl --since "1 hour ago" # Loggar från senaste timmen
```

Tips:

- Använd less för att bläddra bekvämt i längre loggar.
- Filtrera med grep för att söka efter specifika rader:

```
journalctl -u ssh | grep "Failed"
```

Vill du exportera loggar till text?

```
journalctl -u ssh > ssh-logg.txt
```

## *Rsyslog - klassisk textbaserad logghantering*

Utöver journald används ofta rsyslog för att skicka loggar till olika textfiler, vanligtvis under /var/log. Systemet är kraftfullt och används ofta för att:

- Centralisera loggar från flera servrar
- Skicka loggar till externa logghanteringssystem
- Definiera egna regler för filtrering och lagring

 Exempel på vanliga loggfiler:

Fil	Innehåll
/var/log/auth.log	Inloggningar, sudo, autentisering
/var/log/syslog	Allmän systemlogg
/var/log/kern.log	Kärnrelaterade meddelanden
/var/log/apt/history.log	Installationshistorik via apt
/var/log/faillog	Misslyckade inloggningsförsök

Konfigurationen sker i:

- /etc/rsyslog.conf (huvudfil)
- /etc/rsyslog.d/ (egna regler)

För att ladda om rsyslog efter ändring:

`sudo systemctl restart rsyslog`

## *AIDE - Advanced Intrusion Detection Environment*

AIDE är ett verktyg för att kontrollera om något i filsystemet har förändrats – till exempel om en systemfil modifierats utan att du vet om det. Det fungerar som ett ”före-efter”-system: först skapas en säkerhetskopia av systemets skick, och därefter kontrolleras det regelbundet.

## Så här kommer du igång:

```
sudo apt install aide  
sudo aideinit          # Skapar ursprungs databasen  
sudo aide.wrapper --check    # Jämför aktuell status med databasen
```

### Viktigt:

Den ursprungliga databasen (/var/lib/aide/aide.db.new) bör flyttas till en säker plats – helst offline – efter första körningen. Om en angripare når databasen kan den manipuleras för att dölja spår.

### Automatisering:

Schemalägg AIDE så att det körs varje natt med cron eller systemd timers.

Exempel med cron:

```
0 2 * * * /usr/bin/aide.wrapper --check
```

### *Varför är detta viktigt?*

- Loggar ger spårbarhet – du vet vem som gjort vad, och när.
- De används i incidentutredningar vid intrång eller misstag.
- Ett förändrat system utan din vetskaps är ofta tecken på en attack.
- Automatisk övervakning gör det möjligt att reagera snabbt – innan skadan är skedd.

### 9.9.1 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 AIDE:s databas på ett säkert ställe?
9. Hur kan AIDE automatiseras för daglig kontroll?
10. Hur kan loggar användas vid säkerhetsincidenter?

## 9.9.2 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](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/>

### 9.9.3 Egna anteckningar

(Lämna plats för egna reflektioner och kompletterande information här.)

## 9.10 Linux root, sudo och användarbehörighet

I Linux är säkerheten starkt kopplad till hur användarbehörigheter hanteras. Den mest kraftfulla användaren är root, som har obegränsad tillgång till hela systemet. Att förstå hur root, sudo och behörigheter fungerar är en grundförutsättning för att säkra ett Linuxsystem.

### *Root - superanvändaren*

root är den administrativa användaren i Linux – ofta kallad superuser. Den har fullständiga rättigheter till alla filer, kataloger, processer och kommandon. Inloggning som root kan vara farligt, eftersom:

- Alla kommandon körs utan begränsning – ett felsteg kan krascha systemet.
- Root-inloggning loggas ofta inte separat, vilket försvarar spårbarhet.
- Om en angripare får root-access är systemet i praktiken helt övertaget.

Därför rekommenderas att man använder sudo istället – ett verktyg som tillfälligt ger root-rättigheter till vanliga användare.

### *sudo - tillfällig superkraft*

Med sudo kan en vanlig användare köra specifika kommandon med administratörsrättigheter. Exempel:

`sudo apt update`

Vid första användningen (eller efter en timeout) krävs lösenord, vilket loggas i systemet. Fördelarna med sudo:

- Du vet vem som körde vad – loggar skapas i /var/log/auth.log
- Rättigheterna är tillfälliga – du är inte permanent superuser
- Det går att begränsa användares rättigheter till specifika kommandon

Konfigurationen sker i /etc/sudoers, och bör alltid redigeras med:

`sudo visudo`

 Exempel på begränsad sudo-access:

`student ALL=(ALL) NOPASSWD: /usr/sbin/service apache2 restart`

Här tillåts användaren student att starta om Apache utan att ange lösenord - men inget annat.

### *Användar- och gruppering*

Varje användare i Linux har ett UID (User ID), och varje grupp har ett GID (Group ID). Alla filer och kataloger har tre behörighetsnivåer:

Symbol	Betydelse	Vad det ger rätt till
r	read	Läsa filer eller lista kataloger
w	write	Skriva till filer eller ändra innehåll
x	execute	Köra filer eller gå in i kataloger

 Exempel:

`-rwxr-xr-- 1 robin admin 4200 jul 18 10:00 script.sh`

- rwx - användaren robin kan läsa, skriva, köra
- r-x - gruppen admin kan läsa och köra
- r-- - andra användare får endast läsa

### *Byta ägare och rättigheter*

 Byta ägare och grupp:

`sudo chown robin:admin fil.txt`

 Ändra behörigheter (numeriskt eller symboliskt):

chmod 755 script.sh

chmod u+x fil.sh # Lägg till körbarhet för ägare

 Lägg till en användare i en grupp:

sudo usermod -aG sudo student

-aG står för "append to Group" – viktigt för att inte skriva över befintliga grupper.

#### *Vanliga misstag och fallgropar*

- Undvik att ge alla användare full sudo-access utan eftertanke.
- Använd NOPASSWD endast i särskilda, säkra sammanhang.
- Håll koll på vem som kan göra vad – använd loggarna i /var/log/auth.log.
- Logga aldrig in som root om du inte måste – särskilt inte via SSH.

 Tips: På många moderna system (t.ex. Ubuntu) är root-kontot inaktiverat som standard. Administratörer använder istället ett vanligt konto med sudo-rättigheter.

## 9.10.1 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?

## 9.10.2 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>

## 9.10.3 Egna anteckningar

(*Lämna plats för egna reflektioner, exempel och kommandon*)

## 9.11 Linux loggning och övervakning

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 visar senaste händelser med extra information
- Loggar i binärt format – inte direkt läsbara med cat
- Vill du bara visa kernel-relaterade loggar: journalctl -k

rsyslog (traditionellt loggsystem)

Klassiskt loggsystem som skriver till textfiler i /var/log/, t.ex.:

- /var/log/auth.log – inloggningar, sudo-försök
- /var/log/syslog – allmänna systemmeddelanden
- /var/log/kern.log – kernel-meddelanden
- /var/log/dmesg – hårdvarumeddelanden från uppstart

 Tips: Misslyckade sudo-försök loggas i /var/log/auth.log – sök efter rader som innehåller authentication failure.

De flesta Ubuntu-system använder både rsyslog och journald parallellt. Det ger redundans och bättre filtreringsmöjligheter.

### *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

Ger information om minnesanvändning, I/O och nätverkstrafik.

 vmstat = minne & processer, iostat = diskaktivitet, netstat = nätverksanslutningar (ersätts av ss)

#### *fail2ban - skydd mot brute-force*

Fail2ban övervakar loggar (t.ex. /var/log/auth.log) och blockerar IP-adresser som försöker logga in upprepade gånger utan att lyckas. Det är särskilt effektivt för att skydda SSH.

Installera:

sudo apt install fail2ban

Du kan anpassa reglerna i /etc/fail2ban/jail.conf eller skapa en egen jail.local för egna inställningar.

#### *AIDE - Advanced Intrusion Detection Environment*

AIDE fungerar som ett "före-efter"-verktyg för att bevaka systemets tillstånd. Det håller koll på filändringar, rättigheter och misstänkta modifieringar.

Installera:

sudo apt install aide

Initiera databas:

sudo aideinit

Verifiera ändringar:

sudo aide --check

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se första sidan

 **Viktigt:** Förvara AIDE:s ursprungsdatasäkerhet på en separat, skyddad plats så att en angripare inte kan manipulera både filsystem och kontrollfil.

### 9.11.1 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. Nämnn 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"?

## 9.11.2 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>

### 9.11.3 Egna anteckningar

*(Här kan du skriva ner egna exempel, kommandon du testat eller viktiga lärdomar om loggning och övervakning.)*

## 9.12 Linux 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 automatiserade attacker.

 Rekommenderade inställningar för säkrare SSH:

- Byt standardport från 22 till något annat, t.ex. 2222  
(OBS: detta stoppar inte en attack, men minskar mängden skriptade försök)

```
sudo nano /etc/ssh/sshd_config
```

```
# Port 2222
```

- Stäng av inloggning som root  
Root bör aldrig kunna logga in via SSH – använd istället sudo.

```
PermitRootLogin no
```

- Tillåt endast nyckelbaserad autentisering  
Kräver att användaren har en privat SSH-nyckel i sin klient.

```
PasswordAuthentication no
```

- Begränsa vilka användare som får logga in via SSH:

## AllowUsers robin adminuser

- Starta om SSH för att ladda in ändringar:

```
sudo systemctl restart ssh
```

 Tips: Använd fail2ban för att blockera IP-adresser som försöker logga in uppreatade gånger med fel lösenord. Det skyddar särskilt bra mot brute-force-attacker.

## *Brandvägg – UFW (Uncomplicated Firewall)*

En brandvägg kontrollerar vilken trafik som får passera in eller ut från datorn. ufw är ett användarvänligt gränssnitt till den kraftfulla brandväggen iptables.

 Så kommer du igång med UFW:

```
sudo apt install ufw
```

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```

```
sudo ufw allow 22/tcp # Tillåt SSH (eller 2222 om du bytt port)
```

```
sudo ufw enable
```

 OBS: Lägg alltid till regler för fjärråtkomst innan du aktiverar ufw, annars riskerar du att låsa ute dig själv!

 Visa status:

```
sudo ufw status verbose
```

## *TLS/SSL – Krypterad kommunikation*

TLS (Transport Layer Security) är efterföljaren till SSL (Secure Sockets Layer), och används för att skydda data under överföring. Du har sett det i praktiken när du besöker en webbplats med https://.

 Installera Let's Encrypt-certifikat med Certbot:

```
sudo apt install certbot python3-certbot-apache
```

```
sudo certbot --apache
```

- För Nginx: byt --apache mot --nginx.

⌚ Varför TLS/SSL?

- Skyddar all kommunikation från avlyssning
- Bekräftar webbplatsens identitet
- Ger åtkomst till HTTPS – ett krav i moderna webbläsare

⌚ Automatisk förnyelse av certifikat:

Certbot installerar som standard en systemd-timer:

```
sudo systemctl status certbot.timer
```

Om det fungerar visas en aktiv timer som kör certbot renew regelbundet.

## 9.12.1 Kontrollfrågor

1. Vad innebär det att "härdar" 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?

## 9.12.2 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](https://owasp.org/www-project-cheat-sheets/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html)

### 9.12.3 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.)*

## 9.13 Linux 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 med stöd för schemaläggning, inkrementella backuper, lagring till moln och loggning

### *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 sparar, hur de återställs och vem som ansvarar för dem.

### *Offsite & molnbackup*

Om backuper endast sparas lokalt riskerar de att förstöras i brand, stöld eller översvämnning.

Lösning:

- Synka till en annan fysisk plats (t.ex. annan server)
- Använd molnlagring med verktyg som rclone (Google Drive, S3, etc.)
- Kryptera känslig data innan molnuppladdning (ex. med gpg)

### *Vanliga misstag*

- Backupen sparar 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 backup + offsite-lösning + återställningstest

## 9.13.1 Kontrollfrågor

1. Vad är skillnaden mellan full och inkrementell backup?
2. Vilket verktyg i Linux används för effektiv filsynchronisering?
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?

## 9.13.2 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/>

### 9.13.3 Egna anteckningar

*(Här kan du skriva hur du konfigurerat dina egna backupscript, vilka kataloger du prioriterar, eller testa att återskapa ett gammalt tar-arkiv.)*

## 9.14 Linux 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. För systemomfattande jobb används /etc/crontab.

### *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
```

Visa logg:

```
cat /var/log/fail2ban.log
```

fail2ban övervakar bl.a. /var/log/auth.log för att upptäcka 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

## 9.14.1 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. Nämna tre säkerhetsuppgifter du kan schemalägga med cron.

## 9.14.2 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](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/>

### 9.14.3 Egna anteckningar

*(Här kan du skriva vilka säkerhetssuppgifter du har automatiserat, lägga in utdrag från din egen crontab, eller testa att skapa ett eget säkerhetsscript.)*

## 9.15 Linux Säkerhetsverktyg

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 via /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 kompromitterat. De har olika signaturbaser och kompletterar därför varandra.

```
sudo apt install chkrootkit rkhunter
```

```
sudo chkrootkit
```

```
sudo rkhunter --check
```

## 4. Lynis – Sårbarhetsanalys

Gör en säkerhetsgenomgång och listar potentiella risker i rapportformat. Används ofta vid säkerhetsrevisioner och hårdvarugranskningar.

```
sudo apt install lynis
```

```
sudo lynis audit system
```

## 5. ClamAV – Antivirusscanner för Linux

Bra för att analysera exempelvis e-postservrar eller användarkataloger. Ingen realtidsskanning, 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 tidigare 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/`.

## 9.15.1 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?

## 9.15.2 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>

### 9.15.3 Egna anteckningar

(Lämna utrymme för reflektioner, kommandon du vill testa eller egna tips.)

## 9.16 Jämförelse: Säkerhet i Windows och Linux

Att förstå IT-säkerhet innebär inte bara att kunna använda rätt verktyg – utan också att kunna välja rätt verktyg för rätt situation. Många skolor, företag och organisationer använder både Windows och Linux i sin IT-miljö. I detta kapitel jämför vi hur säkerhet fungerar i de två systemen: deras filosofi, styrkor, svagheter och typiska användningsområden.

### 9.16.1 Säkerhetsfilosofi – inbyggt vs. tillagt

Windows är designat med användarvänlighet i fokus och har med tiden byggt på fler säkerhetslager. Säkerheten är integrerad, men ofta beroende av användarinställningar eller företagsstyrda policies. Det mesta är "på" från början, men kan justeras eller inaktiveras.

Linux å andra sidan har säkerhet som en grundbult i systemdesignen. Allt – från åtkomsträttigheter till användarskalsbehörigheter – är strängt separerat. Som standard är systemet mycket restriktivt, och användaren måste aktivt ge tillstånd för att något ska hända.

👉 Filosofiskt:

- Windows: Skydda användaren från misstag.
- Linux: Ge kontroll – men förutsätt ansvar.

## 9.16.2 Kontomodeller och behörighet

Funktion	Windows	Linux
Vanlig användare	Standardkonto – begränsade rättigheter	Normalanvändare – kan ej påverka system
Admin	Administratör – full åtkomst	root – superuser med total kontroll
Gästkonto	Finns, men avaktiverat som standard	Skapas manuellt om det behövs
Åtkomstkontroll	NTFS-rättigheter, Group Policy	Filrättigheter (rwx), sudo, grupper
Inloggning	Lösenord, PIN, Microsoft-konto	Lösenord, SSH, nyckelbaserad inloggning

Linux använder ofta sudo-kommandot för att ge tillfälliga administratörsrättigheter – ett säkrare alternativ än att logga in som root. I Windows kan användare ändå ändå vara konstant inloggade som administratör, vilket ökar risken vid attacker.

## 9.16.3 Malware och hotbild

Windows är det mest använda operativsystemet i världen – vilket gör det till ett större mål för attacker. Malware, ransomware, spyware och phishing riktas ofta mot Windowsanvändare, eftersom angripare får större utdelning där.

Linux är tekniskt svårare att angripa för nybörjare, och används mest på servrar, där attackerna är mer riktade (t.ex. via SSH eller sårbara webbtjänster).

Vanlig malware:

- Windows: .exe-filer, makron i Office, PowerShell-script
- Linux: Rootkits, manipulativa bash-script, oskyddade portar

Båda systemen är säkra – men bara om de är rätt konfigurerade.

## 9.16.4 Antivirus och säkerhetsverktyg

Typ av skydd	Windows	Linux
Antivirus	Inbyggt (Defender)	Kräver manuell installation (ClamAV, Sophos m.fl.)
Brandvägg	Windows Firewall	ufw, iptables, firewalld
Autentisering	NTLM, Kerberos	SSH, PAM, LDAP
Kryptering	BitLocker	LUKS, VeraCrypt
Policyverktyg	Group Policy, AppLocker	SELinux, AppArmor, fail2ban

Windows har fler inbyggda verktyg, särskilt i företagsversioner. Linux kräver att administratören själv installerar och aktiverar dessa – men det ger också större flexibilitet.

Exempel:

- På en Windowsmaskin aktiveras Defender automatiskt.
- På en Ubuntu-server måste du själv installera clamav och köra freshclam för att uppdatera virusdefinitioner.

## 9.16.5 Loggning och övervakning

Loggar är ett av de viktigaste verktygen för att spåra intrångsförsök och misstag.

- Windows: Använder Event Viewer – med kategorier som System, Application, Security. Loggning av t.ex. inloggningar, USB-anslutningar och Windows Defender-hot.
- Linux: Använder journalctl, /var/log/-katalogen, och systemd-loggar. Allt från SSH-försök till kernelloggar sparas.

I båda system kan loggarna exporteras till SIEM-system (t.ex. Wazuh eller Splunk) för central analys.

## 9.16.6 Användningsområden och sårbarheter

Scenario	Windows	Linux
Skrivbordsmiljöer	Vanligt i skolor, företag, privat	Ovanligt (men t.ex. Ubuntu används i utbildning)
Servrar	Vanligt – särskilt AD, Exchange	Dominerar – webbservrar, databaser
Administration	GUI-centrerat (MMC, GPO)	Terminalcentrerat (sudo, systemctl)
Vanliga hot	Phishing, ransomware, makron	Felkonfiguration, root-exploits, publika tjänster

Linux är mer populärt på serversidan, medan Windows är normen på klientdatorer. Därför ser också hoten olika ut. En server som glömts öppen mot internet är farlig – oavsett system.

## 9.16.7 Sammanfattning

Windows och Linux har olika angreppssätt när det gäller säkerhet:

- Windows erbjuder ett mer användarvänligt men samtidigt utsatt ekosystem, med inbyggt skydd och central administration via Group Policy.
- Linux ger större kontroll, men kräver mer ansvar. Det är kraftfullt – men sårbart om du konfigurerar det fel.

Båda systemen är säkra – men bara om de används säkert.

Ett tryggt system handlar inte om vilket operativsystem du väljer – utan hur du hanterar det.

## 9.16.8 Kontrollfrågor

1. Vad är skillnaden mellan säkerhetsfilosofin i Windows och Linux?
2. Hur fungerar sudo i Linux, och varför är det ett säkert alternativ?
3. Vilka typer av skadlig kod är vanligast i Windows respektive Linux?
4. Nämnn minst tre säkerhetsverktyg som är vanliga i Windows.
5. Vad gör LUKS och hur liknar det BitLocker?
6. Vad är syftet med Event Viewer i Windows och hur används det?
7. Vad sparas i /var/log/-katalogen i Linux?
8. Varför är det viktigt att kombinera loggning med övervakning i båda systemen?
9. Vad är AppArmor och vilken roll spelar det i Linuxsäkerhet?
10. Vilket system är vanligast för klientdatorer – och vilket för servrar?

## 9.16.9 Fördjupningslänkar

1. Microsoft – [Windows Security documentation](#)
2. Ubuntu – [Ubuntu Server Guide - Security](#)
3. Red Hat – [SELinux Project](#)
4. AV-Test – <https://www.av-test.org/en/>
5. Comparitech – Windows vs Linux Security Compared -  
<https://bluegoatcyber.com/blog/linux-vs-windows-a-security-comparison/>

## 9.16.10 Egna anteckningar

Här kan du skriva ned egna reflektioner, skillnader du själv märkt, eller anteckningar inför framtida arbete:

## 9.17 IT-säkerhet ur ett verksamhetsperspektiv

### 9.17.1 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 oftast 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 vanligtvis 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.

### 9.17.1.1 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-säkerhetspolicyer.
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 säkerhetspolicyen?
9. Varför är det viktigt att anpassa säkerhetspolicyen till verksamheten?
10. Hur bör en organisation följa upp och förbättra sin säkerhetspolicy?

## 9.17.1.2 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/>

### 9.17.1.3 Egna anteckningar

*(Här kan du anteckna egna reflektioner, exempel eller frågor kring säkerhetspolicys och rutiner.)*

## 9.17.2 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-expertyper – 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.

### 9.17.2.1 Kontrollfrågor

1. Varför anses användaren ofta vara den svagaste länken i säkerhetskedjan?
2. Vad är målet med användarutbildning inom IT-säkerhet?
3. Nämnn 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?

## 9.17.2.2 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/>

### **9.17.2.3 Egna anteckningar**

*(Här kan du skriva ner egna reflektioner, idéer för utbildning, eller exempel på incidenter som hade kunnat undvikas med rätt kunskap.)*

### 9.17.3 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.

### 9.17.3.1 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?

### 9.17.3.2 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/>

### 9.17.3.3 Egna anteckningar

*(Skriv ner exempel från skolan, företagsmiljöer eller projekt du deltagit i.  
Har ni tydliga roller?)*

## 9.17.4 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.

### 9.17.4.1 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.

## 9.17.4.2 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>

### 9.17.4.3 Egna anteckningar

*(Har du varit med om ett IT-problem? Hur löstes det? Vad lärde du dig?)*

## 9.17.5 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 säkerhetspolicy 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.

## 9.17.5.1 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 policymen?
7. Vad innebär uppföljning och hur kan den göras?
8. Ge exempel på något som kanstå i en policy.
9. Hur kan dokumentation hjälpa vid en incident?
10. När ska en policy revideras?

## 9.17.5.2 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>

### 9.17.5.3 Egna anteckningar

*(Finns det en policy där du är? Har du läst den? Vad skulle du vilja ändra eller lägga till?)*

## 9.17.6 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

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se första sidan

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.

## 9.17.6.1 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?

## 9.17.6.2 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/>

### 9.17.6.3 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.)

## 9.18 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.

### 9.18.1 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 få mottagaren att klicka på en länk, lämna ifrån sig lösenord eller installera skadlig kod.

Exempel på phishing:

- Ett mejl från "Skatteverket" som ber dig logga in med BankID.
- Ett SMS från "PostNord" med en länk till en falsk spårningssida.
- Ett samtal från någon som utger sig för att vara från IT-avdelningen och ber om ditt lösenord.

Phishing kan vara massutskick, men också riktade attacker som spear phishing (mot en specifik person) eller CEO-fraud (där angriparen låtsas vara chef och kräver exempelvis en snabb överföring).

Skydd:

- Utbildning är avgörande – användare måste veta hur angreppen ser ut.
- Använd e-postfilter, multifaktorsautentisering (MFA) och varningssystem för ovanliga språk eller avsändare.
- Tänk efter innan du klickar – fråga dig själv: "Verkar detta rimligt?"

## 9.18.2 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 låsa upp filerna.

Ransomware sprids ofta via phishingmejl, sårbarheter i system eller oskyddade RDP-portar. Vissa varianter använder double extortion – de stjäl filerna först och hotar sedan att publicera dem.

Exempel på kända ransomware: WannaCry, REvil, LockBit.

Skydd:

- Uppdatera programvara och operativsystem regelbundet.
- Ta offline-backuper – se till att backupen inte påverkas av kryptering.
- Använd starka lösenord och MFA.
- Öva på incidenthantering – vet ni vad ni ska göra om det händer?

## 9.18.3 Zero-day exploits

En zero-day är en sårbarhet som är okänd för programvarans tillverkare – och som därför ännu inte har någon patch. Angripare kan utnyttja den direkt, innan ett skydd finns tillgängligt.

Exempel: Log4Shell (2021) – en allvarlig zero-day i Java-loggning som drabbade miljontals system.

Eftersom zero-days inte kan blockeras via traditionella patchar krävs andra skyddsåtgärder.

Skydd:

- Minimera attackytan: stäng onödiga portar och ta bort överflödiga tjänster.
- Använd IDS/IPS-system och loggning för att upptäcka avvikande beteende.
- Tillämpa Least Privilege – begränsa vad användare och processer får göra.

- Håll dig uppdaterad via säkerhetsflöden som CERT-SE.

### 9.18.4 Honeypots – lura angriparen

En honeypot är en digital fälla. Den ser ut som ett riktigt system – till exempel en SSH- eller webbserver – men är skapad för att bli attackerad.

Syftet är att:

1. Locka angripare och avslöja deras metoder
2. Skydda riktiga system genom att fungera som avledare
3. Samla in information om hot och angrepp

I skolmiljöer eller labbar används ofta verktyg som Cowrie (för SSH/Telnet) eller Dionaea (för att fånga malware).

En honeypot måste isoleras från produktionsnätverket – annars riskerar den att utnyttjas som språngbräda in i resten av systemet.

### 9.18.5 Penetrationstester – simulerade attacker

Ett penetrationstest (pentest) är en kontrollerad attack som genomförs av en etisk hackare – med systemägarens tillstånd – för att hitta sårbarheter innan angripare gör det.

Ett pentest kan inkludera:

- Sårbarhetsskanning
- Exploatering av kända brister
- Social engineering-tester
- Fysisk intrångstest (t.ex. försöka ta sig in i ett serverrum)

Vanliga verktyg:

- Kali Linux
- Metasploit
- Nmap
- Burp Suite

Viktigt: Ett pentest får aldrig ske utan godkännande. Etiska hackare följer lagar, dokumenterar sina åtgärder och lämnar en rapport till uppdragsgivaren.

## 9.18.6 Säkerhet i molnet och vid distansarbete

När allt fler jobbar hemifrån och IT-tjänster flyttas till molnet förändras också hotbilden. Säkerhet måste nu omfatta både molnplattformar och användare utanför kontoret.

### **Utmaningar i molnet:**

- Oklara ansvarsgränser – vem skyddar vad?
- Felaktiga konfigurationer, som öppna S3-buckets
- Otillräcklig rättighetsstyrning

### **Utmaningar vid distansarbete:**

- Osäkra Wi-Fi-nätverk
- Privata enheter utan skydd
- Minskad kontroll och övervakning

### **Skyddsåtgärder:**

- Tillämpa Zero Trust: Ingen är automatiskt betrodd
- Använd MFA överallt
- Centraliserad identitets- och åtkomsthantering
- VPN och endpoint-skydd
- Loggning och övervakning även utanför kontorsnätverket

Molntjänster kan vara mycket säkra – men bara om de är rätt konfigurerade, och om användarna vet vilket ansvar de har.

## 9.18.7 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. Nämnn två risker med dåligt konfigurerade molntjänster.
9. Hur kan man skydda sig vid distansarbete?
10. Varför är MFA extra viktigt vid molnanvändning?

## 9.18.8 Fördjupningslänkar

- CERT-SE - Aktuella säkerhetshot  
<https://www.cert.se>
- KnowBe4 - Träning mot phishing  
<https://www.knowbe4.com>
- NoMoreRansom - Hjälp mot ransomware  
<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>

## 9.18.9 Egna anteckningar

(Lämna plats för egna reflektioner, case eller exempel.)

## 9.19 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.

### 9.19.1 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 samlar in, analysera och lagra loggar på ett säkert sätt.

### 9.19.2 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 det lättare att upptäcka mönster och hot.

#### Linux:

- Syslog – äldre system som skriver loggar till /var/log/. Vanliga verktyg är rsyslog och syslog-ng.
- journald – nyare loggsystem som används i systemd-baserade distributioner. Visas med journalctl.

## Windows:

- Event Viewer – inbyggt verktyg som grupperar loggar i System, Security och Application.
- För central hantering används exempelvis Event Forwarding, Graylog, Wazuh eller Elastic Stack (ELK).

💬 Loggformaten skiljer sig – men syftet är detsamma: Vad hände, när, hur och varför?

## 9.19.3 Övervakningsverktyg – Wazuh, Zabbix, Nagios

Loggning är passiv – den registrerar vad som hänt. Övervakning är aktiv – den reagerar på vad som händer just nu.

### Exempel på övervakningsverktyg:

- Wazuh – Open source XDR/EDR och SIEM. Samlar loggar, upptäcker hot och larmar.
- Zabbix – övervakar CPU, RAM, disk, nätverk m.m. och ger grafisk översikt.
- Nagios – klassiker inom driftövervakning, särskilt för nätverkstjänster.

### Verktygen kan:

- Skicka e-post eller SMS vid fel
  - Skapa grafer över trender (t.ex. CPU-belastning)
  - Identifiera attacker och onormal aktivitet
  - Dokumentera SLA, upptid och tillgänglighet
- 💡 Bäst skydd får du genom att kombinera loggar, övervakning och larm.

## 9.19.4 SIEM – Samlad analys av säkerhetsloggar

SIEM står för Security Information and Event Management. Ett SIEM-system samlar in och analyserar loggar från många källor – och letar efter mönster och hot i realtid.

### **SIEM hjälper dig att:**

- Korrelera händelser ("logg A + logg B = möjlig attack")
- Identifiera ovanlig aktivitet
- Reagera direkt på incidenter
- Få en samlad säkerhetsöversikt

### **Exempel på SIEM-lösningar:**

- Wazuh - inkluderar SIEM-funktioner
- Graylog - fokuserar på logg- och eventanalys
- Splunk - kommersiell SIEM-lösning med kraftfull analys
- ELK Stack - öppen och flexibel (Elasticsearch, Logstash, Kibana)

 SIEM är särskilt viktigt i större nätverk där många system interagerar.

## **9.17.5 Att tänka på vid övervakning – etik och lagstiftning**

Att övervaka system är nödvändigt – men det innebär också ansvar.  
Övervakning och loggning påverkar individers integritet och måste ske på ett korrekt sätt.

Grundregler:

- Informera användare om att loggning sker
- Spara inte mer än nödvändigt
- Skydda loggarna mot obehörig åtkomst
- Följ GDPR, NIS2 och andra relevanta regelverk

 Etik handlar om tillit. Övervakning ska ske för att skydda, inte för att spionera.

## 9.19.6 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?

## 9.19.7 Fördjupningslänkar

- Wazuh – Officiell webbplats

<https://wazuh.com>

- Graylog – Logghantering

<https://www.graylog.org>

- Zabbix – Övervakning

<https://www.zabbix.com>

- Splunk – SIEM-lösning

<https://www.splunk.com>

- MSB – Loggning och incidenthantering

<https://www.msb.se/sv/rad-till-verksamheter/informationssakerhet/tekniska-atgarder/loggning-och-incidenthantering/>

## 9.19.8 Egna anteckningar

(Lämna plats för reflektioner, egna verktygstips eller exempel.)

## 9.20 Säkerhet i skarpa miljöer och incidenthantering

Att förstå teorin bakom IT-säkerhet är en sak – men det är i verkligheten, med röriga miljöer och slarviga användare, som säkerheten sätts på prov. I skarpa miljöer måste man inte bara förebygga attacker – utan också snabbt upptäcka dem, begränsa skadan och återställa systemen.

### 9.20.1 Vad menas med "skarp miljö"?

En skarp miljö är en miljö i drift, med riktiga användare, verkliga data och aktiva arbetsprocesser. Det kan handla om företag, skolor, sjukhus eller myndigheter – miljöer där driftstopp eller dataintrång får verkliga konsekvenser.

Exempel:

- En skola där elevregister läcker
- Ett företag vars webshop slås ut av ransomware
- En kommun som förlorar åtkomst till sina e-postservrar

Krav i skarpa miljöer:

- Tydlig backupstrategi
- Regelbundna uppdateringsrutiner
- Övervakning av system och nätverk
- Färdig incidentplan

### 9.20.2 Vad gör man när det händer något?

Ingen miljö är helt säker. Oavsett hur bra skydd du har kan incidenter inträffa – från skadlig kod till dataintrång eller systemfel.

En incidenthanteringsplan bör bestå av fem steg:

1. Identifiera – Vad har hänt?
2. Meddela – Informera rätt personer (IT, chef, juridik, etc.)

3. Isolera – Begränsa skadan (koppla ur nätverk, stäng ned system)

4. Analysera – Undersök orsaken

5. Återställ – Återstarta systemen och stärka skyddet

 Ha kontaktvägar och ansvarsfördelning klara i förväg. I en kris finns ingen tid för improvisation.

### 9.20.3 Att kommunicera i kris – vem säger vad till vem?

I en IT-incident är kommunikationen ofta lika viktig som tekniken. Otydlig information kan skapa:

- Panik bland användare
- Skadat förtroende hos elever, kunder eller medarbetare
- Onödig spridning av rykten

En bra kommunikationsplan innehåller:

- Vem ansvarar för att informera?
- Vilka ska få information? (personal, elever, externa aktörer)
- Vad ska sägas – och vad bör undvikas?
- Hur ska det sägas? (e-post, möte, pressmeddelande)

Exempel från skolmiljö:

Om en elev laddar ner ransomware i datasalen – vem informerar lärarna, eleverna och vårdnadshavarna?

### 9.20.4 Dokumentation, bevis och återställning

När incidenten är under kontroll börjar det viktiga efterarbetet. Då handlar det om att lära sig av det som hänt och förhindra att det händer igen.

Efterarbete inkluderar:

- Dokumentera allt (tider, IP-adresser, konton, åtgärder)
- Spara loggar och filer (kan behövas som bevis)
- Gör en teknisk och organisatorisk analys

- Förbättra rutiner, uppdatera system, utbilda personal

Externa kontakter kan krävas:

- Integritetsskyddsmyndigheten (vid personuppgiftsläckor)
- Försäkringsbolag
- Polisen (om det rör sig om brott)

Backup:

Vet du var backupen finns? Och hur snabbt du kan återställa systemen?

### 9.20.5 Vanliga misstag vid incidenthantering

1. Ingen övning – En plan på papper räcker inte. Öva i praktiken!
2. Ingen loggning – Utan loggar är det omöjligt att förstå vad som hänt.
3. För sen reaktion – Många attacker kan stoppas i tid med snabb respons.
4. Kommunikationskaos – Olika versioner från olika håll förvärrar situationen.
5. Ingen uppföljning – Utan analys kommer samma problem tillbaka.

 Tips: Gör en årlig övning – exempelvis ”Vad gör vi om någon installerar ransomware?” – och låt olika roller delta i scenariot.

### 9.20.6 Incidentprotokoll – exempel från verkligheten

När något allvarligt händer i en IT-miljö är det avgörande att man dokumenterar allt som sker – både för att förstå vad som hänt och för att kunna följa upp händelsen på rätt sätt. Ett incidentprotokoll fungerar som ett loggboksliknande dokument som följer hela händelseförfloppet: från upptäckt till åtgärd.

*Exempel: Ransomware i skolmiljö*

Händelse

Protokollpost

Datum och tid 2025-03-05, kl. 09:42

Rapporterad av IT-lärare (Robin Bräck), efter att elever rapporterat att

Händelse	Protokollpost
	filer inte gick att öppna
Typ av incident	Ransomware-attack (misstänkt genom nedladdning av .zip-fil i datasalen)
Berörda system	Tre elevdatorer i datasal B104 (Linux Mint), en lokal filserver (Ubuntu)
Första åtgärd	Datorerna kopplades direkt ur nätverket (både Wi-Fi och kabel). Server stängdes av.
Kontaktade personer	Rektor, skolans IT-ansvarig, leverantör av antiviruslösning
Teknisk analys	Loggar från /var/log/auth.log och fail2ban visade inloggningssök från okända IP-adresser.
Källhändelse	Elev laddade ner ett påstått "Minecraft-modpaket" från en okänd webbplats. Filen innehöll ett script som startade kryptering av hemkatalogen.
Påverkan	Filer i hemkatalogen låsta. Ingen backup fanns lokalt. Servern återställd från fjärrbackup (nattlig rsync).
Kommunikation	Lärarna informerades via internkanal. Eleverna informerades lektionen efter. Vårdbudshavare fick e-post via skolplattformen.
Återställning	Systemen ominstallerades. Berörda elever fick nya inloggningssuppgifter. Hemkataloger återskapades från 2 dagar gamla offsite-backuper.
Efterarbete	Genomgång i klassen om IT-säkerhet och phishing. Webfilter uppdaterades. Nya riktlinjer för nedladdningar.
Lärdomar	<ul style="list-style-type: none"><li>- Webfilter måste vara mer aggressivt</li><li>- Backuper måste testas oftare</li><li>- Elever behöver mer träning i att känna igen</li></ul>

Händelse	Protokollpost
misstänkta filer	

 Tips: Ett sådant protokoll bör sparas på en skyddad plats, och gärna vara i mallform så att alla i IT-organisationen vet vad som ska fyllas i.

## 9.20.7 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?

## 9.20.8 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>

## 9.20.9 Egna anteckningar

(Plats för reflektioner, rutiner ni har på skolan, eller saker du vill testa i praktiken.)

## 9.21 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, lurar att lämna ifrån oss information, eller glömmer uppdatera system i tid. Det är här social engineering kommer in - konsten att manipulera människor snarare än maskiner.

### 9.21.1 Vad är social engineering?

Social engineering är en form av psykologisk manipulation som syftar till att få någon att göra något de inte borde. Det kan handla om att klicka på en länk, lämna ut ett lösenord eller installera skadlig kod - men istället för att tvinga sig in med teknik, så lurar angriparen dig att öppna dörren själv. Det bygger ofta på stress, auktoritet ("Jag ringer från IT"), nyfikenhet ("Se bifogade dokument") eller rädska ("Ditt konto stängs om du inte loggar in nu!"). Ett vanligt exempel är att någon ringer till receptionen och låtsas vara tekniker. De ber om fjärråtkomst till en dator för att "installera en uppdatering", men i själva verket tar de kontroll över systemet.

### 9.21.2 Vanliga sociala attacker

Det finns många olika varianter av social engineering, och angriparna anpassar sig efter den kanal som funkar bäst. Phishing sker ofta via e-post, där du lockas att klicka på en falsk länk - kanske för att "hämta ett paket". Spear phishing är en mer riktad variant, där ett mejl riktas till en specifik person, till exempel ett falskt mejl till rektorn från "ekonomiavdelningen". Vishing är telefonbedrägeri där någon ringer och låtsas vara från banken, medan smishing sker via SMS. Pretexting bygger på en förtroendebaserad bluff, som en falsk kollega som ber om dina inloggningsuppgifter. Tailgating innebär att någon fysiskt följer efter dig genom en säkerhetsdörr med en ursäkt som att de glömt sitt passerkort.

### 9.21.3 Varför fungerar det så ofta?

Människan är ofta den svagaste länken i säkerhetskedjan. Inte för att vi är dumma, utan för att vi vill hjälpa till, har bråttom, litar på andra och inte

alltid är vana vid digitala hot. Angripare är ofta skickliga på att skapa trovärdiga situationer som spelar på våra instinkter. En välformulerad bluff kan lura både nybörjare och erfarna användare. Flera studier visar att över 90 % av alla cybersäkerhetsincidenter är relaterade till mänskligt beteende.

#### 9.21.4 Träning och medvetenhet

För att skydda sig mot social engineering krävs medvetenhet. Det handlar inte om att alla ska bli säkerhetsexperter, utan att alla ska känna igen de vanligaste fällorna och våga säga nej. Det bästa skyddet är en kultur där säkerhet pratas om, där det är okej att vara misstänksam, och där man vet vem man ska kontakta om något känns fel. Träning kan ske genom bluffmejl som skickas ut internt i syfte att testa och utbilda, genomgångar av riktiga exempel, rollspel eller påminnelser i vardagen. Ju mer vi pratar om hoten, desto bättre rustade blir vi.

#### 9.21.5 Att skapa en stark säkerhetskultur

Säkerhet handlar i grunden om människor. Ett låst serverrum hjälper inte om någon släpper in en främling "för att de såg ut att jobba här". En stark säkerhetskultur kännetecknas av att det är lätt att rapportera misstänkta händelser, att regler är tydliga, att man får beröm snarare än skäll när man säger ifrån, och att säkerhet tas upp ofta – inte bara när något redan har gått fel.

## 9.21.6 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.

## 9.21.7 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/>

© 2025 Robin Bräck

Det här verket är skyddat av upphovsrätt. Se första sidan

## 9.21.8 Egna anteckningar

(Plats för egna reflektioner, exempel från skolan eller egna erfarenheter.)

## 9.22 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.

### 9.22.1 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 på sådant som kan loggas är inloggningfsöök (både lyckade och misslyckade), filsystemsåtkomst, programinstallationer, systemuppdateringar, brandväggsregler som ändrats samt trafik som blockeras eller släpps igenom. Loggar fungerar som en svart låda i IT-system – ju mer detaljerad loggning, desto bättre analysmöjlighet vid en incident.

### 9.22.2 Vad som bör loggas

Att logga allt är sällan praktiskt. Det ökar lagringsbehovet och försvårar analys. Fokus bör ligga på händelser som är säkerhetskritisika: inloggningar (särskilt felaktiga), ändringar i användarbehörigheter, konfigurationsändringar, start och stopp av tjänster, brandväggsändringar, systemfel och kraschloggar, samt åtkomst till känsliga filer eller API:er. I Active Directory-miljöer bör kontoskapande och Group Policy-ändringar loggas. I webbtjänster är det viktigt att registrera IP-adresser, användaragenter, sessions-ID, begärda URL:er och statuskoder – men man bör undvika att lagra för mycket persondata.

### 9.22.3 Övervakning i realtid

I vissa miljöer krävs mer än bara passiv loggning – man behöver kunna upptäcka hot i realtid. Det gäller särskilt system som är exponerade mot

internet, har fjärråtkomst eller innehåller känslig data. Wazuh kombinerar logginsamling och incidentrespons, Fail2ban reagerar på mönster som många felinloggningar, och SIEM-system som Splunk, ELK eller Microsoft Sentinel används för att upptäcka komplexa hot. IDS/IPS-system kan dessutom reagera på misstänkt nätverkstrafik.

#### 9.22.4 Revision och efteranalys

Revision innebär att man granskar loggar i efterhand för att förstå en incident eller kontrollera att regler följs. Det kan handla om att ta reda på vem som ändrat i en databas, när ett konto skapades, eller vilka IP-adresser som haft åtkomst. Revision är centralt för att visa efterlevnad av exempelvis GDPR, ISO 27001 eller NIS2, men det är också ett viktigt verktyg i det dagliga säkerhetsarbetet.

#### 9.22.5 Juridiska aspekter på loggning

Loggfiler innehåller ofta personuppgifter – som användarnamn och IP-adresser – och omfattas därför av dataskyddslagstiftning. Det är viktigt att informera användare om att loggning sker, att radera loggar efter en rimlig tid, och att aldrig logga känslig information som lösenord eller kortnummer. Loggfiler ska även skyddas mot obehörig åtkomst. En loggfil är alltså både en teknisk och juridisk resurs.

#### 9.22.6 Vanliga misstag

Det är vanligt att man tror att loggning är aktiverad – utan att ha kontrollerat det. Många system kräver manuell aktivering. Ett annat misstag är att spara loggar lokalt utan backup – något som kan få stora konsekvenser vid t.ex. ransomware. Många analyserar heller inte loggar förrän något redan har gått fel. Och kanske vanligast av allt: man glömmer att tänka på integritet – överdrivet detaljerade loggar kan i sig utgöra ett hot.

## 9.22.7 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?

## 9.22.8 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/>

## 9.22.9 Egna anteckningar

(Lämna plats för egna reflektioner och anteckningar här.)

## 9.23 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.

### 9.23.1 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. Vanliga orsaker till dataförlust inkluderar ransomware, oavsiktlig radering, hårdvaruhaveri (t.ex. döda SSD:er), brand, översvämnning, elfel, sabotage eller 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.

### 9.23.2 Typer av backup

Det finns flera typer av backup, med olika för- och nackdelar beroende på miljö och behov. En full backup innebär att en komplett kopia av all data tas. Den är längsammare men enkel att återställa. Inkrementell backup sparar bara det som ändrats sedan senaste backupen – effektivt men kräver fler steg vid återställning. Differentiell backup sparar allt som ändrats sedan senaste fulla backup och är en kompromiss mellan snabbhet och enkelhet. Snapshot är en ögonblicksbild av ett system, vanligt i virtualiseringsmiljöer som Proxmox. Molnbaserad backup innebär att data skickas till en extern tjänst som AWS eller Backblaze – bra vid fysiska katastrofer men beroende av uppkoppling och avtal. Offline-backup innebär att data sparas på fristående media som kopplas bort från nätverket – extra skydd mot ransomware.

### 9.23.3 3-2-1-regeln

En klassisk tumregel för säker backup är 3-2-1-regeln: ha tre kopior av datan, på två olika typer av media, och en kopia offsite. Ett exempel är originalet på servern, en kopia på en lokal NAS, och en tredje kopia i molnet via Backblaze. Denna metod skyddar mot tekniska fel, mänskliga misstag och fysiska katastrofer.

### 9.23.4 Återställning – ofta glömd men avgörande

Att kunna återställa är viktigare än att kunna ta backup. Det är lätt att tro att backupen fungerar – tills den behövs. Därför bör återställning testas regelbundet. Det är viktigt att dokumentera hur återställning går till steg för steg, att återställa både enstaka filer och hela system inklusive databaser, samt att kontrollera integriteten hos backupfiler med exempelvis checksummor. Tiden det tar att återställa är också viktig – ett långsamt återställningsförfarande kan bli kostsamt.

### 9.23.5 Kryptering och integritet

Backupar innehåller ofta känslig information. Om den datan hamnar i fel händer kan det få allvarliga konsekvenser. Därför bör backupfiler krypteras – till exempel med VeraCrypt eller en inbyggd lösning – och backupservrar skyddas med brandvägg och stark autentisering. Det är också klokt att kontrollera loggar för att se om någon obehörig försökt läsa eller ladda ner en backup. En läckt backup kan ge en angripare tillgång till hela din miljö – vilket kan vara värre än att inte ha någon backup alls.

### 9.23.6 Vanliga misstag

Ett vanligt fel är att bara ha en backup – om den går förlorad finns inget skydd kvar. Ett annat misstag är att spara backupen på samma disk eller server som originalet – om den kraschar är båda borta. Många backuprutiner är manuella och glöms bort, därför bör de automatiseras. Att inte testa återställning gör att man bara tror att det fungerar, och att inte kryptera backupfiler innebär en risk för dataläckor.

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan

## 9.23.7 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?

## 9.23.8 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/>

## 9.23.9 Egna anteckningar

(Lämna plats för egna reflektioner och anteckningar här.)

## 9.24 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.

### 9.24.1 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 inkluderar åtkomst från olika platser och enheter, automatisk synkronisering och versionering, skydd mot lokala katastrofer samt skalbarhet. Nackdelar är behovet av stabil internetanslutning, potentiella kostnader vid stora datamängder, integritetsrisker och minskad kontroll jämfört med lokal lagring.

Att förstå var och hur datan lagras är avgörande – särskilt ur integritets- och GDPR-perspektiv.

### 9.24.2 Olika typer av molnbackup

Molnbackup finns i olika modeller:

- Molnet som primär backup: Hela säkerhetskopieringen sker i molnet, t.ex. via Backblaze eller iDrive. Passar hemanvändare eller småföretag.
- Hybridbackup: Data sparas både lokalt och i molnet, vilket ger snabb återställning lokalt och skydd vid katastrof.
- SaaS-backup: Säkerhetskopiering av molntjänster som Microsoft 365 eller Google Workspace. Viktigt att veta att leverantören inte ansvarar för din data.

- Backup i molninfrastruktur (IaaS/PaaS): Används när du har servrar i exempelvis Azure eller AWS och behöver extern backup.

### 9.24.3 Kryptering och integritet i molnbackup

Säker backup kräver kryptering. Två vanliga modeller:

- Klientsidans kryptering: Daten krypteras innan den lämnar datorn. Endast du har nyckeln. Ger hög säkerhet men kräver säker nyckelhantering.
- Serversidans kryptering: Daten skickas okrypterad och krypteras av leverantören. Lättare att hantera, men lägre säkerhetsnivå.

Om du själv har nyckeln (end-to-end encryption) har du kontrollen. Om leverantören har nyckeln finns risk för intrång, dataläckor eller statlig insyn.

Rättsliga krav att tänka på:

- GDPR: Du ansvarar för hur persondata behandlas, även i molnet.
- Schrems II: Om data förs över utanför EU krävs extra skydd.
- Dataplacering: Kontrollera var informationen faktiskt lagras.

### 9.24.4 Återställning från molnet – utmaningar och tips

Att ta backup är lätt – det är återställningen som visar om den fungerar. Vanliga problem är lång nedladdningstid, bandbreddsgränser, borttappade filversioner och krypterade backuper utan fungerande nyckel.

Tips: Testa återställning regelbundet, dokumentera processen, och överväg tjänster som kan leverera backupdata på fysisk disk ("data shuttle").

### 9.24.5 Exempel på populära molnbackup-tjänster

Några exempel på tjänster:

- Backblaze: Obegränsad backup, enkel prissättning. Begränsad konfiguration.

- iDrive: Kombinerar moln och lokal backup. Gränssnittet kan vara rörigt.
- CrashPlan: Bra för företag, versionshantering. Ej längre för privatpersoner.
- Wasabi: Billigt, inga egressavgifter. Kräver manuell hantering.
- Google Workspace Vault: Integrad med Google, bra sök. Ej traditionell backup.

För privatpersoner kan även OneDrive och iCloud användas – men de erbjuder begränsad kontroll och automatisering.

#### 9.24.6 Fallgropar och rekommendationer

Molnet är inte ett mirakel – du har fortfarande ansvar. Det är lätt att förväxla synkronisering med riktig backup, att missa vilka filer som faktiskt inkluderas, eller att ha för kort versionhistorik.

Rekommendation: Använd hybridlösning (lokalt + moln), välj tjänst med stark kryptering och dokumentera alla backup- och återställningsrutiner. Och glöm inte att testa dem regelbundet.

## 9.24.7 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?

## 9.24.8 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>

## 9.24.9 Egna anteckningar

(Lämna plats för reflektioner, exempel från undervisningen eller egna molnerfarenheter.)

## 9.25 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 – Websidor 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 kompromitterad maskin för att ta sig vidare i ett nätverk.
- Backdoors – Dolda åtkomstvägar skapade av utvecklare eller angripare.

## Phases of the Intrusion Kill Chain



### *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 bygga skydd mot dem.

#### *Red team / Blue team – övning i offensivt och defensivt tänkande*

Inom cybersäkerhet används ofta begreppen red team och blue team för att öva både attacker och försvar. Ett red team agerar som angripare och försöker bryta sig in i ett system med verkliga metoder – men inom kontrollerade ramar. Ett blue team är försvararna som ska upptäcka, stoppa och dokumentera angreppen i realtid.

Syftet är att förbättra säkerheten genom realistiska scenarier där båda sidor får träna sina färdigheter. Ofta används även begreppet purple team, där red och blue team samarbetar för att maximera lärande, identifiera luckor och stärka organisationens skyddsförmåga.

## 9.25.1 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?

## 9.25.2 Fördjupningslänkar

- OWASP – Cyber Kill Chain Explained  
[https://owasp.org/www-community/attack-methods/Cyber\\_Kill\\_Chain](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>

### 9.25.3 Egna anteckningar

(Lämna plats för egna reflektioner och fördjupningar här.)

## 9.26 Grundläggande säkerhetsprinciper

Att förstå IT-säkerhet handlar inte bara om att veta hur attacker fungerar – utan också om hur man bygger ett robust försvar. Det finns ett antal återkommande säkerhetsprinciper som används inom allt från små nätverk till stora myndighetssystem. De utgör grunden för säkra arkitekturen, rutiner och beslut.

### 9.26.1 Least Privilege (minsta privilegium)

Principen om minsta privilegium innebär att varje användare, program eller process bara ska ha de behörigheter som krävs för att kunna utföra sin uppgift – och inte mer.

Exempel:

- En IT-tekniker kan behöva administratörsbehörighet tillfälligt, men bör normalt arbeta från ett vanligt konto.
- Ett backup-program bör ha läsrättigheter till filer, men inte kunna ta bort eller ändra dem.

Syftet är att minska skadan om ett konto eller en process blir kompromitterad. Ju mindre rättigheter, desto mindre kan gå fel.

### 9.26.2 Defense in Depth (försvar i flera lager)

Defense in Depth handlar om att bygga flera lager av skydd, så att om ett misslyckas finns andra kvar. Tänk på en borg: murar, vallgrav, vakter och reservutgångar.

Inom IT kan det handla om:

- Kryptering av hårddiskar
- Brandväggar och antivirus
- IDS/IPS-system
- Tvåfaktorsautentisering
- Backup och återställningsplaner

Principen bygger på att inget skydd är perfekt - men flera tillsammans kan skapa ett mycket starkare försvar.

### 9.26.3 Security by Design

Security by Design innebär att säkerhet inte ska vara något man lägger till efteråt - det ska vara en integrerad del från början.

Exempel:

- En webbapplikation bör redan från planeringsfasen inkludera skydd mot t.ex. SQL-injection.
- Ett operativsystem bör vara säkert i sitt standardläge, inte först efter flera konfigurationssteg.

Inom moderna utvecklingsmodeller, som DevSecOps, samarbetar utveckling, drift och säkerhet redan från start.

### 9.26.4 Fail Secure / Fail Safe

Hur ett system beter sig vid fel är en viktig säkerhetsfråga:

- Fail Secure: Systemet låser sig vid fel - skyddet går före tillgängligheten.
- Fail Safe: Systemet öppnas vid fel - tillgängligheten går före skyddet.

Exempel:

- Ett bankvalv ska vara fail secure.
- Ett brandskyddssystem ska vara fail safe - alla dörrar ska kunna öppnas vid brand.

Rätt princip måste väljas utifrån vad systemet skyddar.

### 9.26.5 Separation of Duties

Separation of Duties innebär att ansvar fördelar mellan flera personer för att minska risken för miss bruk eller misstag.

Exempel:

- En person får konfigurera backup.
- En annan får återställa data.

Ingen ska ha full kontroll över hela processen – varken i ekonomi, revision eller IT. Det minskar risken för fusk, fel eller sabotage, och ökar spårbarheten.

## 9.26.6 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?

## 9.26.7 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](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/>

## 9.26.8 Egna anteckningar

(Lämna utrymme för reflektion, minnesanteckningar eller egna exempel.)

## 9.27 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 framtidig IT-tekniker? Och vad väntar runt hörnet när det gäller hot, teknik och lagstiftning?

### 9.27.1 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.

### 9.27.2 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.

### 9.27.3 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 vetskaps (s.k. Shadow IT)

Tips: Se alltid till att det finns en balans mellan användarvänlighet och säkerhet – inte ett motsatsförhållande.

### 9.27.4 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?"

## 9.27.5 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.

Framtida lagstiftning

IT-säkerhet påverkas allt mer av internationella regelverk.

EU:s AI Act kommer att reglera hur AI får användas – även ur ett  
säkerhetsperspektiv.

NIS2-direktivet ställer högre krav på cybersäkerhet i samhällsviktig  
verksamhet.

Att känna till lagstiftningen är en del av yrkesansvaret.

## 9.27.6 Riktiga incidenter – vad kan vi lära oss?

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.

## **9.27.7 Sammanfattnings – 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.

## 9.27.8 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?

## 9.27.9 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/>

## 9.27.10 Egna anteckningar

(Lämna plats för reflektion, tankar och exempel från egna erfarenheter.)

© 2025 Robin Bräck  
Det här verket är skyddat av upphovsrätt. Se första sidan