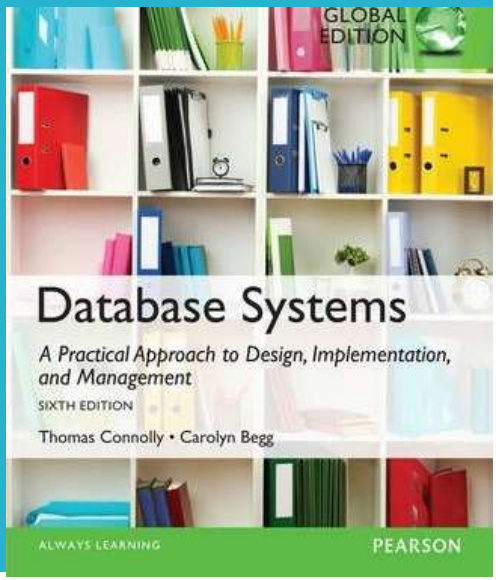
Leksjon 5:
Sikkerhet og databaseadministrasjon

Jarle Håvik
DAT2000 – Database 2, Campus Ringerike H20
Presentasjonen er basert på kapittel 20 i Connolly og Begg
I tillegg er det lagt ut en artikkel under Filer i Campus som omhandler MAC.



University of South-Eastern Norway
School of Business

Agenda

- Introduksjon
- Data administrator vs. Database administrator
- Database administrator i fokus
- Database administrasjonsverktøy i PostgreSQL
- Database sikkerhet
- DBMS & Web sikkerhet

Introduksjon

- Vi skiller mellom 2 roller for hvem som er ansvarlig for å styre og kontrollere aktivitetene som er forbundet med data i organisasjonen:
  - **Data administratoren (DA)** – har ansvar for de tidlige fasene i livssyklusen ved utviklingen av databasessystemet (planlegging -> logisk design)
  - **Database administratoren (DBA)** – er opptatt og har ansvar for det som kommer etter på (applikasjon/fysisk design -> operasjonelt vedlikehold)

Data Administrator vs. Database Administrator

	Data Administrator	Database Administrator
Ansvar	Har overordna ansvar for dataressursen, som omfatter planlegging-, utvikling og vedlikehold av standarder, forretningsregler og prosedyrer, samt konseptuelt og logisk databasedesign.	Ansvarlig for håndteringen av den fysiske realiseringen av databasesystemet. Dette omfatter blant annet det fysiske databasedesignet (mapping) og implementeringen av det. Videre har DBA-en ansvar for sikkerhetsinnstillinger (påloggінger/brukere/roller), integritetskontroller, overvåking av systemytelse og reorganisering av databasen når det er nødvendig.
Oppgaver	• Involvert i den strategiske it planleggingen • Fastsetter de langsiktige målene • Utarbeider standarder, forretningsregler (policies) og prosedyrer • Bestemmer data behovet – er den som er ansvarlig for kravspesifikasjonen • Utarbeider konseptuell og logisk database design • Utvikler og vedlikeholder virksomhetens datamodell • Koordinerer systemutviklingen • Ledelsesorientert • DBMS-uavhengig	• Evaluerer nye DBMS-er • Eksekverer planer for å nå målene • Håndhever standarder, forretningsregler (policies) og prosedyrer • Implementerer data heovene • Implements data requirements • Utvikler logisk og fysisk databasedesign • Monitorerer og kontrollerer databasen • Teknisk orientert • DBMS-avhengig

Database Administrator

- **Detaljerte oppgaver:**

Evaluating and selecting DBMS products

Undertaking physical database design

Implementing a physical database design using a target DBMS

Estimating volumes of data and likely growth

Determining patterns and frequencies of data usage

Defining security and integrity constraints

Liaising with database system developers

Developing test strategies

Training users

Responsible for 'signing off' the implemented database system

Monitoring system performance and tuning the database, as appropriate

Performing backups routinely

Ensuring recovery mechanisms and procedures are in place

Ensuring documentation is complete including in-house produced material

Keeping up to date with software and hardware developments and costs, and installing update necessary

Database Administrator oppgaver .

- DBA-en har ansvar for flere kritiske oppgaver:
- **Design av konseptuelle og fysiske skjema:**
  - DBA er ansvarlig for å samhandle med brukerne av systemet for å få forståelse for hvilke data som skal lagres i DBMS-et og hvordan de mest trolig kommer til å bli brukt.
  - DBA-en må på bakgrunn av denne kunnskapen designe det konseptuelle skjemaet for database (herunder bestemme hvilke relasjoner som skal lagres) og det fysiske skjema (hvordan de skal lagres).
  - DBA kan også designe mye brukte deler av det eksterne skjemaet, selv om brukerne sannsynligvis utvider dette skjemaet ved å lage flere views.

Database Administrator oppgaver ..

- **Sikkerhet og autorisering:**
  - DBA er er ansvarlig for å sikre at uautorisert datatilgang ikke er tillatt. Generelt sett bør ikke alle ha tilgang til alle dataene.
  - Dette løser en I DBMS ved at brukerne får tillatelse til noen bestemte views og tabeller.
    - For eksempel, selv om du kanskje tillater studenter å finne ut kursinnmeldinger og hvem som underviser i et gitt kurs, vil du ikke at studentene skal se ansattes lønn eller karakterinformasjon til andre studenter.
  - DBA kan håndheve dette ved å lage et eget kursInformasjonsview som kun henter data fra databasen, og så gi studentene tillatelse til dette viewet.

Database Administrator Task

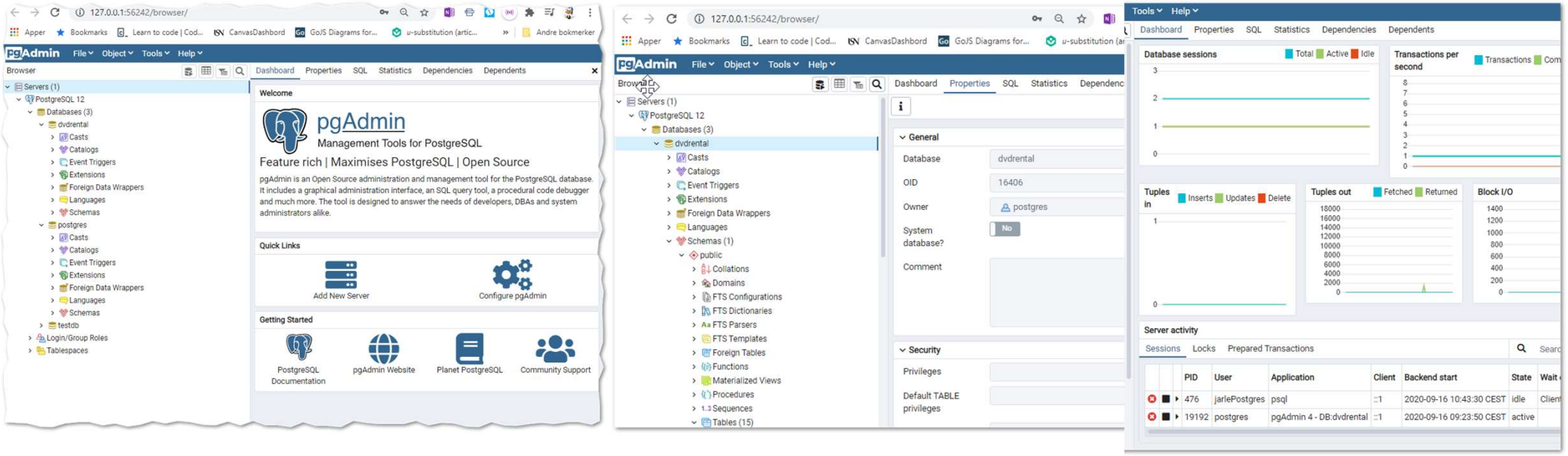
- **Datatilgjengelighet og gjennoppretting etter feil:**
  - DBA må ta nødvendig skritt for å sikre at om systemet feiler, så vil brukerne kunne fortsette å få tilgang til så mye av de uforstyrrede data som mulig. DBA må også arbeide for å gjenopprette dataene til en konsistent tilstand.
  - DBMS gir programvarestøtte for disse funksjonene, men det er DBA som er ansvarlig for å implementere prosedyrer for å sikkerhetskopiere dataene med jevne mellomrom og vedlikeholde logger over systemaktivitet (for å lette gjenoppretting etter et krasj).

Database Administrator oppgaver…

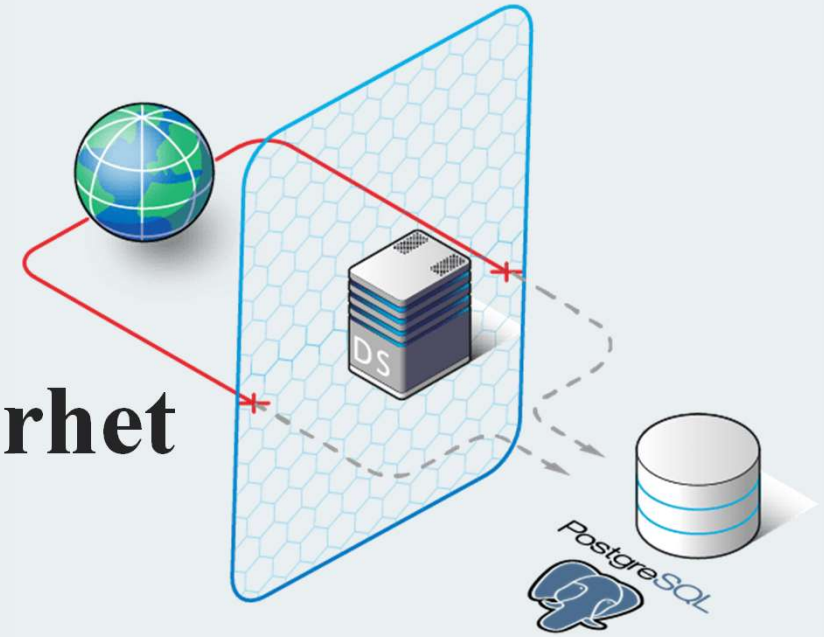
- **Finstilling - tuning av databasen:**
  - Brukernes behov vil sannsynligvis utvikle seg med tiden.
  - DBA er ansvarlig for å gjøre endringer i databasen, spesielt de konseptuelle og fysiske skjemaene, for å sikre tilstrekkelig ytelse etter hvert som kravene endres.

DBA verktøy i PostgreSQL

- pgAdmin (III/4) er den mest populære og mest funksjonsrike grafiske administrasjons- og utviklingsplattformen for PostgreSQL.



Databasesikkerhet



Database sikkerhet

- Data er en verdifull ressurs som må kontrolleres og administreres strengt, som med enhver bedriftsressurs.

- En del av eller alle bedriftsdataene kan ha strategisk betydning og må derfor beskyttes og holdes konfidensielle.

Database sikkerhet ..

Beskyttelsen av (eller mekanismer for beskyttelse) databasen mot uautorisert tilgang, enten forsettlig eller utilsiktet.

- Omfatter maskinvare, programvare, personer og data
- Har som mål å minimere tap forårsaket av forventede hendelser på en kostnadseffektiv måte
- Krever passende kontroller definert i systemets oppdragsmål

Hvorfor er dette viktig?

1) Økende datamengde lagret på datasystemer

2) Det faktum at tap eller utilgjengelighet av data kan være katastrofal

Tre hovedmål

1. **Hemmelighold(Secrecy):**

   Informasjon skal ikke avsløres til uautoriserte brukere.

   Eksempel: en student skal ikke få undersøke andre studenters karakterer.

2) **Integritet (Integrity)**:

   Bare autoriserte brukere skal ha lov til å endre data.

   Eksempel: studenter kan få se karakterene sine, men ikke tillatt å endre dem.

3) **Tilgjengelighet (Availability)**:

   Autoriserte brukere skal ikke nektes tilgang.

   For eksempel: en foreleser som ønsker å endre karakter, bør få gjøre det.

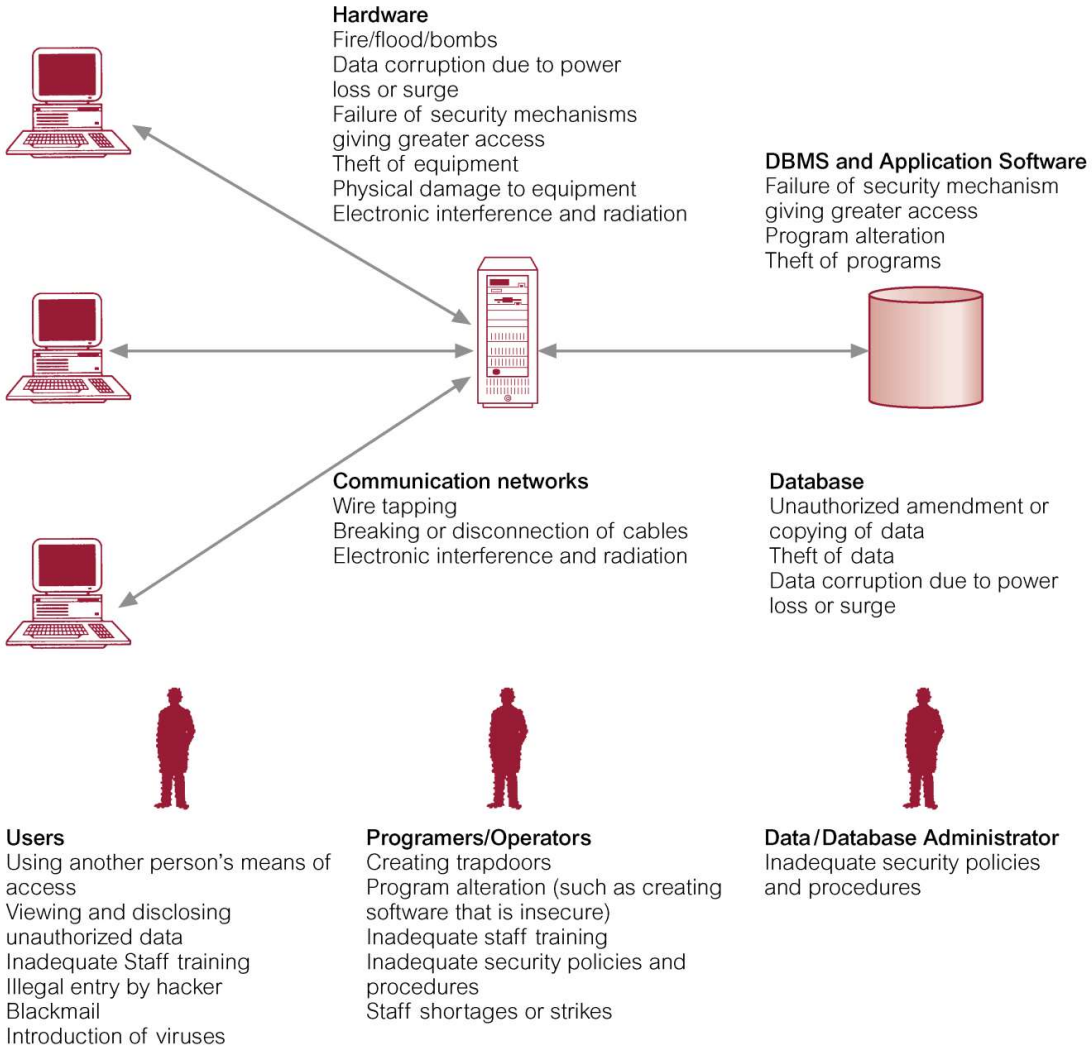
Databasesikkerhet

- Omfatter tiltak mot:
  - Tyveri og svindel
  - Tap av konfidensialitet (hemmelighold)
  - Tap av personvern (privacy)
  - Tap av integritet
  - Tap av tilgjengelighet

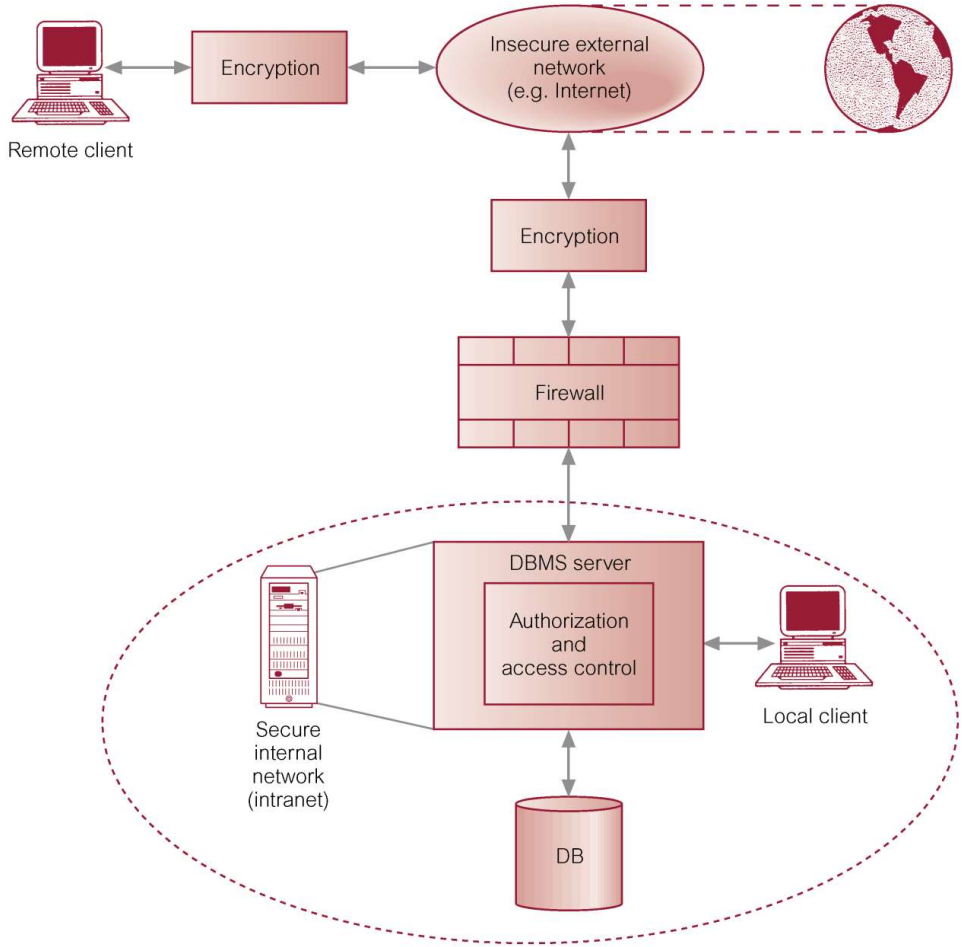
Eksempler på vanlige bruddsituasjoner

- Svindel eller tap av personvern trenger ikke føre til endringer i databasen.
- Tyveri og svindel påvirker databasemiljøet og hele organisasjonen, men trenger ikke å medføre endring av dataene. Det det derimot kan føre til er tap av konfidensialitet eller tap av personvern, som i sin tur kan svekke omdømmet til virksomheten.
  - Konfidensialitet fokuserer på å opprettholde hemmelighold av **kritiske data** til organisasjonen (tap av konfidensialitet fører til *tap av konkurranseevne*)
  - Personvern er opptatt av å beskytte **data om enkeltpersoner** (tap av personvern kan resulterer i at det blir tatt *søksmål mot organisasjonen*)
- Tap av integritet resulterer i ugyldige eller korrupte data (inkonsistente data) og kan få alvorlige konsekvenser for driften av en organisasjon.
- Tap av tilgjengelighet resulterer i at **dataene, eller systemet, eller begge deler ikke kan aksesseres**, og vil i sin tur kunne påvirke organisasjonens *økonomiske ytelse alvorlig eller forårsake datakorrupsjon*

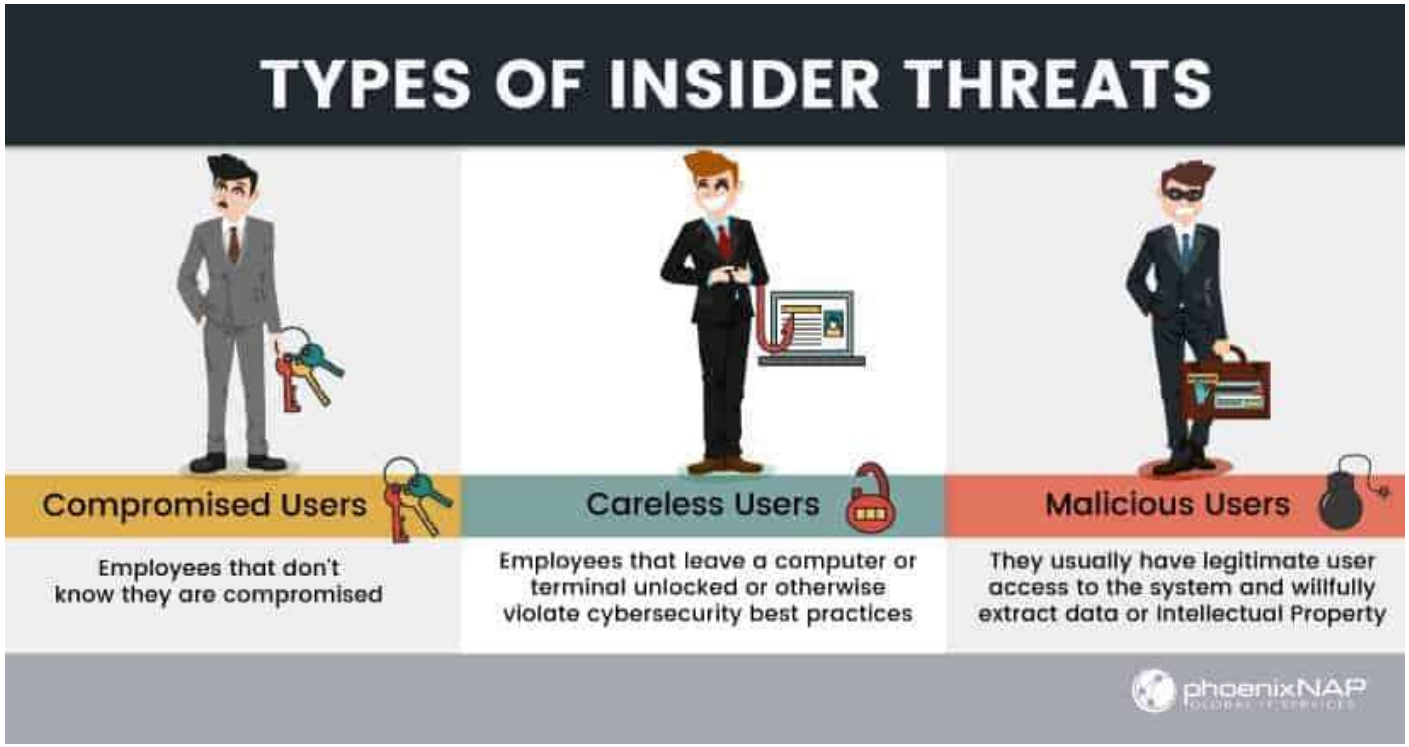
Oversikt over trusler mot datasystemer



Typisk fler-bruker datamaskinmiljø



Trusler fra egne ansatte



https://phoenixnap.com/blog/insider-threats

Databasesikkerhet

- Trussel
  - *Enhver situasjon eller hendelse, bevisst eller utilsiktet, som vil påvirke et system og følgelig en organisasjon negativt*
    - Håndgripelig (materiell) skade (tap av maskinvare, programvare eller data)
    - Immateriell skade (tap av troverdighet eller klienttillit)

Databasesikkerhet

- **Gjennoppretting fra en trusselsituasjon avhenger av**:
  - Om organisasjonen investerte tid og krefter i å identifisere muligheten for trusselen på forhånd eller ikke – føre var prinsippet
  - Etablering av mottiltak og beredskapsplaner i tilfelle en trusselsituasjon oppstår
  - Tiden som trengs for å gjenopprette systemet

- **Vurderingen av alvorlighetsgraden til en trussel er avhengig av**:
  - Typen virksomhet
  - Betydningen/Konsekvenser av trusselens innvirkning på organisasjonen

Mottiltak - Databasert kontroll

Forbundet med fysisk kontroll av administrative prosedyrer og inkluderer:

- Autorisasjon
- Tilgangskontroller
- Views
- Sikkerhetskopiering og gjenoppretting
- Integritet
- Kryptering

Mottiltak - Databasert kontroll

- Autorisering (Autorization)
  - Innvilgelsen av en rettighet eller et privilegium, som gjør det mulig for et subjekt å lovlig ha tilgang til et system eller et objekt i systemet.
  - Autorisasjon er en mekanisme som avgjør om en bruker er, hvem han eller hun hevder å være.

Authorization is the function of specifying access rights/privileges to resources, which is related to information security and computer security in general and to access control in particular.[1] More formally, "to authorize" is to define an access policy.

https://en.wikipedia.org/wiki/Authorization

Mottiltak - Databasert kontroll

- Tilgangskontroll (Access control)
  - Basert på tildeling (**GRANT**) og tilbakekalling (**REVOKE**) av privilegier.
  - Et privilegium gir en bruker rett til å opprette eller få tilgang til (som les (Read), skrive (Write) eller modifisere (modify)) gitte databaseobjekt (for eksempel en relasjon, view og index) eller å nytte visse DBMS-verktøy.
  - Privilegier gis til brukere for at de skal kunne utføre oppgavene som kreves i arbeidet deres.

Mottiltak - Databasert kontroll

- De fleste DBMS tilbyr en tilnærming kalt **D**iscretionary **A**ccess **C**ontrol (DAC)
- SQL standard støtter DAC gjennom kommandoene GRANT og REVOKE.
- GRANT-kommandoen gir privilegier til brukere, mens REVOKE tar de vekk.

Strukturen til GRANT-utsagnet:
GRANT < liste med privilegier, f.eks SELECT, INSERT, UPDATE, DELETE>
ON <relasjon eller et annet objekt>
TO < liste over autoriserte ID-er >
WITH GRANT OPTION; →
Brukeren som tildeles privilegiet kan gi det videre til andre.
Trenger da ikke gi ALLE privilegiene som er tildelt videre.

Strukturen til REVOKE-utsagnet:
REVOKE <liste med privilegier>
ON <relasjon eller et annet objekt>
FROM <liste over autoriserte ID-er>
CASCADE/RESTRIC; →
Tildelingen fra tilbakekalleren er ikke lenger gjeldende

Eksempel GRANT - REVOKE

- *Jarle har opprettet Bil-tabellen*

1. *GRANT SELECT ON* Bil *TO* Anne, *WITH GRANT OPTION* ***(utført av Jarle)***

2. *GRANT SELECT ON* Bil *TO* Halvor *WITH GRANT OPTION* ***(utført av Anne)***

3. *REVOKE SELECT ON* Bil *FROM* Anne *CASCADE* ***(utført av Jarle)***

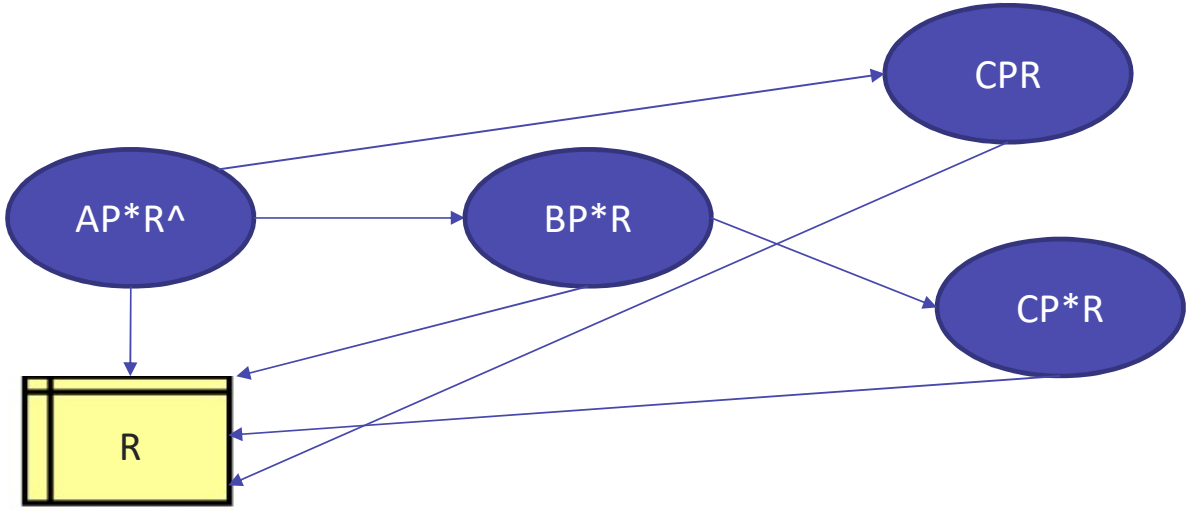
- Halvor sitt privilegium sies å være forlatt (abandoned) når privilegiet det ble hentet fra er fjernet. Gjelder når Cascade er satt. Se også at en har et ytterligere parameter som kan gis ved REVOKE – GRANT OPTION FOR. Les mer om det her.

Mottiltak - Databasert kontroll-eksempel

Bruker A gir (GRANTs) privilegium P på relasjon R til bruker B med WITH GRANT OPTION (*).

Notasjonen (AP * R ^) betyr at A eier R ^ der P * er et privilegium med WITH GRANT OPTION.

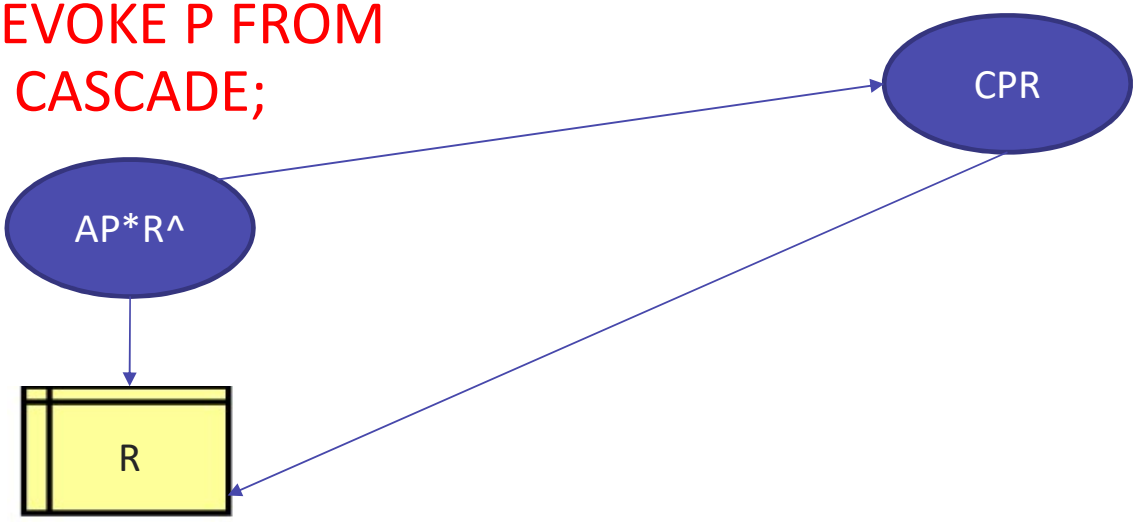
Deretter gir B P * til C samtidig med at også A gir privilegiet P til C.



Kan C gi privilegiet videre til en ny bruker?

Mottiltak - Databasert kontroll

A utfører:
REVOKE P FROM
B CASCADE;



Vil C nå kunne gir privilegiet videre?

Mottiltak - Databasert kontroll

- DAC har til tross for å vere effektiv en del svakheter. Spesielt ved at en uautorisert bruker kan lure en autorisert bruker til å avsløre sensitive data.
- En ekstra tilnærming kan være nødvendig og den kan oppnås ved en kontrollmekanisme kalt **Mandatory Access Control** - obligatorisk tilgangskontroll (MAC).

Mandatory Access Control .

- **MAC** kan implementeres ved å ta i bruk en populær modell kalt **Bell-LaPadula**, og i henhold til denne modellen vil:
  - Hvert databaseobjekt tildeles en sikkerhetsklasse, modellen har fire klasser:
    - Top Secret (**TS**),
    - Secret (**S**),
    - Confidential (**C**),
    - Unclassified (U).
  - klassene er rangert som **TS> S> C> U**, der data i klasse TS har et høyere sikkerhetsnivå enn S-klassedata osv.

Mandatory Access Control ..

- Hvert subjekt (les: bruker etc.) tildeles en godkjenning for en sikkerhetsklasse som gjelder lese- og skriverestriksjoner på databaseobjektet
  - Lese restriksjoner (Read): subjekt **S** kan kun lese objekt **O** dersom (**S**) >= klasse (**O**) → en bruker med sikkerhetsklarering **TS** kan lese et objekt med **S**, **C**, and **U** klasser, mens en bruker med **C** klarering kan ikke lese et objekt med **TS** class.
  - Skrive restriksjoner (Write): **S** kan skrive **O** berre om klasse (**S**) <= klasse (**O**) → en bruker med klarering **S** kan kun skrive objekter med **S** and **TS** klasser.
  - Sjå også artikkel om MAC som ligger under filer i Canvas.

Bell-LaPudula også på tupler

- **MAC** reglene gjelder for relasjoner, men også tupler! Anta at hver tuple er tildelt et sikkerhetsnivå, dette kalles en flernivåsrelasjon(multilevel relation

- **Eksempel:** Hvilke poster kan leses av brukere med TS-, S-, C- og U-sikkerhetsklaring, gitt følgende forhold på flere nivåer?

client No	f Name	l Name	tel No	pref Type	max Rent	security Class
CR76	John	Kay	0207-774-5632	Flat	425	**C**
CR56	Aline	Stewart	0141-848-1825	Flat	350	**C**
CR74	Mike	Ritchie	01475-392178	House	750	S
CR62	Mary	Tregar	01224-196720	Flat	600	S

Anta nå at en bruker med klaring C ønsker å legge inn en tupel - rad (CR74, David, Sinclaire) i Client-relasjonen. **Hva tror du ville skje ?!**

Bell-LaPudula - polyinstantiering

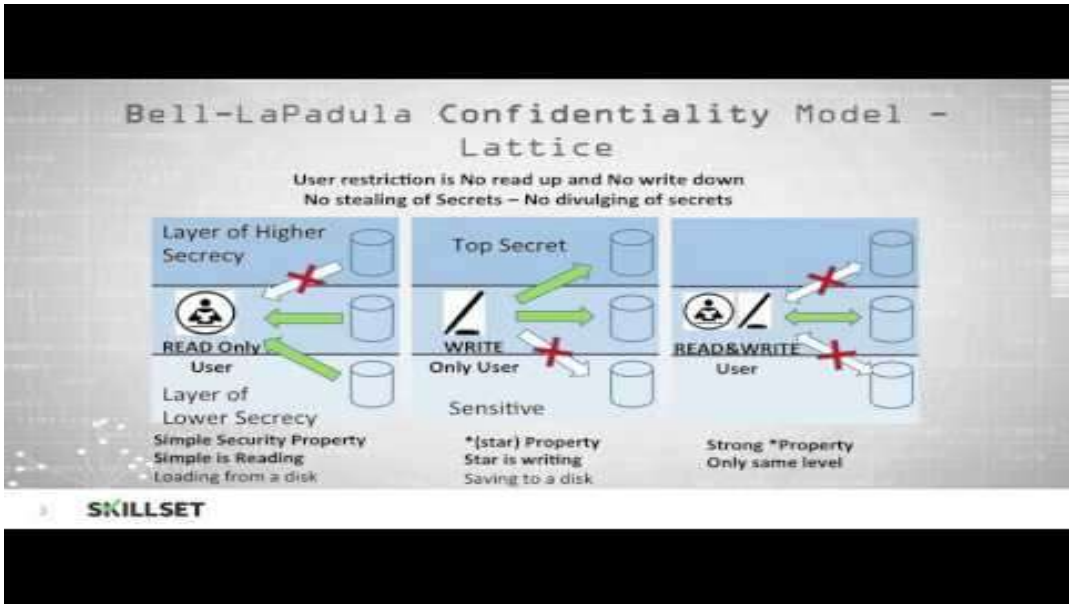
Å legge til posten bryter med primærnøkkelskranken, og den eksisterende posten har en høyere sikkerhetsklasse (S), men:

- Ved å betrakte securityClass som en del av primærnøkkelen clientNo, så kan den nye posten legges til relasjonen! → dette omtales som **polyinstantiation** (polyinstantiering).
  - *Tilstedeværelsen av dataobjekter som ser ut til å ha forskjellige verdier for brukere med forskjellige klareringer*

client No	F Name	L Name	Tel No	prefType	Max Rent	securityClass
CR76	John	Kay	0207-774-5632	Flat	425	**C**
CR56	Aline	Stewart	0141-848-1825	Flat	350	**C**
CR74	Mike	Ritchie	01475-392178	House	750	S
CR62	Mary	Tregar	01224-196720	Flat	600	S
CR74	David	Sinclaire	Blank	Blank	Blank	**C**

Hvilke poster kan nå leses av brukere med TS-, S-, C- og U-klaring?

Bell-La Pedula og Biba



Mottiltak - Databasert kontroll

- View
  - Er det dynamiske resultatet av en eller flere relasjonsoperasjoner som opererer på basis relasjonene for å produsere en ny relasjon.
  - Et view er en virtuell relasjon som ikke eksisterer faktisk i databasen, men som opprettes på forespørsel fra en bestemt bruker, på forespørselstidspunktet.

```
SELECT VIEW YoungSailors (sic, age, rating)
    AS SELECT S.sid, S.age, S.rating
    FROM Sailors S
    WHERE age < 18;
```

Mottiltak - Databasert kontroll

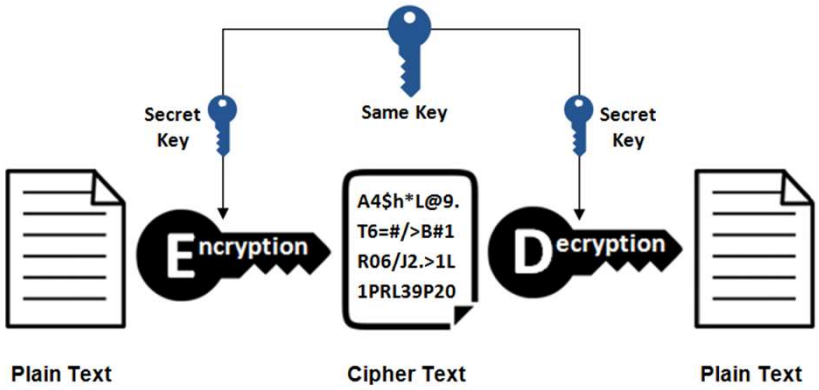
- Backup
  - Prosess med jevnlig å ta en kopi av databasen og loggfilen (og muligens programmer) til offline lagringsmedier.
- Journaling
  - Prosess med å oppbevare og vedlikeholde en loggfil (eller journal) over alle endringer i databasen for å muliggjøre effektiv gjenoppretting i tilfelle feil.

Mottiltak - Databasert kontroll

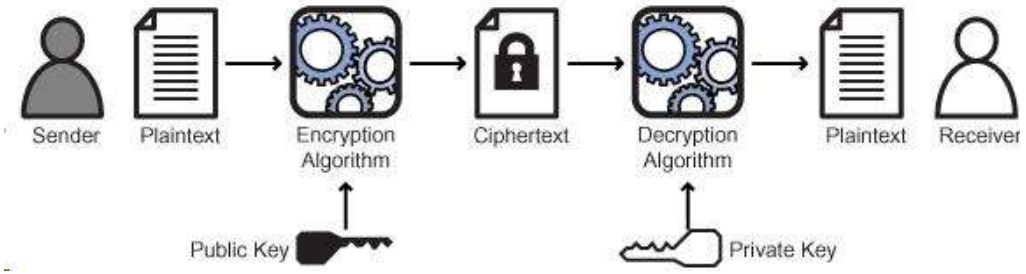
- Integritet
  - Forhindrer at data blir ugyldige, og dermed gir opphav til misvisende eller uriktige resultater.
- Kryptering
  - Kodingen av dataene med en spesiell algoritme som gjør dataene uleselige for ethvert program uten dekrypteringsnøkkelen.

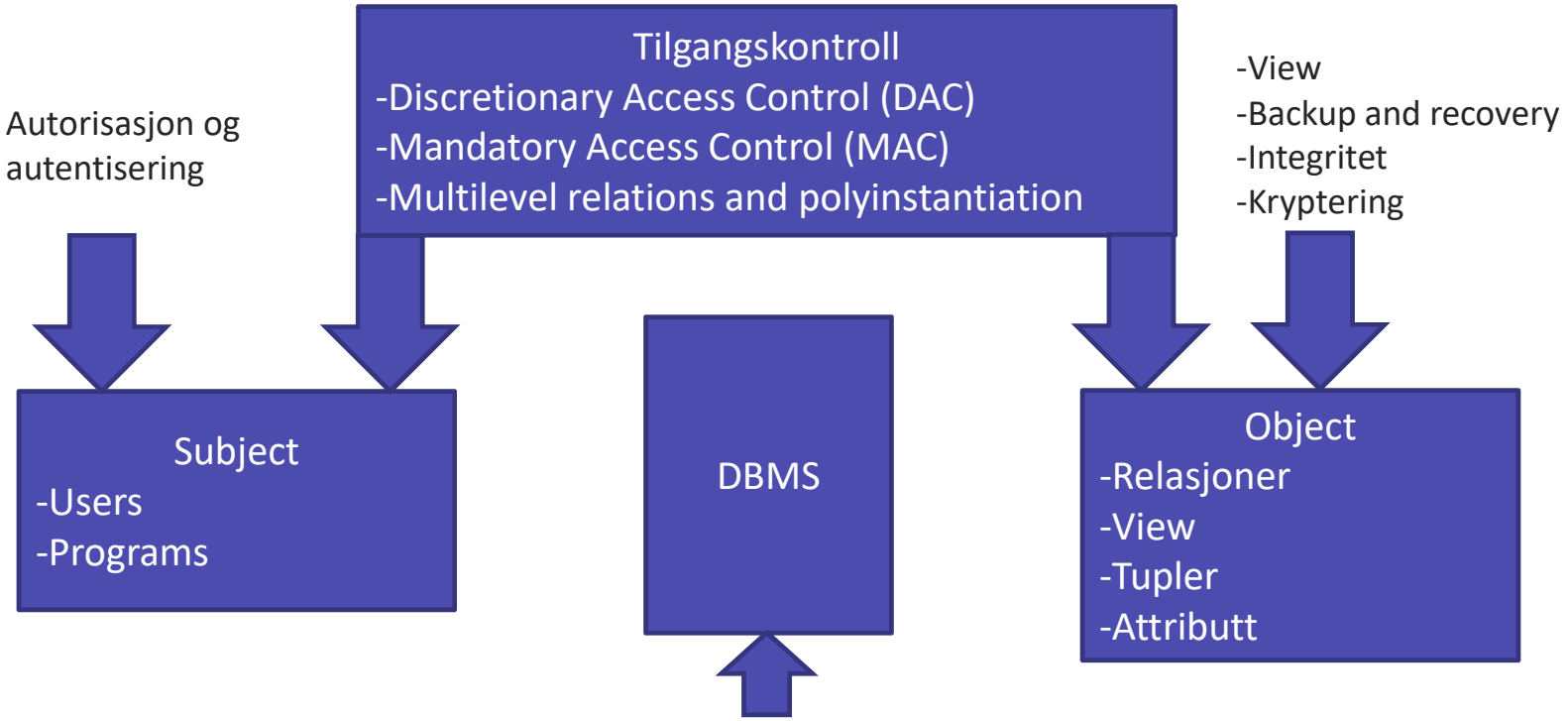
Kryptering: To krypteringssystem

Symetrisk kryptering (f.eks Data Encryption Standard [DES])



Asymetrisk kryptering «public key encryption» (f.eks RSA)





- Holder oversikt over hvordan privilegier tildeles / tilbakekalles til brukere
- Gir GRANT- og REVOKE-kommandoer til DBAer
- Gir krypteringsfasiliteter
- Bestemmer om en gitt bruker kan lese eller skrive på et gitt objekt basert på objektets sikkerhetsnivå (sikkerhetsklasser) og brukerens klarering

GRANT? REVOKE?

Sikkerhetsklasser? Klarering?

DBMS og nettsikkerhet
Cybersikkerhet

DBMS og nettsikkerhet

(1 av 3)

- Internett-kommunikasjon er avhengig av TCP/IP som den underliggende protokollen.
- Imidlertid ble TCP/IP og HTTP ikke designet med sikkerhet i tankene. Uten spesiell programvare reiser all internettrafikk ‘i det åpne’ og alle som overvåker trafikken kan lese den.

DBMS og nettsikkerhet

Må sikre at en under overføring av informasjon over Internett :

- Gjør den utilgjengelig for alle andre enn avsender og mottaker (personvern – privacy);
- ikke blir endret under overføring (integritet - integrity);
- mottakeren kan være sikker på at den kom fra avsenderen (ektehet - authenticity);
- senderen kan være sikker på at mottakeren er ekte (ikke-fabrikert – non-fabrication);
- avsenderen kan ikke nekte for at han eller hun sendte den (ikke-avvisning – non-repudiation).

DBMS og nettsikkerhet

(3 av 3)

- Tiltak omfatter:
  - Mellomtenere - Proxy servers
  - Brannmurer - Firewalls
  - Meldingsfordøyelsesalgoritmer og digitale signaturer (Message digest algorithms and digital signatures)
  - Digitale sertifikat - Digital certificates
  - Kerberos
  - SSL-Secure sockets layer og HTTPS -Secure HTTP
  - SET - Secure Electronic Transactions og SST-Secure Transaction Technology
  - Java sikkerhet
  - ActiveX sikkerhet
  - Salting

DBA & Sikkerhet

- **Audit trail** - loggen over oppdateringer med autorisasjons-ID (for brukeren som utfører transaksjonen) lagt til hver loggoppføring
  - Denne loggen er bare en mindre utvidelse av loggmekanismen som brukes til å gjenopprette fra krasj.
- DBA kan velge å føre en logg over alle handlinger, inkludert lesinger, utført av en bruker
  - Å analysere hvordan DBMS-et ble aksessert kan bidra til å forhindre sikkerhetsbrudd ved å identifisere mistenkelige mønstre før en inntrenger endelig lykkes med å bryte inn, eller det kan bidra til å spore en inntrenger etter at et brudd er oppdaget.
  - Her er en kort artikkel om de 10 mest vanlige database truslene:

    https://www.bcs.org/content-hub/top-ten-database-attacks/