

Andmebaaside turvalisuse tagamine

Teema 5

Andmebaasid II 2017

© Erki Eessaar



Organisatsiooni käsutuses olevad varad

- ♦ Füüsilised varad (nt seadmed, hooned).
- ♦ **Infovarad:**
 - Andmed ("andmed on uus nafta"),
 - IT aparatuur (riistvara, sideseadmed, toiteseadmed jms),
 - andmesidekanalid,
 - tarkvara.
- ♦ Toote valmistamise või teenuse andmise võime.
- ♦ Inimesed.
- ♦ Ainetud varad (maine, kuvand).

23.11.2017

Teema 5

2

Andmebaasid II 2017

© Erki Eessaar

Andmete asukoht

- ♦ Andmebaasisüsteemides loodud andmebaasides.
- ♦ Väljaspool andmebaase olevates failides (nii elektroonilised, kui paberil).
 - Seal on kuni 90% infost.
 - Näiteks asutuste dokumendiregistrid.
- ♦ Andmete kaitse tuleb tagada kõigis asukohtades ja järgnevad **üldpõhimõtted** kehtivad kõikjal.
- ♦ Tehniliste **näidete** osas keskendumine SQL-andmebaasisüsteemidele.

23.11.2017

Teema 5

3

Andmebaasid II 2017

© Erki Eessaar

Andmete puhul tuleb tagada

- ♦ **Konfidentsiaalsus (salastatus)**
 - Andmete konfidentsiaalsus on andmete kättesaadavus ainult selleks volitatud tarbijatele (isikutele või tehnilistele süsteemidele) ning kättesaamatus kõigile ülejäänutele.
- ♦ **Terviklikkus**
 - Andmete terviklikkus on andmete õigsuse/täielikkuse/ajakohasuse tagatus ning päritolu autentsus ja volitatute muutuste puudumine.

23.11.2017

Teema 5

4

Andmebaasid II 2017

© Erki Eessaar

Andmete puhul tuleb tagada (2)

- ♦ **Käideldavus**
 - Andmete käideldavus on kasutamiskõlblike andmete õigeaegne ja hõlbus kättesaadavus eelnevalt kokkulepitud vajalikul/nõutaval tööajal.
- ♦ **Revideeritavus**
 - Tuleb tagada andmetega tehtud toimingute järelvalve ja registreerimine.
 - Kõige lihtsam märgata käideldavuse probleeme.

23.11.2017

Teema 5

5

Andmebaasid II 2017

© Erki Eessaar

Käideldavusest – näide

- ♦ 19.08.2013 – Amazoni Põhja-Ameerika sait polnud **49 minutit** kättesaadav.
- ♦ Toimumata müükidest tingitud kahju suuruseks hinnati peaaegu **2 miljonit USD**.
- ♦ Samal ajal ka probleemid Amazoni pilveteenuste kasutajatel: Instagram, Netflix, Vine, Airbnb, Heroku, IFTTT.

23.11.2017

Teema 5

6

Konfidentsiaalsuse põhjused

- ♦ Andmeväärtuste loomuomane tundlikus (äri- või sõjasaladused).
- ♦ Andmete pärinemine tundlikust allikast.
- ♦ Andmete tundlikus varem avaldatu kontekstis.
- ♦ Andmete deklareerimine tundlikeks (vt isikuandmete kaitse).

Isikuandmete kaitse

- ♦ Kuni **24.05.2018** Eesti Vabariigi isikuandmete kaitse seadus
 - <https://www.riigiteataja.ee/akt/130122010011?leiaKehtiv>
- ♦ Alates **25.05.2018** asendab seda Euroopa Liidu isikuandmete kaitse üldmäärus (*GDPR – General Data Protection Regulation*)
 - <http://www.aki.ee/et/eraelu-kaitse/euroopa-andmekaitse-reform>
 - Ühtlustab korda Euroopa Liidu piires

Isikuandmete kaitse seaduste/määruste eesmärk

- ♦ Kaitsta isikuandmete töötlemisel *füüsilise isiku* (edaspidi isiku) põhiõigusi ja -vabadusi, eelkõige õigust **eraelu puutumatusele**.
- ♦ *Isikuandmed* on mis tahes andmed tuvastatud või tuvastatava füüsilise isiku kohta, sõltumata sellest, millisel kujul või millises vormis need andmed on.

Isikuandmete kaitse järelvalvaja

- ♦ Eestis tegeleb *Andmekaitse Inspeksioon*
 - <http://www.aki.ee/>
- ♦ Järgnev ülevaade põhineb Euroopa Liidu üldmäärusel
 - Ka enne seda kehtinud Eesti Vabariigi isikuandmete kaitse seadus oli eesrindlik ja sätestas paljuski samad põhimõtted.

Isikuandmete liigitus

- ♦ **Otsest** tuvastamist võimaldavad näiteks inimese nimi, sünniaeg, isikukood, e-posti aadress, elukoha aadress, kasutajatunnused, telefoni number, isiku foto.
- ♦ **Kaudselt** on isik võimalik tuvastada näiteks tervisliku seisundi, IP-aadressi, küpsiste, hüüdname, ostuvõime, majandusliku seisundi või auto andmete põhjal.

Eriliigilised isikuandmed

- ♦ Rassiline või etniline päritolu
- ♦ Poliitilised vaated
- ♦ Usulised või filosoofilised veendumused
- ♦ Ametiühingusse kuulumine
- ♦ Geneetilised andmed
- ♦ Biomeetrilised andmed
- ♦ Terviseandmeid
- ♦ Seksuaalelu ja seksuaalne sättumus

Töötlejal peab olema ametisaladuse hoidmise kohustus ja kindlad töötlemise eesmärgid

Andmebaasid II 2017 © Erki Eessaar



Euroopa Liidu isikuandmete kaitse üldmäärus

- Isikult tohib küsida ainult asjakohaseid andmeid, mis on vajalikud andmete töötlemise **eesmärgist** lähtudes.
- Isikuandmeid võib **säilitada** vaid niikaua, kuni nende kogumise algne **eesmärk** on täidetud.
- Turvalisuse tagamiseks tuleb rakendada asjakohaseid **tehnilisi** ja **korralduslike** meetmeid.
- Isikule tuleb esitada teavet tema isikuandmete töötlemise **tingimuste** ning tema **õiguste** kohta.

23.11.2017 Teema 5 13

Andmebaasid II 2017 © Erki Eessaar



Euroopa Liidu isikuandmete kaitse üldmäärus (2)

- Isik peab saama **küsida** andmete töötlejalt millised on tema kohta käivad andmed, kust andmed pärinevad, milleks andmeid kasutatakse, kes andmeid töötleb ning kellele töötleja võib andmeid edastada.
- Isikul on õigust nõuda oma isikuandmete töötlemise lõpetamist, ebaõigete isikuandmete parandamist ja kogutud isikuandmete sulgemist või **kustutamist** (**õigus olla unustatud**).

23.11.2017 Teema 5 14

Andmebaasid II 2017 © Erki Eessaar




Ülekantavusest

- Inimesel on õigus küsida ja saada andmetöötlejalt kõiki **teda puudutavaid isikuandmeid**, mida inimene on andmetöötlejale edastanud **nõusoleku** või **lepingu** alusel.
- Isiku kohta kogutud andmed peavad olema **korrapärased** selliselt, et neid oleks nõudmisel võimalik teisaldada ühest süsteemist teise.

23.11.2017 Teema 5 15

Andmebaasid II 2017 © Erki Eessaar




Ülekantavusest (2)

- Automatiseeritult töödeldavad** andmed tema kohta tuleb esitada isikule **struktureeritult** ja **valdavalt kasutatavas elektroonilises** (masinloetavas) vormingus.
- Kui see on **tehniliselt teostatav**, siis saab inimene nõuda, et üks andmetöötleja edastab andmed **otse teisele andmetöötlejale**.
 - Inimene peaks saama oma andmetega liikuda ühe teenusepakkuja juurest teise juurde.

23.11.2017 Teema 5 16

Andmebaasid II 2017 © Erki Eessaar



Euroopa Liidu isikuandmete kaitse üldmäärus (3)

- Tuleb määrata kindlaks, milliseid andmeid kogutakse **õigustatud huvi** ja milliseid **nõusoleku** alusel.
- Avaliku sektori asutustel ja suurematel andmetöötlejatel on kohustus määrata **andmekaitse spetsialist**.

23.11.2017 Teema 5 17

Andmebaasid II 2017 © Erki Eessaar



Euroopa Liidu isikuandmete kaitse üldmäärus (4)

- Isiku õiguseid ja vabadusi kahjustada võivatest **infoturbeidentidest** tuleb teavitada nii isikut kui järelevalveasutust.
- Isikuandmete töötlemisele tuleb kehtestada **protseduurid, dokumenteerida** isikuandmetega tehtud toimingud.
- Enne** isikuandmete töötlemise alustamist tuleb teostada andmekaitsealane **mõjuhindamine**.

23.11.2017 Teema 5 18

Andmebaasid II 2017 © Erki Eessaar

Euroopa Liidu isikuandmete kaitse üldmäärus (5)

- ♦ **Vaikimisi andmekaitse põhimõte** – andmesubjekti (isik, kelle andmeid töödeldakse) nõusolek andmete töötlemiseks peab olema **vaba tahte väljendus**, mitte vaikimisi eeldus.
 - Nõusolek tuleb anda iga andmete töötlemise eesmärgi ja viisi kohta eraldi.
 - Toodete ja teenuste **algseaded** peavad võimaldama inimesele **maksimaalset** kaitset.

23.11.2017 Teema 5 19

Andmebaasid II 2017 © Erki Eessaar

Euroopa Liidu isikuandmete kaitse üldmäärus (6)

- ♦ Süsteemide ülesehitamisel tuleb lähtuda **lõimitud andmekaitse** põhimõtetest.
 - Ennetav, mitte tagantjärele reageeriv
 - Andmekaitse on püsiseisund
 - Andmekaitse on süsteemi ülesehituse osa
 - Võitma peavad kõik osapooled
 - Turvalisus tuleb tagada andmete eluea algusest lõpuni
 - Andmetöötluse protsess peab olema nähtav ja läbipaistev
 - Kasutajate eraelu tuleb austada

23.11.2017 Teema 5 20

Andmebaasid II 2017 © Erki Eessaar


Euroopa Liidu isikuandmete kaitse üldmäärus (7)

- ♦ Trahvisumma maksimum on **20 miljonit** eurot või **4%** ettevõtte eelmise majandusaasta ülemaailmsest kogukäibest, kumb iganes on suurem.

23.11.2017 Teema 5 21

Andmebaasid II 2017 © Erki Eessaar

Olulisus andmebaaside arendajatele




- ♦ **Seaduse mitte tundmine ei vabasta seaduse täitmise kohustusest!!!**
- ♦ Kui arendate andmebaasi, milles hoitakse isikuandmeid, siis on isikuandmete kaitse seadus ja üldmäärus üheks nõuete allikaks.
 - Muuhulgas tuleb erilisel pöörata tähelepanu isikuandmete **turvalisuse** tagamisele.
 - **Minimaalsuse põhimõte** – piirdu andmete kogumisel ja kasutamisel sellega, mis on vajalik andmete kasutamise eesmärgi saavutamiseks.

23.11.2017 Teema 5 22

Andmebaasid II 2017 © Erki Eessaar

Olulisus andmebaaside arendajatele (2)



- ♦ Selleks, et isikuandmeid õigel ajal (mitte liiga vara ega hilja) **kustutada**, on vaja teada isikuga seotud olemite elutsükli muudatuste aegu (nt millal tellimus täidetuks märgiti, millal esitas klient kaebuse)
 - Seadused võivad kirjutada ette ajaperioodi, kui pikalt tuleb mingeid andmeid säilitada.
 - Andmebaas peab olema disainitud nii, et oleks võimalik kindlaks teha ajaperioodi algus ja lõpp.

23.11.2017 Teema 5 23

Andmebaasid II 2017 © Erki Eessaar

Kuidas kustutada isikuandmed anonümiseerides isikuga seotud andmed?

- ♦ Isik (isik_id, eesnimi, ...) Primaarvõti (isik_id);
- ♦ Tellimus (tellimuse_kood, tellija_id **DEFAULT -1,...**) Primaarvõti (tellimuse_kood) Välisvõti (tellija_id) Viitab Isik (isik_id) **ON DELETE SET DEFAULT**;

23.11.2017 Teema 5 24

Andmebaasid II 2017 © Erki Eessaar

Kommentaare

- ♦ Tabelis *Isik* rida anonüümse isiku kohta, kus **isik_id=-1**
- ♦ *tellij_id* veerg saab olla kohustuslik (NOT NULL).
- ♦ Tabelite *Isik* ja *Tellimus* ühendamisel pole vaja välisühendamist (*outer join*).

23.11.2017 Teema 5 25

Andmebaasid II 2017 © Erki Eessaar

Minu isikuandmete kasutamine riiklikes süsteemides


- ♦ Andmejälgija (päringud erinevatesse riiklikesse andmekogudesse)
 - <https://www.eesti.ee/et/turvalisus-ja-riigikaitse/turvalisus/isikuandmete-kasutamine/>
- ♦ **Revideeritavus** – kõigist päringutest jääb jälj
- ♦ **Konfidentsiaalsus** – kas minu andmeid kasutavad ainult need, kellel vaja?
- ♦ **Ennetusmeede** – võimalikel paharettidel on teada, et nende tegevus ei jää salajaseks

23.11.2017 Teema 5 26

Andmebaasid II 2017 © Erki Eessaar

Andmebaaside ohud

- ♦ Halvamine – teenuse **katkemine**
 - Andmed hävivad, muutuvad kättesaamatuks või kasutuskõlbmatuks.
- ♦ Infopüük – andmete lubamatu **lugemine**
 - Mittevõlgitatud osapool saavad andmeid lugeda.
- ♦ Andmete lubamatu **muutmine**
 - Modifitseerimine
 - Mittevõlgitatud osapool teeb olemasolevates andmetes muudatusi.
 - Võltsing
 - Fiktiivsete andmete lisamine.



23.11.2017 Teema 5 27

Andmebaasid II 2017 © Erki Eessaar

Andmebaaside ohud

- ♦ Ohud realiseeruvad läbi *nõrkuste* ehk *turvaaukude*, mis on kaitstava objekti suvalised nõrgad kohad, mille kaudu saab realiseeruda objekti varasid ähvardav oht.
- ♦ Ohuallikad.
 - Stiihilised ohud.
 - Ründed.

23.11.2017 Teema 5 28

Andmebaasid II 2017 © Erki Eessaar

Stiihilised ohud

- ♦ Keskkonna ohud
- ♦ Tehnilised rikked ja defektid
- ♦ Inimohud

23.11.2017 Teema 5 29

Andmebaasid II 2017 © Erki Eessaar

Ründed

- ♦ Ründed, mis lähtuvad inimestest, kes mingitel motiividel (nt materiaalsed, kättemaksuiha, tunnustuse vajadus) soovivad sihilikult kahju tekitada.
 - Füüsilised ründed
 - Ressursside blokeerimine (nt võrgu ülekoormamine)
 - Ressursside volitusteta kasutus
 - Infopüük, sh kommunikatsioonikanalite pealtkuulamine
 - ...

23.11.2017 Teema 5 30

Rünnaku põhjused

- ♦ *Kriminaalsed* (kuni 90%) – saada kasu.
- ♦ *Häktivism* – protesteerida, edastada sõnumit.
 - *Terrorism*.
- ♦ *Tööstusspionaaž*. Põhiliselt suurriikide- ja firmade pärusmaa.
- ♦ *Sõda*. Vahendeid ei loeta – kui vaja tehakse ära.

Ründemeetodid

- ♦ Füüsilised
 - Näide: Kangiga arvutiruumi uks lahti
- ♦ Sotsiaalsed
 - Näide: Kogutakse taustinfot inimese kohta, et parool ära arvata
 - Näide: Hirmvara abil, suhtlusosavust rakendades, kasutaja mingitele tegevustele suunamine
- ♦ Tehnilised
 - Näide: SQL süstimine

Tehniliste ründemeetodite näiteid

- ♦ Süsteemi ülekoormamine päringutega (teenusetõkestus)
- ♦ Pahatahtlikud programmid (kahjurvara) – viirused, ussid ja troojahobud
 - Näide aastast 2010 – *Stuxnet ussviirus*, mille sihtmärgiks on Siemensi loodud tööstuste jaoks mõeldud tarkvarasüsteemid.
- ♦ Pealtkuulamine
- ♦ Võltsimine, kellegi teise etendamine
- ♦ Konfidentsiaalsete andmete järeldamine mittekonfidentsiaalsetest

Tehniliste ründemeetodite näiteid (2)

- ♦ Süsteemi manipuleerimine sinna meelega või kogemata jäänud vigade/turvaaukude kaudu
 - Süsteemi manipuleerimine kasutades vaikimisi paroolidega kasutajakontosid
 - SQL süstimine
 - ...



Näide ründemeetodite kombinatsioonist

- ♦ Tehniline ja füüsiline rünne
- ♦ Sõrmejäljelugeja ära petmine sõrmejälje
 - kahedimensionaalse esitusega
 - elektrit juhtiva tindiga paberile trükitud sõrmejälje
 - kolmedimensionaalse esitusega
 - 3D prinditud
 - puust voolitud
 - lateksist tehtud

Andmebaasis võiks olla veerg, milles olevad väärtused on tüüpi *sõrmejälge*. Praktikas on need andmed veerus tüüpi BLOB või VARBINARY.



Tehnilise ründemeetodi näide – SQL süstimine

- ♦ Tehnika, mis kasutab süsteemi andmebaasikihi *haavatavust*.
- ♦ Haavatavus ilmneb, kui kasutaja poolt sisestatud andmeid pole piisavalt kontrollitud ning sisestatud sümbolid mõjutavad ootamatult SQL lause käitumist ja tulemust.

Tehnilise ründemeetodi näide – SQL süstimine (2)

- ♦ 2011. aasta CWE/SANS aruanne "Top 25 kõige ohtlikumat tarkvara viga" seab SQL süstimise ohtlikkuselt esikohale:
- ♦ <http://cwe.mitre.org/data/definitions/89.html>

Tehnilise ründemeetodi näide – SQL süstimine (3)

- ♦ https://en.wikipedia.org/wiki/SQL_injection#Examples
- ♦ 2002–2007 – 20% kõigist turvaaukudest seotud SQL süstimisega (<https://nvd.nist.gov/>)
- ♦ Pilvandetööstlust pakkuv ettevõtte *FireHost*:
 - 2012. aasta esimene kvartal – 277 770 blokeeritud SQL süstimise rünnet
 - 2012. aasta teine kvartal – 469 983 blokeeritud SQL süstimise rünnet
- ♦ Kasv võrreldes esimese kvartaliga 69%

Tehnilise ründemeetodi näide – SQL süstimine (4)

- ♦ 2012. aasta juunis laadis häkkerite rühmitus võrku 6.5 miljonit LinkedIn parooli (täpsemalt nende **SHA-1 abil leitud räsiväärtused**).
- ♦ 2012. aasta juulis postitas häkkerite rühmitus 453491 e-maili aadressi ja **avatekstina esitatud parooli**, mis pärinesid Yahoo! Voices teenuse andmebaasist.

Tehnilise ründemeetodi näide – SQL süstimine (5)

- ♦ 2014. aastal sai teatavaks, et kriminaalne võrgustik kogus 420000-lt veebilehelt kokku 1.2 miljardit unikaalset kasutajanime/parooli kombinatsiooni ning 542 miljonit e-maili aadressi.
 - Pahavaraga nakatatud kasutaja arvuti testib veebilehe külastamisel, kas seda saab SQL süstimise abil rünnata. Kui saab, siis see info läks kriminaalidele, kes hiljem lehte ründasid.

SQL süstimine (PostgreSQL)

- ♦ CREATE TABLE Kasutaja (kasutajanimi VARCHAR(50) NOT NULL PRIMARY KEY, parool VARCHAR(50) NOT NULL);
- ♦ INSERT INTO Kasutaja (kasutajanimi, parool) VALUES ('mikk','seen');
- ♦ INSERT INTO Kasutaja (kasutajanimi, parool) VALUES ('tabel','karu');

SQL süstimine (PostgreSQL) (2)

```
CREATE OR REPLACE FUNCTION f_on_kasutaja(kasutajanimi Kasutaja.kasutajanimi%TYPE,
parool Kasutaja.parool%TYPE) RETURNS BOOLEAN AS $$
DECLARE --Muutujate deklaratsioon
    sql_lause TEXT;
    ridade_arv INTEGER;
    otsus BOOLEAN;
BEGIN
    sql_lause:='SELECT * FROM Kasutaja WHERE kasutajanimi=''' || kasutajanimi || ''' AND
parool=''' || parool || '''';
    RAISE NOTICE 'Täidetav lause: %', sql_lause;
    EXECUTE sql_lause; --Käivitati päring
    /*Leiti päringu leitud ridade arv*/
    GET DIAGNOSTICS ridade_arv = ROW_COUNT;
    IF ridade_arv=0 THEN
        otsus=false;
    ELSE
        otsus=true;
    END IF;
    RETURN otsus; --Tagastan otsuse
END;
$$ LANGUAGE plpgsql SECURITY DEFINER
SET search_path = public, pg_temp;
```

Ärge kasutage sellist funktsiooni!!!

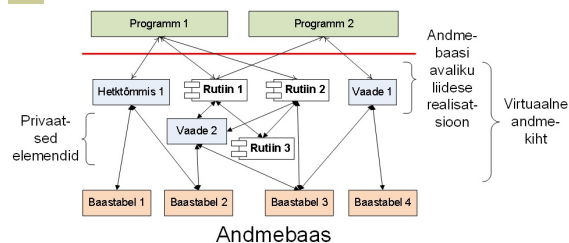
SQL süstimine (PostgreSQL) (3)

- ♦ `SELECT f_on_kasutaja ('mikk', 'seen');`
 - NOTICE: Täidetav lause: `SELECT * FROM Kasutaja WHERE kasutajanimi='mikk' AND parool='seen'`
 - Result: true
- ♦ `SELECT f_on_kasutaja ('a', 'b');`
 - NOTICE: Täidetav lause: `SELECT * FROM Kasutaja WHERE kasutajanimi='a' AND parool='b'`
 - Result: false

SQL süstimine (PostgreSQL) (4)

- ♦ `SELECT f_on_kasutaja ('a', 'b' OR 1=1;--');`
 - NOTICE: Täidetav lause: `SELECT * FROM Kasutaja WHERE kasutajanimi='a' AND parool='b' OR 1=1;--'`
 - `FALSE AND FALSE OR TRUE => FALSE OR TRUE => TRUE`
 - Eeldus – kasutajate tabelis on vähemalt üks kasutaja
 - Lõbusas võtmes selgitus: <https://xkcd.com/327/>

SQL süstimise rünnete tõrje



Andmebaas

Kasutage andmebaasiserveris talletatud rutine. Ärge kasutage andmebaasiserveris talletatud rutiinides *dünaamilist SQLi*. Seadke igale rakenduse kasutajale vastavusse eraldi andmebaasi kasutaja ning andke igale andmebaasi kasutajale tööks vajalik *minimaalne* hulk õiguseid.

SQL süstimise rünnete tõrje – ei kasuta dünaamilist SQLi

```
CREATE OR REPLACE FUNCTION f_on_kasutaja2(knimi Kasutaja.kasutajanimi%TYPE,
prl Kasutaja.parool%TYPE) RETURNS BOOLEAN AS $$
DECLARE
    sql_lause TEXT;
    ridade_arv INTEGER;
    otsus BOOLEAN;
BEGIN
    ridade_arv:=0;
    SELECT INTO ridade_arv Count(*) AS arv FROM Kasutaja WHERE kasutajanimi=knimi AND parool=prl;
    IF ridade_arv=1 THEN
        otsus=true;
    ELSE
        otsus=false;
    END IF;
    RETURN otsus;
END;
$$ LANGUAGE plpgsql SECURITY DEFINER
SET search_path = public, pg_temp;
```

Pakutud lahenduses on endiselt probleem – parool on andmebaasis **avatekstina**.

SECURITY DEFINER ja turvalisus

- ♦ SECURITY DEFINER – funktsioon käivitatakse selle looja õigustega.
- ♦ `SET search_path = public, pg_temp;`
 - See tagab, et skeemist `pg_temp` otsitakse skeemiobjekte ainult siis, kui neid skeemist `public` ei leita (ilma selle määranguta otsitakse neid skeemist `pg_temp` kõige esimesena).
 - Skeemi `pg_temp` saavad andmeid kirjutada kõik soovijad ja ründajad võivad selle ära kasutada.


Tehnilise ründemeetodi näide – paroolikaitse rünne

- ♦ Ründaja poolt genereeritud paroolide katsetamine
 - Sõnastikrünn on jõurünn, mis parooli mõistatamiseks proovib suurest ammenavast loendist võetud sõnu või nende kombinatsioone
- ♦ Õige parooli volitamatu hankimine
 - Näiteks kui parool on rakenduse koodis, mis omakorda versioonihalduse süsteemis.
 - Lahenduseks parooli hoidmine väljaspool rakendust, eraldi failis.

Tehnilise ründemeetodi näide – paroolikaitse rünne (2)

- ♦ Vaikeparooli kasutamine (inimoht – andmebaasi administraator võib olla laisk):
 - Näited Oracle andmebaasis
 - sys/CHANGE_ON_INSTALL
 - scott/tiger

Paroolikaitse rünnete tõrje

- ♦ Hoia parool **salajas**. Kolm inimest suudavad saladust pidada vaid siis, kui kaks on surnud. 
- ♦ Inimeste **koolitus**, et nad ei kasutaks kergesti äraarvatavaid paroole.
 - 1234, 1, 123, 12345, 123456, password, passwd, test
 - <https://majandus24.postimees.ee/2981145/need-on-20-maailma-halvimat-salasona>
- ♦ Süsteemipoolne parooli **tugevuse kontroll** ning nõue muuta regulaarselt parooli.

Paroolikaitse rünnete tõrje (2)

- ♦ Süsteem nõuab esimesel logimisel parooli **muutmist**.
 - Tuleb registreerida info, et järgmisel sisselogimisel vajab parool muutmist
- ♦ Süsteem lubab ebaõnnestunud sisselogimisi järjest piiratud **arv kordi** – peale seda konto lukustatakse.
 - Tuleb registreerida info, et konto on lukustatud

Paroolikaitse rünnete tõrje (3)

- ♦ Andmebaasisüsteem **lukustab** vaikimisi mõningad kontod (nt *scott*) ning andmebaasi administraator peab need eraldi käsuga avama.
- ♦ Andmebaasis hoitakse paroolide **soolatud räsiväärtuseid** (leitud räsifunktsiooniga, mida pole lihtne jõuga murda), mitte nende avateksti.
- ♦ Parooliks *salasõna* asemel **salalause** – pikem, lihtsam meeles pidada.

Paroolikaitse rünnete tõrje (4)

- ♦ Muuda parool raskesti äraarvatavaks, kasutades paroolina kahte neljakohalist *rõumuvat* lauset (luuletust)

Tehnilise ründemeetodi näide – järelaluslikud ründed

- ♦ Püüd tuletada konfidentsiaalseid andmeid mittekonfidentsiaalsetest.
 - **Otsene** rünne. Täpse päringu koostamine, mis tagastab andmed ühe objekti kohta.
 - **Kaudne** (statistiline) rünne. Tulemuse järeldamine statistilistest andmetest paljude objektide kohta.

Tehnilise ründemeetodi näide – järelaluslike ründed (2)

- ♦ Keelatud päringud.
 - `SELECT palk FROM Statistika WHERE sugu='M' AND amet='lasteaia kasvataja';`
- ♦ Lubatud päringud.
 - `SELECT Count(*) AS arv FROM Statistika WHERE sugu='M' AND amet='lasteaia kasvataja';`
 - Tulemus – 1
 - `SELECT Sum(palk) AS summa FROM Statistika WHERE sugu='M' AND amet='lasteaia kasvataja';`
 - Tulemus – 700

Järelduslike ründete tõrje

- ♦ Tõkestamine.
 - *Lävepõhine* tõkestus. Tulemuse väljastamiseks peab ridade arv, mille põhjal see on leitud, jääma mingisse vahemikku.
 - Päringute täitmise otsustamine, kasutades päringute ajalugu.
- ♦ Moonutamine.
 - *Juhuvaimi* kasutamine baasist.
 - Andmebaasist leitud väärtused moonutatakse väikese *juhuvea* lisamisega.

Turvameetmed

- ♦ Eesmärk on vähendada varadele mõjuvaid ohtusid ja seeläbi vähendada turvariski.
- ♦ Võimalik liigitus (rakendamise aja/funktsiooni järgi).
 - Profülaktika e ennetustöö
 - Tugevdusmeetmed – tugevda vaimu
 - Peletusmeetmed – hirmuta vastast
 - Eraldusmeetmed – tugevda keha
 - Tehtud pahanduse avastamine
 - Pahanduse tulemuste likvideerimine

Võrdle haiguste ennetamisega, diagnoosimisega, ravimise / leevendamise! Haigus – turvarike Ravim – turvameede

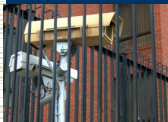


Turvameetmete funktsioonid

- ♦ Profülaktika ehk ennetustöö
 - Tugevdusmeetmed
 - Korra kehtestamine
 - Töötajate turvateadlikkus
 - Normaalsed töötingimused
 - Süstemaatiline kontroll
 - Viirustõrje
 - Peletusmeetmed, et kahandada ründete üritamise tõenäosust
 - Sanksioonid
 - Hoiatav märgistus
 - Nähtavad turvameetmed



Turvameetmete funktsioonid (2)



- ♦ Profülaktika ehk ennetustöö
 - Tõkestus- ja eraldusmeetmed:
 - Ruumiline eraldamine. Näiteks arvutite ja andmekandjate hoidmine erinevates asukohtades.
 - Pääsu reguleerimine. Objektide kasutamise valikuline võimaldamine subjektidele.
 - Salastamine, krüpteerimine.
 - Hävitamine. Teabe usaldusväärne kõrvaldamine.

Tugevdus-, peletus- ja tõkestusmeetmed

- ♦ 10% inimestest ei varasta/peta/valeta kunagi, 10% teevad seda alati, ülejäänute puhul see sõltub.
- ♦ Tugevdus-, peletus- ja tõkestusmeetmed aitavad ausatel inimestel ausaks jääda.
- ♦ <https://blog.codinghorror.com/the-just-in-time-theory/>

Andmebaasid II 2017 © Erki Eessaar

Turvameetmete funktsioonid (3)

- ♦ Turvarikete tuvastamine
 - Operatiivtuvastus – kohene reageerimine
 - Järeltuvastus – kohene registreerimine, et hiljem reageerida
 - Tõendtuvastus – range ja täpne registreerimine, et kehtiks tõendina ka nt kohtus.
- ♦ Infoobjekti turvalisuse ja rikke-eelse oleku taastamine (varundamine, taastamine, asendamine)

Võrdle haiguste diagnoosimisega!

Võrdle haiguste ravimisega!

23.11.2017 Teema 5 61

Andmebaasid II 2017 © Erki Eessaar

Turvameetmete rakendamine

- ♦ Füüsilised turvameetmed
 - Infrastruktuur, füüsilised piirded, valveseadmed, ...
- ♦ Organisatsioonilised turvameetmed
 - Töökorraldus
- ♦ Infotehnilised turvameetmed:
 - riistvaralised,
 - tarkvaralised.

23.11.2017 Teema 5 62

Andmebaasid II 2017 © Erki Eessaar

Turvasüsteemi projekteerimine

- ♦ Turvasüsteemi modelleerimine
- ♦ Ehitamine/hankimine
- ♦ Õiguste jagamine, administreerimine
- ♦ Kaitse tagamine
- ♦ Monitooring
- ♦ Protsessi suunab **turvapoliitika**

23.11.2017 Teema 5 63

Andmebaasid II 2017 © Erki Eessaar

Turvapoliitika

- ♦ (Info)turvapoliitika on eeskirjade, juhiste ja menetluste kogum, mis suunavad varade (peamiselt infovarade) haldust, kaitset ja jaotamist organisatsioonis ning selle IT süsteemides.
 - Näide: <http://kodu.ut.ee/~mroos/turve/turvapoliitika/>

23.11.2017 Teema 5 64

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid

- ♦ Kaitse **mitmekesisus** – mitte loota ühte sorti ja ühest allikast pärit vahenditele.
- ♦ Kaitse **sügavuti** – kasuta mitut järjestikku asuvat kaitsemehhanismi.
- ♦ Anna kasutajatele **vähim võimalik hulk õiguseid**, millega nad saavad oma töö tehtud.
- ♦ **Jälgi** ning **registreeri** süsteemis toimuvat.

23.11.2017 Teema 5 65


Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (2)

- ♦ Vähenda ründaja **motivatsiooni**, mõjutades tema ootuseid rünnaku tulemuste suhtes.
- ♦ <https://technet.microsoft.com/en-us/library/hh278941.aspx#EGAA> (10 muutumatut turvareeglit)
- ♦ **Kontrollitud keerukus** (standardite, reeglite, mustrite jms järgimine) võib muuta rünnaku korraldamise lihtsamaks, sest ründaja teab, mida oodata.

23.11.2017 Teema 5 66

Andmebaasid II 2017 © Erki Eessaar



Kontrollitud keerukuse problemaatika näide

- „E-valimistel korraldatakse rituaalseid turvaprotseduure, mis oluliselt ei suurenda tulemuse usaldusväärsust, kuid mõjuvad hästi asjatundmatule vaatlejale. Samuti ei ole protsessi jälgivale audiitorile antud ülesannet jälgida protsessi turvalisust, vaid juhendist kinnipidamist. See annab omakorda siseründajale võimaluse täpselt planeerida vaatlejatele kuvatavad ekraanipildid, mis ei pruugi kajastada tegelikult arvutis toimuvat. Kuigi serveri ja andmekandjate pakendid avatakse vaatlejate juuresolekul, ei välista see võimalust kompromiteerida seadmeid ja meediat enne pakendamist. Seega, kui audiitoril oleks õigus ja kohustus kontrollida turvalisust, siis peaks ta paluma administraatoritel teha ootamatuid toiminguid, mis veenaksid, et tegu ei ole n-õ valimismeepotiga – ettevalmistatud keskkonnaga vaatlejate petmiseks.“ (Agu Kivimägi, Postimees, 17.11.2013)

23.11.2017 Teema 5 67

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (3)

- Eelista **avatud lahendusi**. Tarkvaraliste turvasüsteemide disainipõhimõtted võiksid olla **avalikud** ja realiseeritsoon **avatud lähtekoodiga** tarkvara.
 - Nõrkused leitakse kiiremini üles ja need mõjutavad väiksemat hulka süsteeme.
 - Salatsemine ei kaitse nõrkuste ilmnemise eest, kuid mõjutatud süsteeme võib siis olla juba väga palju (vt Eesti ID-kaarti juhtumit).

23.11.2017 Teema 5 68

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (4)

- Kerckhoffs'i printsiip**. Krüptosüsteem peab säilitama turvalisuse ka siis kui kõik muu (sh tööpõhimõtte) peale **salajaste võtmete** on teada.
- Iga turvasüsteem sõltub sellest, et **mõned** asjad on **salajased** nagu näiteks
 - krüptosüsteemis salajased võtmed,
 - paljud volitustõendid.

23.11.2017 Teema 5 69

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (5)

- Kerckhoffs'i printsiibi üldistus**: Mida **vähem** asju tuleb süsteemi turvalisuse huvides **tingimata** salajas hoida, seda lihtsam on süsteemi turvalisust säilitada.
 - Kavanda süsteem lähtuvalt eeldusest, et ründaja teab selle tööpõhimõtet.
- Iga **tõeline saladus** on ühtlasi **nõrkus**, mille kaudu süsteemi rünnata.

23.11.2017 Teema 5 70

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (6)

- Ürita luua süsteeme, mille allsüsteemid **ei sõltu** üksteisest **liiga palju** (täielik sõltumatus pole võimalik). Paha kui:
 - ühe allsüsteemi riknemine peatab kogu süsteemi töö,
 - ühe allsüsteemi turvalisuse kompromiteerimine kompromiteerib terve süsteemi turvalisust.
- Ei ole rumalaid kasutajaid, on vaid **tooted**, mis lasevad kasutajatel teha **rumalusi**.

23.11.2017 Teema 5 71

Andmebaasid II 2017 © Erki Eessaar

Turvalisuse tagamise strateegiaid (7)

- Hooli inimestest ja motiveeri neid**.
 - Uuring aastast 2013: Eesti töötajatest teeb oma tööd täie pühendumusega vaid 16 protsenti töötajaid, 64 protsendil on tööst täiesti ükskõik ning *iga viies töötab tööandjale aktiivselt vastu*.
 - Ükskõiksus töö vastu – vähene motivatsioon tagada turvalisust
 - Tööandjale aktiivne vastutöötagamine – võib viia rünnakutele kaasaaitamiseni. Näide – rünnak Sony Pictures vastu 2014. aasta lõpus.

23.11.2017 Teema 5 [Edasi](#) 72

Riistvaralised turvameetmed – liiasus kaitsevahendina

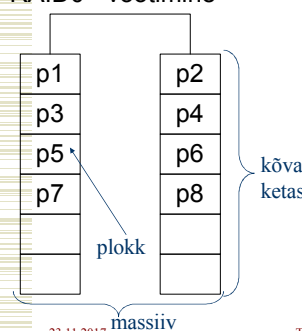
- ♦ Vahendeid *käideldavuse* tagamiseks
 - Arvutite klastrid
 - Arvuti riistvara komponentide (protsessor, kõvaketas, jahuti) dubleerimine

Sõltumatute (odavate) ketaste liiasmassiiv (RAID)

- ♦ Töökindlus – peegeldamine
- ♦ Töökiirus – paralleeltöö
 - **Hargsalvestus (võõtimine)** – mitu ketast moodustavad ühe loogilise salvestusüksuse.
 - Ketta salvestusruum jaotatakse lõikudeks ja ühisel virtuaalsel salvestuspinnal on vaheldumisi lõigud erinevatelt ketastelt.
- ♦ Erinevad konfiguratsioonid: RAID0 – RAID6, lisaks veel nende kombinatsioonid.

RAID0

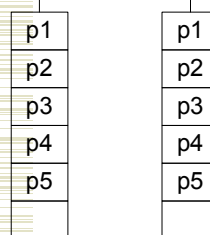
RAID0 - võõtimine



- ♦ Andmed **mitteliiaselt** jaotunud üle kõvaketaste.
- ♦ Massiiv rikneb, kui *kasvõi üks* kõvaketas rikneb.
- ♦ Töökindlus väiksem kui üksikul kettal – iga ketas suurendab vea tekkimise tõenäosust.
- ♦ Kui kettad erineva suurusega, siis on maht piiratud väikseima ketta suurusega.

RAID1

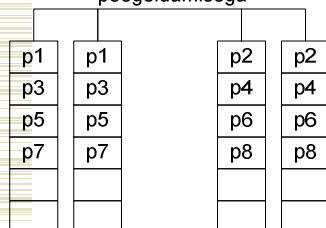
RAID1 – plokkide peegeldamine



- ♦ Igal kõvakettal andmete *täiskoopia*.
- ♦ Massiiv rikneb, kui *kõik* kõvakettad riknevad.

RAID 10

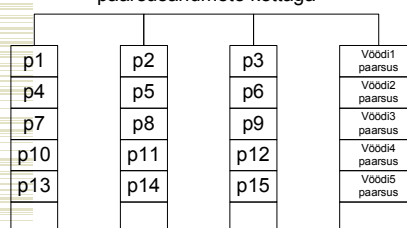
RAID 10 - võõtimine koos peegeldamisega



- ♦ Tuntakse ka kui RAID 1+0
- ♦ Andmete kaotamiseks, kui riknevad kõik ühte massiivi kuuluvad kettad

RAID4

RAID4 – võõtimine koos paarsusandmete kettaga



Paarsusandmed aitavad vältida võõtimisest tulenevat töökindluse vähenemise probleemi.

RAID4 – paarsusandmed

- ♦ Igale kettale mahub ühepalju plokke.
- ♦ Igas positsioonis on üle kõikide ketaste ühtede arv **paarisarv**.
- ♦ Ketas 1: 1|0|1|1|0|0|1|0
- ♦ Ketas 2: 0|1|0|0|1|1|0|0
- ♦ Ketas 3: 0|0|1|0|0|1|0|1
- ♦ Paarsusandmetega ketas: 1|1|0|1|1|0|1|1

RAID4 – ühe ketta andmete taastamine

- ♦ Ketas 1: 1|0|0|0|1|0|0|1
 - ♦ Ketas 2: ?|?|?|?|?|?|?|?
 - ♦ Ketas 3: 1|0|1|0|1|1|1|0
 - ♦ Paarsusandmetega ketas: 0|0|0|1|0|0|0|1
- 0, kui ketastel vastavas positsioonis **paarisarv** ühtesid, 1 kui **paaritu**.
- ♦ Ketas 2: 0|0|1|1|0|1|1|0

RAID4 – probleem

- ♦ Probleem: Mida teha siis, kui andmekettal andmed muutuvad? Näiteks muutub bait nr n
 - Lahendus 1: Lugeda kõikidelt andmeketastelt baidid nr n ja arvutada paarsusandmed uuesti.
 - Tagajärjed: Põhjustab suure hulga lugemisoperatsioone ja mõjub halvasti töökiirusele.
 - Lahendus 2: Leida millised positsioonid on baidis nr n muutunud ja muuta vastavaid positsioone paarsusandmetes.

Lahendus 2 – näide

- ♦ Ketas 1: 1|0|1|1|0|0|1|0
- ♦ Ketas 2: ~~0|1|0|0|1|1|0|0~~ 10000100
- ♦ Ketas 3: 0|0|1|0|0|1|0|1
- ♦ Paarsusandmetega ketas: 1|1|0|1|1|0|1|1
- ♦ Teeme XOR operatsiooni vana ja uue baidi versiooni põhjal:
 - ♦ Vana: 0|1|0|0|1|1|0|0
 - ♦ Uus: 1|0|0|0|0|1|0|0
- ♦ XOR operatsiooni tulemus: 1|1|0|0|1|0|0|0
- ♦ Järeldus: muudatus on toimunud positsioonides 1,2 ja 5.

Lahendus 2 – näide (2)

- ♦ Paarsusandmetega kettal tuleb muuta baidi nr n vastavaid positsioone:
 - 11011011 => 00010011
- ♦ Ketas 1: 1|0|1|1|0|0|1|0
- ♦ Ketas 2: 1|0|0|0|0|1|0|0
- ♦ Ketas 3: 0|0|1|0|0|1|0|1
- ♦ Paarsusandmetega ketas: 0|0|0|1|0|0|1|1

RAID4 – puudused

- ♦ Andmeid saab taastada, kui rikneb vaid üks massiivi kuuluv ketas.
- ♦ Aeglustab andmemuudatusi. Eriti suur koormus paarsusandmetega kettal.

Andmebaasid II 2017 © Erki Eessaar

RAID5

RAID5 – vöötimine koos hajutatud paarsusandmetega

- ♦ RAID4 edasiarendus, kus paarsusandmed on hajutatud erinevate ketaste vahel.
- ♦ Enamasti riistvaraline toetus paarsusandmete arvutamisele.
- ♦ Üks populaarsemaid konfiguratsioone andmebaasiserverites.

23.11.2017 Teema 5 85

Andmebaasid II 2017 © Erki Eessaar

Tarkvaralised turvavahendid

- ♦ Tarkvaralisi turvavahendeid võib rakendada.
 - Operatsioonisüsteem.
 - Andmebaasisüsteem.
 - Rakendustarkvara.

23.11.2017 Teema 5 86

Andmebaasid II 2017 © Erki Eessaar

Operatsioonisüsteemi turvavahendid

- ♦ Kasutaja tuvastamine.
- ♦ Otsepöörduse vältimine failidele:
 - turvameetmed ei toimi, kui kasutajad neist mööda hiilivad.
- ♦ Failide perioodiline varundamine.

23.11.2017 Teema 5 87

Andmebaasid II 2017 © Erki Eessaar



Kasutaja tuvastamine

- ♦ **Autentimine** – olemi väidetava identiteedi tõesuse kontrollimine.
- ♦ **Identimine** – autentimise osa, mille käigus esitatakse süsteemile volitustõend.

23.11.2017 Teema 5 88

Andmebaasid II 2017 © Erki Eessaar

Volitustõendi tüübid






- ♦ *Teadmuslik*. Miski, mida subjekt teab (nt parool).
- ♦ *Esemeline*. Miski, mida subjekt valdab (nt luku võti).
- ♦ *Biomeetriline*. Miski, millest subjekt "koosneb" (nt sõrmejalg, silma võrkkesta muster, DNA)

23.11.2017 Teema 5 89

Andmebaasid II 2017 © Erki Eessaar

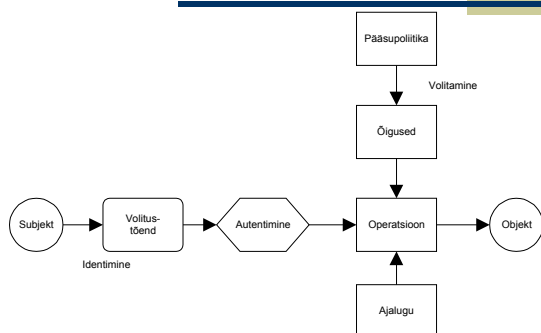
Volitustõendi tüübid (2)

- ♦ Usaldusväärse kolmanda osapoolle *kinnitus*.
- ♦ *Kontekstteave*, nt identifitseeritava asukoht (tohib süsteemi logida teatud kindlatest arvutitest).
- ♦ Volitustõend võib olla **tähtajaline** ja olla **kombinatsioon eelnevatest**.
 - Nt ID-kaarti puhul esemeline (kaart) ja teadmuslik (PIN koodid).

23.11.2017 Teema 5 90

Juurdepääsu reguleerumine andmebaasile ja selle objektidele



23.11.2017

Teema 5

91

Juurdepääsuõiguste jagamine

- ♦ **Minimaalse** pääsu poliitika (suletud süsteem).
 - Kõik, mis pole lubatud, on keelatud.
 - **Levinum!!**
 - Ärge unustage õiguseid suhte lõppedes ära võtta!
- ♦ **Maksimaalse** pääsu poliitika (avatud süsteem).
 - Kõik, mis pole keelatud, on lubatud.

23.11.2017

Teema 5

92

Tehingupõhised pääsupoliitikad

- ♦ Süsteem kontrollib, kas subjekt s tohib sooritada objektiga o tegevust (tööprotsessi sammu) t .
- ♦ Kirjeldab hulga *volitusi*, mis on kolmikud (s, o, t) .
- ♦ t sooritamiseks s poolt võib olla nõutav:
 - s kuulumine mingisse gruppi,
 - o andmed peavad vastama mingitele tingimustele,
 - eelnev tegevuste toimumine,
 - nt tellimust saab kustutada vaid siis, kui see on eelnevalt tühistatud.
 - järgnev tegevuste toimumine.

23.11.2017

Teema 5

93

Rollipõhised pääsupoliitikad

- ♦ Roll on õiguste kogum.
- ♦ Kasutajad kuuluvad ühte või mitmesse rolli.
- ♦ Ühel ja samal kasutajal võib olla erinevates olukordades eri rollid.
- ♦ Sama roll võib olla mitmel kasutajal.
- ♦ Rollid moodustavad rolli-hierarhiaid, kus kehtib õiguste päritavus.

23.11.2017

Teema 5

94

Rollipõhised pääsupoliitikad – näited

- ♦ Rollikandjate arvu piiramine.
- ♦ Staatiline kohustuste lahusus (välistavad rollid) – nt direktor ei saa olla audiitor.
- ♦ Dünaamiline kohustuste lahusus (välistavad rollid konkreetse juhtumi käsitlemisel).
- ♦ Protseduuriline kohustuste lahusus (keelatud juurdepääs järjekuliste protseduuri sammudele).
- ♦ Hiina müüri poliitika.

23.11.2017

Teema 5

95

Kuidas moodustuvad kasutaja õigused?

- ♦ Kasutaja õigused moodustuvad
 - kasutaja rollidele antud õigused +
 - kasutajatele antud õigused +
 - avalikkusele (PUBLIC) antud õigused.

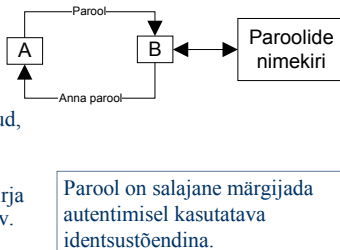
23.11.2017

Teema 5

96

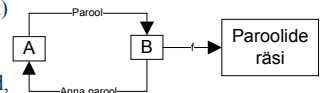
Lihtne paroolikaitse

- A identifitseerib ennast parooli kaudu.
- B kontrollib parooli olemasolu paroolide nimekirjas.
- A peab olema veendunud, et vastaspool on B.
- Probleem** – parool on *avatekst*, mis on nimekirja lekкимisel kõigile loetav.



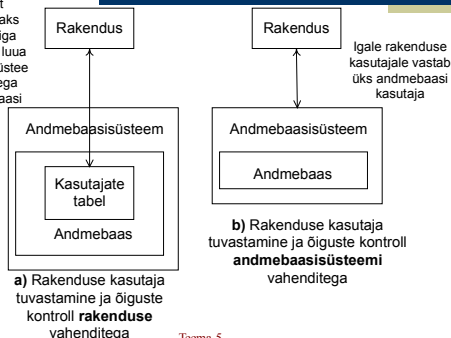
Parandatud paroolikaitse

- A identifitseerib ennast parooli kaudu.
- B kontrollib A parooli x õigsust arvutades räsi $f(x)$ ning võrreldes seda A tegeliku parooli räsiga.
- A peab olema veendunud, et vastaspool on B.
- Kui räsi lekitab, tuleb $f(x)$ järgi leida x' nii, et $f(x')=f(x)$.



Võimalusi paroolikaitse rakendamiseks

Selleks, et rakendus saaks andmebaasiga suhelda tuleb luua andmebaasisüsteemi vahenditega üks andmebaasi kasutaja



Variant a) vs. variant b)

- Eelistatud on kasutada varianti b)**, sest võimaldab maksimaalselt kasutada ära andmebaasisüsteemi sisseehitatud võimalused (kasutajad, rollid, auditeerimine).
- Variandi a) korral tuleb vastavad võimalused ehitada sisse andmebaasi kasutavas programmi.
 - Nendest võimalustest pole kasu, kui andmebaasi ei kasutata SELLE programmi abil.
 - Iga programmi jaoks tuleb need võimalused uuesti luua – väheneb arendajate produktiivsus.

Variandi a) puudused

- Rakendus suhtleb andmebaasiga kui kasutaja, millel on tööks ebavajalikult palju õigusi. See omakorda suurendab **SQL süstímise** rünnaku õnnestumise tõenäosust.
 - Ilmselt realiseerib üks ja sama rakendus mitu erinevat töökohta. Erinevad töökohad vajavad andmebaasis erinevaid õiguseid.

Variandi a) puudused (2)

- Rakendusele vastavale kasutajale tuleb anda töökohtadele vajalike õiguste ühend (*union*).
- Kui rakendusele vastav kasutaja on andmebaasiobjektide omanik, siis on tal nende üle täielik voli.

Kui siiski otsustada kasutada varianti a)

- ♦ Kasutajate tabelis parooli veergu => *räsifunktsioon(sool/parool)*
- ♦ Sool on räsiväärtuse leidmisel paroolile lisatav täidis, mis *raskendab sõnastikrünnet*.
 - Sõnastikrünn on jõurünne, mis parooli mõistatamiseks proovib suurest ammandavast loendist võetud sõnu või nende kombinatsioone.

Kui siiski otsustada kasutada varianti a) (2)

- ♦ Sool võiks olla vähemalt 32 märgi pikkune juhuslik väärtus, mis genereeritakse süsteemi poolt.
- ♦ Sool tuleb koos kasutajanime ja parooli räsiväärtusega samuti andmebaasis salvestada.
 - Kui räsifunktsioon integreerib soola räsiväärtusesse, siis pole eraldi soola veergu vaja.

Kui siiski otsustada kasutada varianti a) (3)

- ♦ Sool *ei pea olema salajane* – võib olla parooliga samas tabelis.
- ♦ Unikaalne soola väärtus tagab, et erinevate kasutajate korral, kellel on *sama parool*, genereeritakse *erinev räsiväärtus*.
 - Kui soola ei kasutata ja kasutajate A ning B paroolidel on sama räsiväärtus, siis tuvastades A parooli saab süsteemi sisse logida nii kasutajana A kui B.

Räsifunktsioonide näiteid

- ♦ Message-Digest algorithm 5 (**MD5**) (128 bitine räsiväärtus) – **ebaturvaline**
 - On väljatöötatud meetodid, kuidas leida MD5 abil leitud räsiväärtuse $f(x)$ korral x' nii, et $f(x')=f(x)$
- ♦ Secure Hash Algorithm (**SHA**)
 - SHA-1 (160 bitine räsiväärtus) – on leitud matemaatilisi nõrkusi
 - SHA-2 perekonna funktsioonid – parem!! Perekonda kuulub neli räsifunktsiooni, mis produtseerivad vastavalt 224, 256, 384 või 512 bitise räsiväärtuse.

Soovitus

- ♦ "Räsifunktsioonide osas tuleb üldise soovitusena loobuda funktsioonide **MD5** ja **SHA-1** kasutamisest. **5 aasta** perspektiivis sobivad kasutamiseks kõik **SHA-2** perekonna liikmed. **10 aasta** perspektiivis tuleks loobuda **SHA-224** kasutamisest; teised **SHA-2** perekonna liikmed on sellel ajahorisondil suure tõenäosusega jätkuvalt turvalised. **SHA3** standardi lõpliku kinnitamise järel on mõistlik kaaluda sellele üleminekut."

(AS Cybernetica, Krüptograafiliste algoritmide elutsükli uuring, ver 4.0, 3. juuni 2015)

Kui siiski otsustada kasutada varianti a) (4)

- ♦ Kasutaja identiteedi kontroll:
 - Leian kasutajanime alusel andmebaasist soola.
 - kas *räsifunktsioon (sool/sisestatud_parool)*= andmebaasis olev sool/parool põhjal leitud räsiväärtus?
 - Kontrollimine serveris (näiteks andmebaasiserveris talletatud rutiini abil).
- ♦ Rakendus suhtleb andmebaasisüsteemiga nagu üks kasutaja.
- ♦ **Rakendusele vastaval kasutajal ei tohi olla üleliigseid õiguseid!**



Miks on vaja soola?

- ♦ Ründaja sõnastikus võivad olla nii sõnad kui nende räsiväärtused. Neid räsiväärtuseid võrreldakse kasutajate tabelis olevate räsiväärtustega.
 - Räsiväärtuste salvestamine sõnastikus suurendab sõnastiku mahtu, kuid kiirendab võrdluste läbiviimist.
- ♦ Lootus, et vähemalt mõne kasutaja korral leidub vaste.

23.11.2017

Teema 5

109

Miks on vaja soola? (2)

- ♦ Kasutajate tabelis m isiku kasutajanimed, paroolid ja soola väärtused.
- ♦ Sõnastikus on n sõna ja nende alusel leitud räsiväärtused.
- ♦ Kui parooli veerus olev räsiväärtus on leitud ainult parooli põhjal.
 - Maksimaalne võrdluste arv $m \cdot n$.

23.11.2017

Teema 5

110

Miks on vaja soola? (3)

- ♦ Kui parooli veerus olev räsiväärtus on leitud soola ja parooli kombinatsiooni põhjal, siis tuleb iga "murtava" parooli jaoks luua eraldi sõnastik – töömaht kõigi kasutajate paroolide "murdmise" proovimiseks läheb suureks.
- ♦ Miks mitte kasutada *kasutajanime* soolana?
 - Saab luua sõnastiku, mis arvestab kasutajanimedid nagu *admin* ja *root*!

23.11.2017

Teema 5

111



Miks on vaja soola? – kokkuvõte

- ♦ Piisavalt pika juhusliku väärtusena leitud soola kasutamine ei lase rünnaku läbiviimiseks kasutada ette valmis tehtud sõnastikke, mis koosnevad sõnade ja nende räsiväärtuste paaridest (*pre-computed dictionary*).
- ♦ Rünnaku läbiviimine muutub *kulukamaks*.
- ♦ <http://www.martinstoeckli.ch/hash/en/>

23.11.2017

Teema 5

112



Mis juhtub kui ründaja saab teada kasutajanime/parooli?

- ♦ Muutub võimalikuks **identiteedivargus** – esinemine kellegi teisenä.
- ♦ Kuna sageli kasutavad inimesed sama parooli korduvalt, siis võib identiteedivargus toimuda paljudes süsteemides.

23.11.2017

Teema 5

113

Räsiväärtuse arvutamine (PostgreSQL) – näide

- ♦ Lisamoodul *pgcrypto*
- ♦ Installeerimiseks oma andmebaasis *apex.ttu.ee* käivitada SQL lause
- ♦ CREATE EXTENSION IF NOT EXISTS pgcrypto WITH SCHEMA public;
 - Loodavad funktsioonid pannakse skeemi *public*
 - <http://www.postgresqlonline.com/journal/archives/165-Encrypting-data-with-pgcrypto.html>

23.11.2017

Teema 5

114

Räsiväärtuse arvutamine (PostgreSQL) – näide (2)

- ♦ CREATE TABLE Users (username VARCHAR(30) NOT NULL, pswhash VARCHAR(60) NOT NULL);
- ♦ CREATE UNIQUE INDEX pk_username ON Users(**Upper(username)**);
 - /*Kasutajanimede tõstutundetu unikaalsuse tagamiseks. Kasutaja1 ja kasutaja1 loetakse samaks kasutajanimeks. */

23.11.2017

Teema 5

115

Räsiväärtuse arvutamine (PostgreSQL) – näide (3)

- ♦ INSERT INTO Users(username, pswhash) VALUES ('martin', public.crypt('gloobus07', public.gen_salt('bf', 11)));
- ♦ Parooli andmebaasis olev räsiväärtus: \$2a\$11\$80VGURm2SVKi9XLNH9AIOzrt9q2q5ufhTJQCp4Y.lh/zlaZp3x6u

Blowfish plokkšifril põhinev algoritm korduste arvuga 11.

23.11.2017

Teema 5

116

Kasutaja autentimise funktsioon (PostgreSQL)

- ♦ /*Esimese parameetri oodatav väärtus on tõstutundetu kasutajanimi ja teise parameetri oodatav väärtus on tõstutundlik parool.*/
- ```
CREATE OR REPLACE FUNCTION f_is_user(text, text)
RETURNS boolean AS $$
DECLARE
 rslt boolean;
BEGIN
 SELECT INTO rslt (pswhash = public.crypt($2, pswhash))
 FROM Users WHERE Upper(username)=Upper($1);
 RETURN coalesce(rslt, FALSE);
END;
$$ LANGUAGE plpgsql SECURITY DEFINER
SET search_path = public, pg_temp;
```

23.11.2017

Teema 5

117

## Kasutaja autentimise funktsioon (PostgreSQL)(2)

- ♦ **Võtan avalikkuselt** õiguse käivitada funktsiooni f\_is\_user(text, text) (see õigus antakse avalikkusele vaikimisi)
  - REVOKE EXECUTE ON FUNCTION f\_is\_user(text, text) FROM PUBLIC;
- ♦ Selleks, et rakendus saaks kasutada funktsiooni f\_is\_user(text, text), tuleb anda rakendusele vastavale kasutajale selle funktsiooni käivitamise õigus.
  - Näide: GRANT EXECUTE ON FUNCTION f\_is\_user(text, text) TO rakendus;
  - rakendus – andmebaasis loodud kasutaja.

23.11.2017

Teema 5

118

## Kasutaja autentimise funktsioon (PostgreSQL)(3)

- ♦ pgcrypto toetab räsiväärtuse leidmiseks mitmeid erinevaid algoritme.
- ♦ Kui loote kasutajate tabeli/kasutajate kontrolli funktsiooni mingis muus skeemis kui andmebaasisüsteemi poolt automaatselt loodud skeem public, siis tuleb rakendusele vastavale kasutajale anda lisaks ka selle skeemi kasutamise õigus!
  - GRANT USAGE ON SCHEMA skeemi\_nimi TO rakendus;

23.11.2017

Teema 5

119

## Kasutaja autentimise funktsioon (PostgreSQL)(4)

- ♦ SELECT f\_is\_user('martin','gloobus');
  - Tulemus: FALSE
- ♦ SELECT f\_is\_user('martin','gloobus07');
  - Tulemus: TRUE
- ♦ SELECT f\_is\_user('Martin','gloobus07');
  - Tulemus: TRUE /\*kasutajanimi pole tõstutundlik\*/
- ♦ SELECT f\_is\_user(NULL,'gloobus07');
  - Tulemus: FALSE

23.11.2017

Teema 5

120

## Näide (PostgreSQL) – kriitika

- ♦ Kasutaja registreerimisel ning sisselogimise käigus kasutaja tuvastamisel edastatakse rakendusest kasutajanimi ja parool andmebaasisüsteemile **avatekstina**. Seega on ründajal võimalus sideliinide pealtkuulamisel kasutajanimi ja parool teada saada.
  - Kui samas füüsilises serveriarvutis, siis pole nii tõsine probleem.
  - Ühenduse turvamiseks HTTPS, SSL.

23.11.2017

Teema 5

121



## Kapseldamine

- ♦ Kapseldamine tähendab objekti andmestruktuuride ja nendega manipuleerimise protseduuride varjamist teiste objektide eest.
- ♦ Andmebaasides saab rakendada seda põhimõtet kasutades **vaateid**, **hetktõmmiseid** ja **rutiine**.
- ♦ *Eeldus* – igale rakenduse kasutajale vastab andmebaasi kasutaja.

23.11.2017

Teema 5

122



## Vaadete kasutamine andmetele juurdepääsu piiramiseks

- ♦ CREATE VIEW IDU\_Aine AS SELECT aine\_kood, nimetus FROM Aine WHERE aine\_kood LIKE 'IDU%' WITH CHECK OPTION;
- ♦ GRANT SELECT, INSERT, UPDATE, DELETE on IDU\_Aine TO Kasutaja;
- ♦ Kasutaja ei saa pöörduda otse tabeli *Aine* poole – kapseldamine.
- ♦ *Kasutaja* saab lisada, kustutada ainult aineid, mille kood algab stringiga "IDU".
- ♦ *Kasutaja* ei saa muuta koodi nii, et see ei algaks enam stringiga "IDU".

23.11.2017

Teema 5

123

## Rutiinide kasutamine andmetele juurdepääsu piiramiseks

- ♦ Rutiin: *Kinnita\_tellimus*
  - Sisaldab lauset: UPDATE Tellimus SET ...
- ♦ Saab määrata, kelle õiguste alusel toimub täitmine
  - **Rutiini looja.** Loojal pidi olema UPDATE õigus tabeli *Tellimus* suhtes. Seega käivitaja vajab vaid EXECUTE õigust rutiini suhtes.
  - **Rutiini käivitaja.** Käivitaja vajab nii EXECUTE õigust rutiini suhtes kui UPDATE õigust *Tellimus* suhtes.

23.11.2017

Teema 5

124

## Rutiinide kasutamine andmetele juurdepääsu piiramiseks (2)

- ♦ Täitmine **rutiini looja** õigustes.
  - Laseb üles ehitada süsteemi, kus andmebaasi kasutajal pole õiguseid baastabelite suhtes.
  - Kui selline rutiin on rünnatav SQL süstimise meetodil, siis saab palju kurja teha, sest rutiini loojal oli ilmselt palju õiguseid.
  - Oracles vaikimisi määrang.
  - PostgreSQLis tuleb määrata kasutades SECURITY DEFINER fraasi (pole vaikimisi).

23.11.2017

Teema 5

125

## Vaated ja rutiinid – kokkuvõte

- ♦ Andmeid tuleb kaitsta *mitmekesiselt* ja *sügavuti*.
- ♦ Juurdepääs andmetele ainult rutiinide, hetktõmmiste ja vaadete kaudu võimaldab realiseerida mitmekihilises turvasüsteemis ühe kihi.

23.11.2017

Teema 5

126

## Trigerid

- ♦ Triger saab reageerida andmemuudatustele
- ♦ Sõltuvalt andmebaasisüsteemist võib ka panna reageerima sisselogimisele, skeemi muudatusele, õiguste muudatusele
- ♦ Triger võib selle peale:
  - kontrollida tegevuse lubatavust
  - keelata tegevuse läbiviimine
  - logida tegevus ja/või kedagi teavitada



## Andmebaasis andmete krüpteerimine

- ♦ Krüpteerimine on andmete kodeerimine spetsiifilise algoritmi abil, mis muudab andmed mittekasutatavaks kõigi selliste programmide jaoks, millel ei ole **dekrüpteerimise** võti.
- ♦ Krüptosüsteemi kuuluvad.
  - Krüpteerimise võti.
  - Krüpteerimise algoritm, mis krüpteerimise võtme abil kodeerib andmed.
  - Dekrüpteerimise võti.
  - Dekrüpteerimise algoritm, mis dekrüpteerimise võtme abil dekodeerib andmed.

## Andmebaasis andmete krüpteerimine (2)

- ♦ **Sümmeetriline** krüpteerimine – krüpteerimiseks ja dekrüpteerimiseks sama võti.
  - Näide: Data Encryption Standard (DES)
- ♦ **Asümmeetriline** krüpteerimine – krüpteerimiseks ja dekrüpteerimiseks erinev võti.
  - Näide: RSA algoritm
- ♦ Väärtuse  $v$  krüpteerimisel leidub dekrüpteerimise algoritm ja võti, mille alusel saab leida algse  $v$ .
- ♦ Väärtuse  $v$  asendamine **räsiväärtusega**  $v'$  ei saa  $v$ -d tagasi teisendada (mõnikord öeldakse, et tegemist on **ühesuunalise krüpteerimisega**).

## Meetmed kasutajaliideses

- ♦ Kasutajaliides peaks võimaldama ainult lubatud tegevusi.
  - Kasutajakeskkonna (menüüd, nupud) genereerimine/programmeerimine vastavalt kasutusõigustele/rollidele.
- ♦ Veateated ei tohiks reeta tundlikku infot süsteemi "hingeelu" kohta (nt andmebaasi struktuuri).

## Testandmebaasid

- ♦ Andmebaasi ja programmide testimiseks mõeldud testandmebaasides ei tohiks olla *konfidentsiaalseid andmeid*.
- ♦ Testandmete genereerimine programmi poolt.
  - Testandmed peavad olema piisavalt realistlikud, sest andmekäitluskeele lausete täitmisplaanid sõltuvad andmete jaotusest!
  - Programm võib luua testandmebaasi tööbaasis olevate andmete moonutamise abil.




## Andmete maskimine

- ♦ Algsed väärtused asendatakse uute väärtustega, mis on leitud olemasolevatest väärtustest mingi algoritmi alusel.
- ♦ Andmetest säilib **maskimata** versioon.
- ♦ Kuigi lihtsaimad meetodid kustutavad väärtuseid, asendavad neid XXXX vms, on eesmärgiks saada **tõepärasena** paistvad andmed.



Andmebaasid II 2017 © Erki Eessaar



## Andmete maskimine (2)

- ♦ Võimaldab luua konfidentsiaalsetest andmetest **mittekonfidentsiaalseid**.
- ♦ Uuest väärtustest on raske, kuid mõnikord mitte päris võimatu, algseid väärtuseid teada saada. Tuleb valida sellised maskimise meetodid, et see poleks võimalik.

23.11.2017 Teema 5 133

Andmebaasid II 2017 © Erki Eessaar




## Miks läheb andmete maskimist vaja?

- ♦ Süsteemi **testimiseks** vajalike andmete saamiseks.
- ♦ Kasutajate **koolitamiseks**.
- ♦ Andmetest sõltuvate **andmebaasirakenduste vigade** põhjuste leidmiseks.
- ♦ Sünteetilised (genereeritud) andmed ei ole piisavalt realistlikud ja head.

23.11.2017 Teema 5 134

Andmebaasid II 2017 © Erki Eessaar



## Andmete maskimise realiseerimise strateegiad

- ♦ **Staatiline** maskimine – andmebaasist luuakse maskitud andmeid sisaldav koopia. Uute andmete lisamiseks tuleb luua koopia uuesti.
- ♦ Staatiline maskimine koos võimalusega **uusi andmeid** juurde kanda.
- ♦ **Dünaamiline** maskimine – andmeid maskitakse käigult. Andmetest ei teki koopiat.

23.11.2017 Teema 5 135

Andmebaasid II 2017 © Erki Eessaar




## Asenduste (moonutuste) näiteid

| Meetod                                | Väärtus enne   | Väärtus pärast |
|---------------------------------------|----------------|----------------|
| Asendamine                            | Martin         | Madis          |
| Segamine veeru piires                 | Martin         | Tarmin         |
| Väikese juhuvea lisamine (kuupäevale) | 11.03.2017     | 17.03.2017     |
| Väikese juhuvea lisamine (arvule)     | 10             | 11             |
| Kustutamine                           | 10             | NULL           |
| Märkide asendamine                    | 30235087868705 | 3023508786XXXX |

23.11.2017 Teema 5 136

Andmebaasid II 2017 © Erki Eessaar



## Liiasus kaitsevahendina

- ♦ Väljade, ridade või kogu andmebaasi kontrollkoodid.
  - Sisestusvigade, muudatuste avastamine.
- ♦ Plokiaheldus.
  - Andmetest räsiväärtuste leidmine ja räsiväärtuste koopiade hajutatud paigutamine eesmärgiga valvata selle üle, et keegi andmeid ei muudaks.
  - Iga uus räsiväärtus sõltub eelnevast.

23.11.2017 Teema 5 137

Andmebaasid II 2017 © Erki Eessaar

## Liiasus kaitsevahendina (2)

- ♦ Andmete varundamine.
  - Varundamine on liiasusel põhinev käideldavuse ja tervikluse tugevdamise abinõu, mis tähendab infosüsteemi varuvarade loomist või soetamist varade osalise või täieliku hävimise või kasutamiskõlbmatuks muutumise puhuks.
- ♦ Andmete replikeerimine.
  - Andmebaasist on (enamasti erinevatel serveritel) koopiad.

[PostgreSQL](#)

23.11.2017 Teema 5 138

## Kontrollkoodid – ISBN

- ♦ ISBN kood on: 1 55860 432 4
- ♦ Kontroll:
- ♦  $1*1 + 2*5 + 3*5 + 4*8 + 5*6 + 6*0 + 7*4 + 8*3 + 9*2 = 158$
- ♦ Kontrollkood =  $158 \text{ MOD } 11 = 4$ , sest  $158 = 14*11 + 4$
- ♦ Kui 11-ga jagamise tulemusel tekib jääk on "10", siis asendatakse see ISBN numbris X-ga.

## Sisestusvigade avastamine kontrollkoodiga

- ♦ Üksik puuduv sümbol
  - 1 5860 432 4  $\Rightarrow$  1 5860 432 4
  - 1 5860 432 4  $\Rightarrow$  Kontrollkood =  $120 \text{ MOD } 11 = 10 = X$
- ♦ Üksik ülearune sümbol
  - 1 55860 432 4  $\Rightarrow$  1 55860 432 4
  - 1 55860 432 4  $\Rightarrow$  Kontrollkood =  $213 \text{ MOD } 11 = 4$

## Sisestusvigade avastamine kontrollkoodiga (2)

- ♦ Üksik vale sümbol
  - 1 55860 432 4  $\Rightarrow$  1 55861 432 4
  - 1 55861 432 4  $\Rightarrow$  Kontrollkood =  $164 \text{ MOD } 11 = 10 = X$
- ♦ Paariviisi ära vahetatud sümbol
  - 1 55860 432 4  $\Rightarrow$  1 55680 432 4
  - 1 55680 432 4  $\Rightarrow$  Kontrollkood =  $160 \text{ MOD } 11 = 6$

## Liiasus kaitsevahendina (2)

- ♦ Eesti Vabariigil plaan avada Euroopa Liidu riikides *andmesaatkondi*, kuhu paigutatakse koopiad tähtsamatest Eesti andmekogudest.
- ♦ Eesmärgiks tagada riikluse aluseks olevate infosüsteemide jätkuv toimimine (käideldavus).
- ♦ Andmetest räsiväärtuste (kontrollkoodid) leidmine ja plokiahelduse (*blockchain*) abil sidumine aitab leida loata muudatusi.

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks

- ♦ Saab määrata, millistest asukohtadest (IP aadressid) võib võtta milline kasutaja ühendust milliste andmebaasidega ning kuidas peab ühenduse soovija ennast tuvastama (*pg\_hba.conf* konfiguratsioonifail).
- ♦ Kasutajad, rollid.
- ♦ Õiguste andmine kasutajatele/rollidele ja nende äravõtmine.

## PostgreSQL SQL dialekti lauseid

- ♦ CREATE USER – kasutaja loomine
- ♦ CREATE ROLE – rolli loomine
- ♦ GRANT lause – õiguste või rollide andmine
- ♦ REVOKE lause – õiguste või rollide äravõtmine

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks (2)

- ♦ Vaated, funktsioonid – saab keelata otsese pöördumise tabelite poole.
- ♦ Lisamoodul *pgcrypto* – funktsioonid andmete krüpteerimiseks /deküpteerimiseks ja räsiväärtuste leidmiseks.
- ♦ *Hot standby* ja *Streaming replication* (voogreplikeerimine)
  - Saab luua varuandmebaasi, kuhu kantakse jooksvalt üle põhjandmebaasis tehtud muudatused.
  - Varuandmebaasis saab teha päringuid.
  - *Streaming replication* võimaldab andmemuudatusi üle kanda väga lühikese viiteajaga või täiesti sünkroonselt.

23.11.2017

Teema 5

145

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks (3)

- ♦ Alates PostgreSQL 9.5 võimalus defineerida baastabeliga seotud *turvapoliitika*d ja nende järgimist sisse/välja lülitada.
- ♦ Iga turvapoliitika võimaldab defineerida kaks *tõeväärtusavaldist*, mis määravad, milliseid tabeli **ridu**:
  - kasutaja/roll näeb ja õiguste olemasolul võib muuta ning kustutada,
  - millistele tingimustele peavad vastama selle poolt INSERT ja UPDATE lausetega tabelisse lisatavad read.

23.11.2017

Teema 5

146

## Turvapoliitika näide

- ♦ Poliitikate järgimise sisselülitamine
  - ALTER TABLE Kasutaja ENABLE ROW LEVEL SECURITY;
- ♦ Poliitika loomine
  - CREATE POLICY kasutaja\_poliitika ON Kasutaja TO kasutaja\_r USING (kasutajanimi = current\_user) WITH CHECK (kasutajanimi = current\_user AND on\_aktiivne=TRUE);
    - kasutaja\_r on kasutaja või rolli nimi

23.11.2017

Teema 5

147

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks (4)

- ♦ Poliitika saab siduda spetsiifilise lause tüübiga.
  - SELECT, INSERT, UPDATE, DELETE
- ♦ Vaikimisi rakenduvad kõigile kasutajatel/rollidele kõikide võimalike lause tüüpide korral.
- ♦ Kitsendavad GRANT lausetega antud õiguseid.

23.11.2017

Teema 5

148

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks (5)

- ♦ Turvapoliitikad sobivad kasutamiseks kui soovitakse realiseerida süsteem, kus rakendus pöördub **otse baastabelite** poole.
- ♦ Kui rakendus kasutab andmebaasi läbi **avaliku liidese** (funktsioonid, vaated, hetktõmmised), siis saab sama tulemuse saavutada andes kasutajale valikuliselt õiguseid kasutada avaliku liidese elemente.

23.11.2017

Teema 5

149

## Näiteid PostgreSQL võimalustest andmete turvalisuse tagamiseks (6)

- ♦ Toetus SELinux turvalahendusele. Ühtne juurdepääsu poliitika ja selle esitusviis nii andmebaasi kui operatsioonisüsteemi tasemel.
  - Andmebaasiobjektiga saab siduda turvalipiku (SECURITY LABEL lause)
  - Kui kasutaja *k* soovib kasutada andmebaasiobjekti *o*, siis otsustab süsteem *k*-ga ja *o*-ga seotud lipikute alusel, kas lubada juurdepääsu või mitte.
  - Lipikute abil määratud piirangud rakenduvad lisaks GRANT/REVOKE abil määratud piirangutele.

23.11.2017

Teema 5

150

## Näiteid Oracle võimalustest andmete turvalisuse tagamiseks

- ♦ Kasutajad, rollid
- ♦ Õiguste andmine kasutajatele/rollidele ja nende äravõtmine
- ♦ Vaated, rutiinid – saab keelata otsese pöördumise tabelite poole
- ♦ Kasutajate profiilid
  - Saab nt määrata parooli vahetuse sageduse

## Näiteid Oracle võimalustest andmete turvalisuse tagamiseks (2)

- ♦ Serveripoolne parooli sobivuse kontroll enne parooli jõustamist
- ♦ Rolli parool, turvaline rakenduse roll, mitte-vaikimisi roll
  - Kasutaja peab enda rolli aktiveerimiseks läbima turvakontrolli
- ♦ Andmete läbipaistev krüpteerimine
  - *Transparent Data Encryption*

## Näiteid Oracle võimalustest andmete turvalisuse tagamiseks (3)

- ♦ Andmete sihilik moonutamine
  - Andmete läbipaistev redigeerimine – päringutulemusi muudetakse käigupealt
    - *Transparent data redaction*
    - Realiseerib *dünaamilist* andmete maskimist
  - Enne andmete testandmebaasi kandmist tundlike andmete leidmine ja asendamine
    - *Data Masking*
    - Realiseerib *staatilist* andmete maskimist

## Mida tähendab läbipaistvus?

- ♦ **Läbipaistvus** tähendab, et süsteem teeb tegevusi andmete küsimise järel küsija jaoks nähtamatult – andmeid küsivaid rakendusi ei ole vaja muuta

## Näiteid Oracle võimalustest andmete turvalisuse tagamiseks (4)

- ♦ *Fine-grained access control* ja *label security*
  - Juurdepääsu piiramine tabeli valitud ridadele
- ♦ Auditeerimine
  - Andmebaasis tehtavate tegevuste jälgimine
- ♦ *Data-guard*
  - Jooksvalt täiendatav varuandmebaas

## Näiteid Oracle võimalustest andmete turvalisuse tagamiseks (5)

- ♦ Päringud mineviku ajahetke seisuga
  - Saab vaadata, milliseid andmeid võis ründaja näha
- ♦ *Oracle Access Management (single sign-on)*
  - Üks parool paljude andmebaaside jaoks
- ♦ Rollide ja õiguste analüüs
  - Võimalus tuvastada kasutajate üleliigseid õiguseid

## Oracle serveri arhitektuur alates Oracle 12c – kasutajad/rollid

- ♦ Üldised kasutajad/rollid
  - Defineeritakse konteinerandmebaasi tasemel.
  - Kasutaja-definieeritud üldiste kasutajate/rollide nimed peavad algama märkidega `c##` või `C##`.
  - Võib võtta ühendust nii juurkonteineriga kui ka kõigi alamandmebaasidega, mille suhtes on sellele õigused antud.
- ♦ Lokaalsed kasutajad/rollid
  - Defineeritakse alamandmebaasi tasemel.
  - Erinevates alamandmebaasides võivad olla sama nimega lokaalsed kasutajad.

23.11.2017

Teema 5

157

## Kasutaja loomine (Oracle)

- ♦ `CREATE USER c##ashwini IDENTIFIED BY out_standing1 DEFAULT TABLESPACE users QUOTA 10M ON users TEMPORARY TABLESPACE temp QUOTA 5M ON system;`

23.11.2017

Teema 5

158

## Kasutaja ja rollide haldamine

- ♦ `CREATE ROLE c##tootaja;`
- ♦ `CREATE ROLE c##tootaja IDENTIFIED BY M2AE199K;`
- ♦ `GRANT CONNECT TO c##tootaja;`
- ♦ `GRANT RESOURCE TO c##tootaja;`
  - `CONNECT` ja `RESOURCE` on süsteemi-definieeritud rollid
- ♦ `GRANT c##tootaja TO c##ashwini;`

23.11.2017

Teema 5

159

## Kasutajate ja rollide haldamine (2)

- ♦ `ALTER USER c##ashwini IDENTIFIED BY uusparool;`
- ♦ `ALTER USER c##ashwini ACCOUNT LOCK;`
- ♦ `ALTER USER c##ashwini ACCOUNT UNLOCK;`
- ♦ `ALTER USER c##ashwini DEFAULT ROLE ALL EXCEPT c##tootaja;`

23.11.2017

Teema 5

160

## Rolli parool, mitte-vaikimisi roll

- ♦ Otse andmebaasi poole pöörduv kasutaja peab endale antud mitte-vaikimisi rolli aktiveerimiseks teadma parooli.
- ♦ Rakendus aktiveerib käivitumisel rolli.
  - `SET ROLE c##tootaja IDENTIFIED BY M2AE199K;`
- ♦ Kasutaja saab endale rolliga antud õiguseid kasutada ainult rakenduse kaudu.
- ♦ Enne rakenduse sulgemist peab rollid deaktiveerima:
  - `SET ROLE NONE;`

23.11.2017

Teema 5

161

## Turvaline rakenduse roll, mitte-vaikimisi roll

- ♦ Sellise rolliga on seotud pakett (antud näite korral `c##hr.approles_package`), mis teostab erinevaid turvakontrole.
  - `CREATE ROLE c##acme_hr_role IDENTIFIED USING c##hr.approles_package;`
- ♦ Kui rolliga on seotud turvakontrole teostav pakett, siis peab kasutaja mitte-vaikimisi rolli aktiveerimiseks need turvakontrollid läbima.

23.11.2017

Teema 5

162

## Kasutaja profiili loomine (Oracle)

- ♦ CREATE PROFILE c##prof LIMIT  
FAILED\_LOGIN\_ATTEMPTS 4  
PASSWORD\_LOCK\_TIME 30 --päevade arv  
PASSWORD\_LIFE\_TIME 90 --päevade arv  
PASSWORD\_GRACE\_TIME 3 --päevade arv  
PASSWORD\_REUSE\_TIME 60 --päevade arv  
PASSWORD\_REUSE\_MAX 1  
/\*60 päeva pärast saan kasutada sama parooli, kui selle aja jooksul olen vähemalt ühe korra parooli muutnud\*/  
PASSWORD\_VERIFY\_FUNCTION verify\_function;

23.11.2017

Teema 5

163

## Kasutaja profiili loomine (Oracle) (2)

- ♦ Kasutajale määratakse profiil:
  - ALTER USER c##ashwini PROFILE c##prof;
- ♦ Kasutaja profiil asendatakse:
  - ALTER USER c##ashwini PROFILE c##prof\_piirangutega;
- ♦ Profiil kustutatakse ja selle profiiliga kasutajad saavad DEFAULT profiili (vaikimisi ei sea see profiil piiranguid):
  - DROP PROFILE c##prof CASCADE;

23.11.2017

Teema 5

164

## Kasutaja tegevusvabaduse kitsendamine SQL\*Plusis

- ♦ Kasutaja c##tudi1 ei tohi SQL\*Plusi kaudu andmebaasis käivitada DELETE lauseid:
  - INSERT INTO PRODUCT\_USER\_PROFILE (product, userid, attribute, char\_value) VALUES ('SQL\*Plus', 'C##TUDI1', 'DELETE', 'DISABLED');
  - COMMIT;
  - C##TUDI1@ORCL > DELETE from t990999\_koristamine;
  - SP2-0544: Command "delete" disabled in Product User Profile

23.11.2017

Teema 5

165

## Õigused

- ♦ Süsteemiõigused – jagab administraator
  - CREATE SESSION
  - CREATE TABLE
  - CREATE ANY TABLE
  - ALTER ANY TABLE
  - SELECT ANY TABLE
  - DELETE ANY TABLE
  - ...
- ♦ Objektiõigused – jagab skeemiobjekti omanik või administraator
  - Nt SELECT, INSERT, UPDATE, DELETE õigus konkreetse tabeli suhtes
  - Alates Oracle 12.1 eristatakse andmete lugemisel SELECT õigust (õigus lugeda ja ridu või tabelit **lukustada**) ja READ õigust (õigus ainult lugeda)

23.11.2017

Teema 5

166

## Vähim võimalik hulk õiguseid (Oracle)

- ♦ Piira ALTER SESSION, ALTER SYSTEM õiguste andmist
- ♦ Piira ANY tüüpi õiguste andmist (nt CREATE ANY TABLE)
- ♦ Piira "WITH ADMIN" ja "WITH GRANT" määranguid õiguste jagamisel. Võimaldavad õiguseid edasi jagada.
- ♦ Piira juurdepääsu süsteemikataloogile

23.11.2017

Teema 5

167

## Vaadete kasutamine (Oracle)

```
CREATE TABLE Message (
 message_id NUMBER(10) GENERATED AS IDENTITY
 CONSTRAINT pk_messages PRIMARY KEY,
 sender VARCHAR2(20) NOT NULL,
 receiver VARCHAR2(20) NOT NULL,
 message_body VARCHAR2(2000) NOT NULL,
 time DATE DEFAULT LOCALTIMESTAMP(0) NOT NULL);
```

Loon kõik skeemiobjektid  
skeemis nimega c##naited.

23.11.2017

Teema 5

168

Andmebaasid II 2017 © Erki Eessaar

## Vaade, mis leiab igale kasutajale tema poolt saadetud teated (Oracle)

```
CREATE OR REPLACE VIEW
 my_sent_message AS
SELECT sender, reciever, message_body, time
FROM Message
WHERE sender = user
WITH CHECK OPTION;
```

**WITH CHECK OPTION** – andmemuudatus läbi vaate peab rahuldama vaate alampäringu tingimusi.

23.11.2017 Teema 5 169

Andmebaasid II 2017 © Erki Eessaar

## Vaade, mis leiab igale kasutajale just talle saadetud teated (Oracle)

```
CREATE OR REPLACE VIEW
 message_to_me AS
SELECT sender, reciever, message_body, time
FROM Message
WHERE reciever = user
WITH READ ONLY;
```

**WITH READ ONLY** – vaate kaudu ei saa andmeid muuta.

23.11.2017 Teema 5 170

Andmebaasid II 2017 © Erki Eessaar

## Õiguste jagamine

- ♦ GRANT SELECT, INSERT ON my\_sent\_message TO c##ylioplane;
- ♦ GRANT SELECT ON message\_to\_me TO c##ylioplane;
- ♦ Kasutaja c##tud1 kuulub rolli c##ylioplane.

23.11.2017 Teema 5 171

Andmebaasid II 2017 © Erki Eessaar

## Teadete süsteemi kasutamine kasutaja C##TUD1 poolt (Oracle)

- ♦ INSERT INTO c##naited.my\_sent\_message (sender, reciever, message\_body) VALUES ('C##TUD1', 'C##TUD11', 'Teade');
- Lisamine õnnestub
- ♦ INSERT INTO c##naited.my\_sent\_message (sender, reciever, message\_body) VALUES ('C##TUD2', 'C##TUD11', 'Teade');
- ERROR at line 1: ORA-01402: view WITH CHECK OPTION where-clause violation

23.11.2017 Teema 5 172

Andmebaasid II 2017 © Erki Eessaar

## AUTHID CURRENT\_USER (Oracle)

- ♦ CREATE TABLE Toode(toode\_id NUMBER PRIMARY KEY, nimi VARCHAR2(100) UNIQUE);
- ♦ INSERT INTO Toode(toode\_id, nimi) VALUES (1, 'Tallinna kilud');
- ♦ INSERT INTO Toode(toode\_id, nimi) VALUES (2, 'Kirde sai');
- ♦ COMMIT;

```
CREATE OR REPLACE PACKAGE Types AS
TYPE cursorType IS REF CURSOR;
END Types;
/
```

Loon kõik skeemiobjektid skeemis nimega c##naited.

23.11.2017 Teema 5 173

Andmebaasid II 2017 © Erki Eessaar

## AUTHID CURRENT\_USER (Oracle) (2)

| SECURITY DEFINER                                                                                                                                                                                         | SECURITY INVOKER                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>create or replace function f_kaupade_nimekiri_definer return types.cursorType as l_cursor types.cursorType; begin open l_cursor for select nimi from c##naited.toode; return l_cursor; end; /</pre> | <pre>create or replace function f_kaupade_nimekiri_invoker return types.cursorType AUTHID CURRENT_USER as l_cursor types.cursorType; begin open l_cursor for select nimi from c##naited.toode; return l_cursor; end; /</pre> |

23.11.2017 Teema 5 174



## AUTHID CURRENT\_USER (Oracle) (3)

- ♦ GRANT EXECUTE ON f\_kaupade\_nimekiri\_definer TO C##ylioplane;
- ♦ GRANT EXECUTE ON f\_kaupade\_nimekiri\_invoker TO C##ylioplane;
  - /\*Login sisse kasutajana, kellel on roll C##Ylioplane.\*/
- ♦ SELECT \* FROM c##naited.toode;
  - /\*Ei saa täita – pole päringu tegemise õigust selle tabeli suhtes\*/
- ♦ SELECT c##naited.f\_kaupade\_nimekiri\_definer FROM dual;
  - /\*Täidetakse\*/
- ♦ SELECT c##naited.f\_kaupade\_nimekiri\_invoker FROM dual; /\*Ei saa täita – käivitajal ei ole tabeli c##naited.Kaup suhtes päringu tegemise õigust\*/

## Fine grained access control (FGAC)

- ♦ Aitab tagada andmete *konfidentsiaalsust*.
- ♦ GRANT lausega saab anda õiguse terve tabeli suhtes.
- ♦ FGAC abil saab anda õiguseid üksikute tabeli ridade suhtes.
- ♦ Alternatiiv vaadete ja andmebaasis talletatud rutiinide kasutamisele.

## Kuidas FGAC töötab?

- ♦ Kasutaja andmekäitluskeele (SELECT, INSERT, UPDATE, DELETE) lausesse võidakse süsteemi poolt lisada enne täitmist WHERE klausel.
- ♦ Tuleb luua poliitika funktsioon (mis lisab kitsenduse mingi tabeli põhjal tehtud andmekäitluskeele lausesse) ja see registreerida.
- ♦ Kasutatakse *virtuaalsete privaatsete andmebaaside* loomiseks.

## FGAC – tabeli näide (autor T. Kyte)

- ♦ CREATE TABLE data\_table (some\_data VARCHAR2(30), owner VARCHAR2(30) DEFAULT USER); -- omanik c##naited
- ♦ GRANT ALL ON data\_table TO PUBLIC;
- ♦ CREATE PUBLIC SYNONYM data\_table FOR c##naited.data\_table;
- ♦ INSERT INTO data\_table ( some\_data, owner ) VALUES ( 'Some Data Owned by C##TUD1', 'C##TUD1' );
- ♦ COMMIT;

## FGAC – poliitika funktsiooni näide

```
create or replace function security_policy_function(p_schema
in varchar2, p_object in varchar2)return varchar2
as
begin
if (user = p_schema) then return '';
else return 'owner = USER';
end if;
end;
```

Tabeli omanik näeb  
kõiki andmeid.  
Ülejäänud kasutajad  
näevad ridu, mille  
omanikuks nad on

## FGAC – poliitika registreerimine

```
begin
dbms_rls.add_policy
(object_schema => 'c##naited',
object_name => 'data_table',
policy_name => 'MY_POLICY',
function_schema => 'c##naited',
policy_function => 'security_policy_function',
statement_types => 'select, insert, update, delete',
update_check => TRUE,
enable => TRUE);
end;
```

## FGAC – kasutamine

- ♦ Päringu teeb kasutaja C##TUD2:
  - Käivitav: SELECT \* FROM data\_table;
  - Täidetav: SELECT \* FROM data\_table WHERE owner='C##TUD2';
  - Ei näe ühtegi rida.
- ♦ Päringu teeb kasutaja C##TUD1:
  - Käivitav: SELECT \* FROM data\_table;
  - Täidetav: SELECT \* FROM data\_table WHERE owner='C##TUD1';
  - Näeb rida, kus omanikuks on C##TUD1.

## FGAC – kasutamine

- ♦ Päringu teeb kasutaja C##NAITED (tabeli omanik):
  - Käivitav: SELECT \* FROM data\_table;
  - Täidetav: SELECT \* FROM data\_table;
  - Näeb kõiki tabeli ridu.

## Label security

- ♦ Aitab tagada andmete *konfidentsiaalsust*.
- ♦ FGAC edasiarendus.
- ♦ Iga tabeli reaga on seotud *lipik*.
- ♦ Rea lipik määrab:
  - turvalisuse aste,
  - kategooria, kuhu rida kuulub,
  - rea seotus gruppidega.

## Label security (2)

- ♦ Kasutaja või rolliga on seotud *lipik*.
- ♦ Lipik määrab:
  - turvalisuse astme vahemik,
  - kategooriad,
  - grupid.

## Label security (3)

- ♦ Kasutaja saab kasutada rida, kui:
  - see kuulub tema turvalisuse astme vahemikku,
  - kõik rea kategooriad on ka kasutaja kategooriad,
  - vähemalt üks reaga seotud grupp ja kasutajaga seotud grupp kattuvad.
- ♦ Saab luua *turvapoliitikaid* ja neid jooksvalt muuta.

## Label security (4)



User session label is UNCLASSIFIED

| GRADE     | RATE | ROW LABEL        |
|-----------|------|------------------|
| Manager   | 600  | UNCLASSIFIED     |
| Senior    | 400  | UNCLASSIFIED     |
| Director  | 750  | HIGHLY_SENSITIVE |
| Principal | 600  | SENSITIVE        |
| Senior    | 450  | SENSITIVE        |

## Auditeerimine

- ♦ Aitab tagada andmete *revideeritavust*.
- ♦ Võimaldab anda korralduse jälgida andmebaasis tehtavaid tegevusi (andmed nende kohta registreeritakse süsteemikataloogis) ja vaadata hiljem andmeid selle kohta.
  - `AUDIT SELECT, INSERT, UPDATE, DELETE ON c##naited.dept BY ACCESS WHENEVER SUCCESSFUL;`
  - `AUDIT DELETE ANY TABLE;`
  - `AUDIT CREATE SESSION;`

23.11.2017

Teema 5

187

## Auditeerimine (2)

- ♦ `SELECT username, Count(*) AS arv  
FROM sys.dba_audit_session  
WHERE CURRENT_DATE<timestamp+  
interval '7' day  
GROUP BY username  
ORDER BY Count(*) DESC;`

Andmebaasi sisselogimiste arv kasutajate kaupa viimase 7 päeva jooksul.

23.11.2017

Teema 5

188

## Fine grained auditing (1)

- ♦ Saab jälgida spetsiifilisi päringuid/andmemuudatusi.
- ♦ Aitab tagada andmete *revideeritavust*.
- ♦ `EXECUTE dbms_fga.add_policy('C##NAITED', 'EMP', 'policy1', 'deptno = 20');`
  - Saab ka määrata, et lisaks auditeerimisele käivitub mingi funktsioon (nt saadab e-maali)

23.11.2017

Teema 5

189

## Fine grained auditing (2)

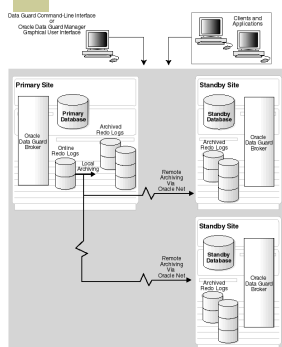
- ♦ `SELECT * FROM Emp WHERE deptno = 20;`
  - Käivitas auditeerimise
- ♦ `SELECT * FROM Emp WHERE deptno = 10;`
  - Ei käivita auditeerimist
- ♦ `SELECT timestamp, db_user, os_user, userhost, object_schema, sql_text FROM dba_fga_audit_trail WHERE policy_name='POLICY1';`
  - Auditeerimise logi

23.11.2017

Teema 5

190

## Oracle data guard



- ♦ Aitab tagada andmete *käideldavust*.
- ♦ Pakub võimalust kanda andmeid põhiandmebaasist varuandmebaasi *sünkroonselt* (tagab maksimaalse andmete turvalisuse) või *asünkroonselt* (tagab maksimaalse andmemuudatuste kiiruse).

23.11.2017

Teema 5

191

## Oracle data guard (2)

- ♦ Kui kanda muudatusi varuandmebaasi *plokkide kaupa*, siis ploki riknemine põhiandmebaasis kandub üle varuandmebaasidesse.
  - 2010 oli suurpanga JPMorgan Chase infosüsteemis sellel põhjusel katkestus.
  - Blokeeriti 132 miljoni USD väärtuses ülekandeid ning läks kaotsi umbes 1000 õppelaenu taotlust ning 1000 autoliisingu taotlust.
  - <http://www.dbms2.com/2010/09/17/jp-morgan-chase-oracle-database-outage/>

23.11.2017

Teema 5

192

## Andmete krüpteerimine

- ♦ Aitab tagada andmete *konfidentsiaalsust*.
- ♦ Alates Oracle10g pakett *dbms\_crypto*
- ♦ Sisaldab muuhulgas funktsioone.
  - Sümmeetrilist võtit kasutavate krüpteerimisalgoritmide *DES* ja *TripleDES* kasutamiseks.
    - Võtme genereerimine
    - Krüpteerimine
    - Dekrüpteerimine
  - Räsiväärtuse arvutamiseks *MD5*, *MD4*, *SHA-1*, *SHA-2* algoritmide abil

23.11.2017

Teema 5

193

## Transparent Data Encrypton

- ♦ Aitab tagada andmete *konfidentsiaalsust*
- ♦ Andmefailide tasemel andmed krüpteeritud
  - Saab krüpteerida kogu andmebaasi, üksikud tabeliruumid, või üksikud tabelite veerud
- ♦ Andmebaasi kasutaja näeb krüpteerimata andmeid – krüpteerimine/dekrüpteerimine tema jaoks nähtamatu (*transparent*)
  - Pole vaja rakendusi ümber kirjutada

23.11.2017

Teema 5

194

Dünaamiline  
andmete  
maskimine!

## Transparent Data Redaction

- ♦ Aitab tagada andmete *konfidentsiaalsust*
- ♦ Enne andmeväärtuste kasutajatele väljastamist andmebaasisüsteem muudab neid jooksu pealt vastavalt andmebaasis defineeritud/salvestatud poliitikale.
- ♦ Andmebaasis salvestatud andmed ei muutu.
- ♦ Andmeid kasutavaid rakendusi ei ole vaja ümber kirjutada.

23.11.2017

Teema 5

195

## Transparent Data Redaction (2)

- ♦ Eesmärgiks on kasutajate (eeskätt andmeid rakenduste kaudu kasutavate lõppkasutajate) eest andmeväärtuseid peita või isegi juhuslikult genereeritud väärtustega eksitada.

|          | Salvestatud andmed     | Väljastatud andmed   |
|----------|------------------------|----------------------|
| Täielik  | 10/09/1992             | 01/01/2001           |
| Osaline  | 987-65-4328            | XXX-XX-4328          |
| RegExp   | first.last@example.com | [hidden]@example.com |
| Juhuslik | 5105105105105100       | 5500000000000004     |

23.11.2017

Teema 5

196

## Rollide ja õiguste analüüs

- ♦ Aitab tagada andmete konfidentsiaalsust
- ♦ Sageli kasutajatel (sh rakendustel) üleliigseid õiguseid, mida nad ei vaja – turvarisk
- ♦ Igal kasutajal peaks olema *minimaalne hulk õiguseid*, et saaks oma töö tehtud
- ♦ Paketi *dbms\_privilege\_capture* abil saab jälgida andmebaasi kasutamise käigus kasutatud õiguseid ning tuvastada kasutamata õiguseid.

23.11.2017

Teema 5

197

## Eesti andmekogude turvalisuse tagamisest

- ♦ ISKE on infosüsteemide kolmeastmeline *etalonturbe* süsteem.
  - Saab kasutada nii riigi-, kohaliku omavalitsuse kui ka äriettevõtete infosüsteemides.
  - Etalonturbe süsteem kirjeldab tüüpilised riskid ja kuidas neid maandada – üks suurus kõigile.
- ♦ ISKE jagab infovarad kolmele turbeastmele: madal (L), keskmine (M) ja kõrge (H).

23.11.2017

Teema 5

198

## Eesti andmekogude näiteid

- ♦ Kõrge (H) turbeaste.
  - Eesti rahvastikuregister
  - Tervise infosüsteem
- ♦ Keskmine (M) turbeaste.
  - Eesti väärtapaberite keskreister.
  - Kohustusliku kogumispensioni register.
- ♦ Madal (L) turbeaste.
  - Töötuskindlustuse andmekogu.

## Eesti andmekogude turvalisuse tagamisest (2)

- ♦ ISKE turbeaste määratakse kasutades hindamisskaalat, kus andmete turvaeasmärke – *käideldavus, terviklikkus, konfidentsiaalsus* – hinnatakse neljapallisel skaalal.
  - Tulemuseks turvaklass. Näide: **K2T3S1**
- ♦ Turvaklassi järgi määratakse nõutav turbeaste.
- ♦ ISKE määrab ära ka turvameetmed igale turbeastele.

## Eesti andmekogude turvalisuse tagamisest (3)

- ♦ Valitsuse määrusega on ISKE kehtestatud riigi ja kohaliku omavalitsuse andmekogude pidajatele.
- ♦ SKE mõte on, et turbeaste tagamine oleks selgelt demonstreeritud.
- ♦ ISKE rakendusjuhend ja IT-turbejuhend.
  - <https://www.ria.ee/ee/iske.html>

## Eesti andmekogude turvalisuse tagamisest (4)

- ♦ Kõik riigi ja kohaliku omavalitsuse andmekogud peavad *regulaarselt* läbima ISKE auditi, mille eesmärk on anda sõltumatu hinnang, kas andmekogu pidamisel on ISKE turvameetmed rakendatud korrektselt, standardi nõudeid jälgides.
  - 2014 – ISKE rakendatud 59% nõutud süsteemides ja rakendamine auditeeritud umbes 20%-s.
  - Rakendamist pärsvad raha ja spetsialistide puudus.
- ♦ Hinnanguliselt kulub ühe keskmise andmekogu ISKE juurutamiseks kulub **400** inimtundi.

## Lõppsõna

- ♦ Pole olemas absoluutset turvalisust.
- ♦ Turvalisust mõõdetakse rünnakute keerukuse järgi.
- ♦ Turvalisust on raske/võimatu tõestada ja seda pole ka ühegi tänapäeva süsteemi korral tehtud.
- ♦ Tuleb arvestada teadmatute riskidega.

## Mõningad teema põhimõisted

