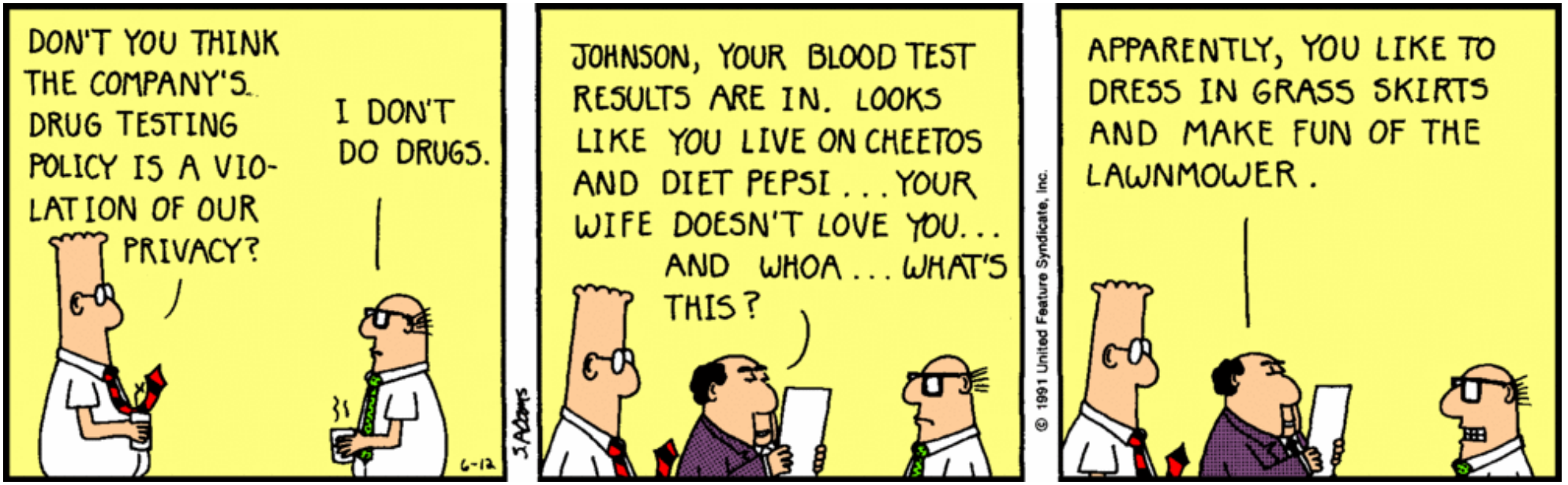

DON'T YOU THINK THE COMPANY'S DRUG TESTING POLICY IS A VIOLATION OF OUR PRIVACY?
I DON'T DO DRUGS.
6-12
JOHNSON, YOUR BLOOD TEST RESULTS ARE IN. LOOKS LIKE YOU LIVE ON CHEETOS AND DIET PEPSI . . . YOUR WIFE DOESN'T LOVE YOU. . . AND WHOA . . . WHAT'S THIS?
APPARENTLY, YOU LIKE TO DRESS IN GRASS SKIRTS AND MAKE FUN OF THE LAWNMOWER.
© 1991 United Feature Syndicate, Inc.

ZVOP-1, 3. odstavek, 22. člen:

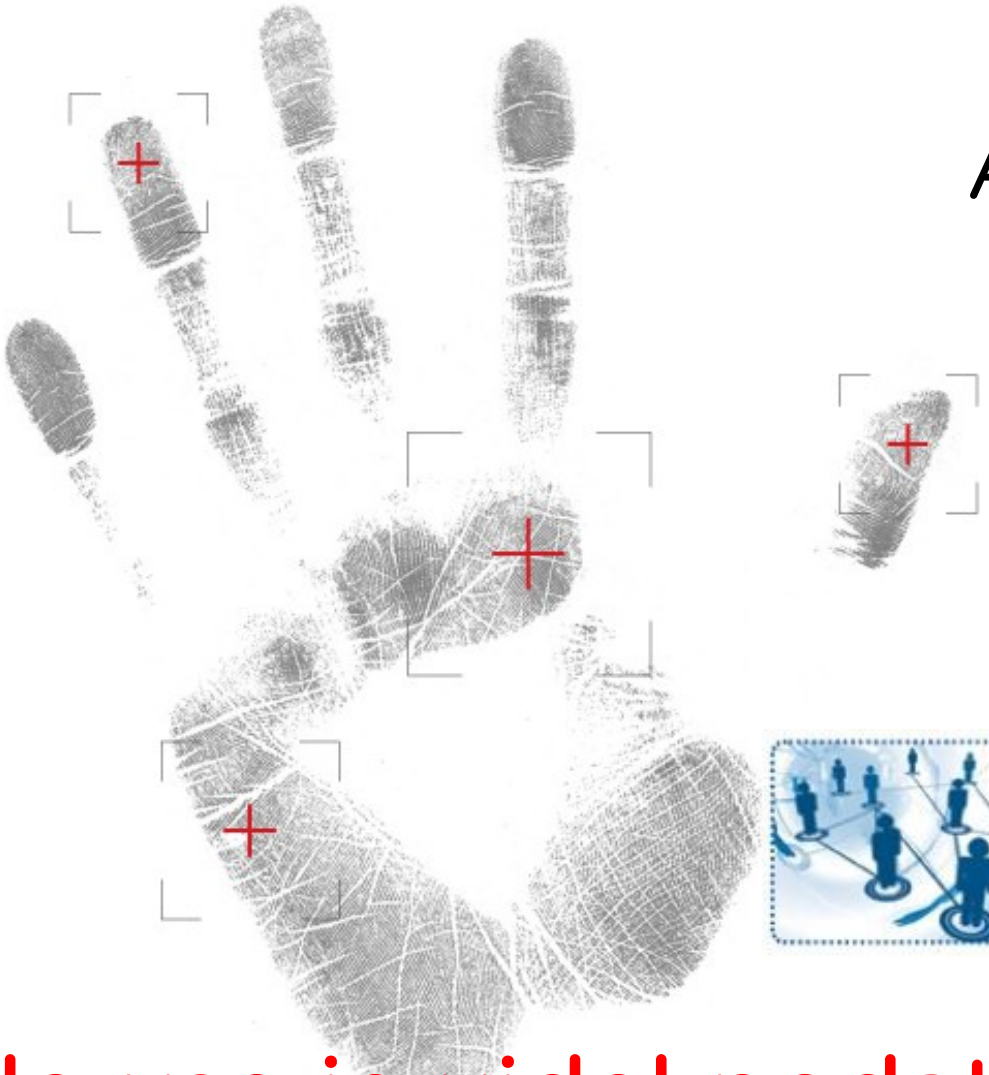
Upravljavec osebnih podatkov mora za vsako posredovanje osebnih podatkov zagotoviti, da je mogoče pozneje ugotoviti, kateri osebni podatki so bili posredovani, komu, kdaj in na kakšni podlagi, in sicer za obdobje, ko je mogoče zakonsko varstvo pravice posameznika zaradi nedopustnega posredovanja osebnih podatkov.

Kdo vse je videl podatke?

select * from delavci;

Abakus ARBITER

ORACLE Gold Partner



Boris Oblak

Abakus plus d.o.o.

ORACLE CERTIFIED PROFESSIONAL

17. Strokovno srečanje

SIOUG 2012

Slovensko društvo Oracle uporabnikov

Kongresni center Hotel Mons Ljubljana, 15. - 17. oktober

Kdo vse je videl podatke Marije Novak?

O podjetju

ORACLE Gold Partner

Zgodovina:

- od 1992, 20 zaposlenih
- Oracle zbirka podatkov, GNU/linux (1995)
- Dobitniki srebrnega priznanja za inovacije – Aerodrom Ljubljana: Flight Information System
- Dobitniki srebrnega priznanja za inovacije – Arbiter

Razvoj in vzdrževanje:

- Razvoj visoko razpoložljivih sistemov z OS GNU/linux
- Sistemska podpora in uglaševanje sistemov z OS GNU/linux
- Uglaševanje in administracija zbirk podatkov Oracle

Agenda

- poljudni del: predstavitev problema
- tehnični del: prikaz možne rešitve

Beleženje Marije Novak

- predavanje Stanke Šalamun (povod: evropski dan varstva osebnih podatkov)
- kdo ve več o nas samih kot mi sami?
- nakup zdravil?
- gibanje mobilnega telefona?
- stanje na računu tako, da bi se nas splačalo oropati?
- podatki, ki jih moramo posredovati zaradi zakonodaje oz. zato, da si po nepotrebnem ne zakompliciramo življenja

Masovni zbiranje OP

- informacijska tehnologija omogoča poceni masovno zbiranje podatkov kot tudi napredno iskanje
- kopičenje OP v vladnih in komercialnih bazah podatkov
- samo zakonodaja nas ščiti pred popolnim popisom

Mačehovski odnos do podatkov

- Kako skrbite za naše OP?
  »Hvala, dobro, kar brez skrbi!«
- preveriti trditve
  - kot navaden državljan
  - brez pravnikov
  - banke, finančne dejavnosti, zavarovalnice, zdravstvene in izobraževalne dejavnosti, sodstvo, javno upravo, komercialne družbe (trgovine), organizacija nagradnih iger

Abakus
As na disku.

ZVOP-1

- 6. člen:
  Upravljavec osebnih podatkov - je fizična ali pravna oseba, ki sama ali skupaj z drugimi določa namene in sredstva obdelave osebnih podatkov oziroma oseba, določena z zakonom, ki določa tudi namene in sredstva obdelave

Pravica posameznikov do seznanitve

- 30. člen: Upravljavec OP mora posamezniku na njegovo zahtevo:
  - 3. posredovati izpis OP, ki so vsebovani v zbirki OP in se nanašajo nanj
  - 4. posredovati seznam uporabnikov, katerim so bili posredovani OP, kdaj, na kakšni podlagi in za kakšen namen

AS Abakus As na disku.

Postopek seznanitve

- 31. člen
  Upravljavec OP mora posamezniku omogočiti vpogled, prepis, kopiranje in potrdilo po 1. in 2. točki prvega odstavka 30. člena tega zakona praviloma istega dne, ko je prejel zahtevo, najpozneje pa v 15 dneh, ali pa ga v 15 dneh pisno obvestiti o razlogih, zaradi katerih izdaje potrdila ne bo omogočil.

Generiranje dnevniških zapisov

- aktivnost, da je upravljavec dodal OP v svojo bazo
- občasni klici k upravljavcu so (naj bi) povzročili zapis revizijske sledi
  - DURS – stanje plačanih davkov
  - banka: stanje kredita, prošnja za kredit
  - dvig zdravila na recept
  - ...

Rezultati

- dopis do upravljavca – zahteva po podatkih, s katerimi moramo biti seznanjeni po zakonu
- 60 zahtev – 43 odgovorov

AS Abakus
As na disku.

Končne ugotovitve

- da lahko ob poznavanju davčne številke nekatere OP dobimo kar s telefonskim klicem (DURS - prihodki in davčni dolg)
- da je na nivoju procesov in papirnatih zbirk več reda, kar pa se tiče računalniških zbirk in računalniške podpore obdelavi OP pa pravniki, ki ponavadi odgovarjajo na zahteve glede OP, niso usposobljeni ali seznanjeni z zbirkami
- da je le eden od upravljavcev poimensko navedel imena zaposlenih, ki so vpogledovali v njihove OP

Informacijska pooblaščenka

- priporočila, predstavljena v primeru:
  - IS v zdravstvu, iskanje po kriteriju »Novak« vrne 100 pacientov
  - mediji objavijo, da ima znana oseba AIDS
  - kakšne so možnosti za zagotavljanje sledljivosti?

Zagotavljanje sledljivosti

- dve možnosti
- **možnost 1**:
  takšna poizvedba se zabeleži pri vsakem od pacientov, katerega osebni podatki so bili prikazani v določenem trenutku na izhodni enoti. Pri posamezniku se mora v tem primeru zabeležiti, kdo in kdaj je s pomočjo poizvedbe prišel do osebnih podatkov posameznika.

Zagotavljanje sledljivosti-2

- **možnost 2**:
  poizvedba mora dati ob ponovitvi enak rezultat, torej se beleži rezultat poizvedbe v času. Zabeležiti se mora, kdo je izvedel poizvedbo, kdaj in kakšen je bil rezultat poizvedbe v tem času.

Zagotavljanje sledljivosti-3

- če ni možno zagotoviti naknadnega ugotavljanja, do katerih OP je na ta način uporabnik dostopil (do nekaterih ali do vseh), je potrebno predvidevati, da se je uporabnik s poizvedbo seznanil z vsemi prikazanimi osebnimi podatki in temu ustrezno to zabeležiti v dnevniški zapis.

Možnosti upravljavca

- aplikacija sama skrbi za revizijsko sled (RS)
- izkoristimo možnosti podatkovne zbirke (vklop AUDIT_TRAIL)
- neodvisno orodje za vodenje revizjskih sledi (RS)

Vgrajeno v aplikacijo

- prednosti:
  - točno se ve, za katere OP je treba voditi RS in kakšen je namen vpogleda
- slabosti:
  - dopolnitve aplikacije, upoštevanje sprememb
  - ni nadzora administratorjev
  - ni nadzora nad drugimi orodji za dostop
  - degradacija performans, diskovni prostor
  - »stranska vrata« v aplikacijah
  - OP in RS sta v isti bazi!
  - ne zna zanesljivo odgovoriti na primer IP

Podatkovna zbirka

- vklop AUDIT_TRAIL, kreiranje triggerjev
- prednosti:
  - podatkovna zbirka sama skrbi za RS
- slabosti:
  - programiranje triggerjev, spremembe
  - administratorji lahko potvorijo RS
  - degradacija performans, diskovni prostor
  - ne zaveda se namena zbiranja OP
  - OP in RS sta v isti bazi!
  - ne zna zanesljivo odgovoriti na primer IP

Orodje za vodenje RS

- prednosti:
  - ni sprememb aplikacij
  - RS je (navadno) ločena od OP
  - administrator ne more potvoriti RS
  - minimalna degradacija performans, ni naraščanja velikosti baze OP
- slabosti:
  - ne zaveda se namena zbiranja OP
  - ne zna zanesljivo odgovoriti na primer IP

Katera rešitev ustreza ZVOP-u?

- select * from delavci where priimek = 'Novak';
- nobena od rešitev ne zadosti 1. in 2. možnosti IP
- najbližje je namensko orodje:
  - beleži RS poizvedb drugih orodij
  - onemogoča potvarjanje RS
  - baza RS je ločena od baze OP

Kje so težave?

- select * from delavci where priimek = 'Novak';

```
CREATE TABLE delavci (
    delavec_id INT NOT NULL PRIMARY KEY,
    ime VARCHAR2 (30) NOT NULL,
    priimek VARCHAR2 (30) NOT NULL,
    davcna_st VARCHAR2 (30) NOT NULL,
    naslov VARCHAR2 (50) NOT NULL,
    op_3 VARCHAR2 (10),
    op_4 VARCHAR2 (10)
);
```

Kje so težave?-2

```
INSERT INTO delavci VALUES (1, 'Igor', 'Novak', '99228811', 'Zagorje 13', '3', '4');
INSERT INTO delavci VALUES (2, 'Brane', 'Novak', '99338817', 'Sempeter 3', '32',
'4');
INSERT INTO delavci VALUES (3, 'Stanko', 'Novak', '99448816', 'Zgornje Gorje 1',
'AZ', '4');
INSERT INTO delavci VALUES (4, 'Miha', 'Novak', '99558815', 'Breg 43', 'EP', '4');
INSERT INTO delavci VALUES (5, 'Peter', 'Novak', '99668818', 'Dolenja vas 1', 'LJ',
'4');
INSERT INTO delavci VALUES (6, 'Igor', 'Kastelic', '99245811', 'Zapoge 8', 'BO',
'4');
INSERT INTO delavci VALUES (7, 'Roman', 'Mrak', '9922866', 'Zagornji kihovci 5',
'KR', '4');
INSERT INTO delavci VALUES (8, 'Sergej', 'Kovac', '99228832', 'Za potokom 3', 'KK',
'4');
INSERT INTO delavci VALUES (9, 'Boris', 'Grohar', '99228842', 'Most 9', '3', 'C');
INSERT INTO delavci VALUES (10, 'Jernej', 'Kavka', '99228844', 'Smrekov gozd 198',
'AA', 'X');
INSERT INTO delavci VALUES (11, 'Branko', 'Kopitar', '99228859', 'Bitnje 7', '33',
'34');
```

```
AUDIT ALL ON delavci;
```

Kje so težave?-3

```
select delavec_id, ime, priimek from delavci where priimek = 'Novak'

DELAVEC_ID IME                            PRIIMEK
---------- ------------------------------ ------------------------------
         1 Igor                           Novak
         2 Brane                          Novak
         3 Stanko                         Novak
         4 Miha                           Novak
         5 Peter                          Novak

--
SELECT username, timestamp, sql_text FROM dba_audit_trail dat WHERE dat.owner =
'SCOTT' AND dat.obj_name = 'DELAVCI';

USERNA TIMESTAMP           SQL_TEXT
------ ------------------- ----------------------------------------
SCOTT  2012-10-07 10:38:24 select delavec_id, ime, priimek from del
                           avci where priimek = 'Novak'

-- dodajanje zapisa? brisanje zapisa?
```

Kje so težave?-4

```
ALTER TABLE delavci ADD datum_kreiranja DATE;

--
-- če zanemarimo spremembe, lahko potem najdemo vse zapise, ki jih je sprožil
-- select delavec_id, ime, priimek from delavci where priimek = 'Novak'
-- Pogoj je, da prepovemo brisanje delavcev (kar niti ni tako nenavadno)

SELECT delavec_id, ime, priimek
  FROM delavci
 WHERE priimek = 'Novak'
   AND datum_kreiranja <= <datum>;
```

Lahko vedno vrnemo isto vsebino?

```
CREATE GLOBAL TEMPORARY TABLE gtt$delavci_ids (
   delavec_id INT
);

INSERT INTO gtt$delavci_ids VALUES (1);
INSERT INTO gtt$delavci_ids VALUES (2);
INSERT INTO gtt$delavci_ids VALUES (3);

SELECT d.delavec_id, d.ime, d.priimek
  FROM delavci d,
       gtt$delavci_ids di
 WHERE d.delavec_id = di.delavec_id;

DELAVEC_ID IME                            PRIIMEK
---------- ------------------------------ ------------------------------
         1 Igor                           Novak
         2 Brane                          Novak
         3 Stanko                         Novak
```

Označiti vse zapise

```
-- naredimo audit tabelo
CREATE TABLE delavci_aud (
   delavec_id INT NOT NULL,
   ts TIMESTAMP NOT NULL
);

-- naredimo funkcijo z avtonomno transakcijo
CREATE OR REPLACE FUNCTION delavci_id (p_delavec_id IN delavci.delavec_id%TYPE)
RETURN delavci.delavec_id%TYPE IS
   PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
   INSERT INTO delavci_aud VALUES (p_delavec_id, SYSTIMESTAMP);
   COMMIT;
   RETURN (p_delavec_id);
END;

-- kreiramo view, v katerem uporabimo funkcijo
CREATE OR REPLACE VIEW scott.delavci_vw AS
   SELECT delavci_id (d.delavec_id) delavec_id,
          d.ime, d.priimek, d.davcna_st, d.naslov
     FROM delavci d;
```

Označiti vse zapise-2

```
select delavec_id, ime, priimek from delavci_vw where priimek = 'Novak';

DELAVEC_ID IME                            PRIIMEK
---------- ------------------------------ ------------------------------
         1 Igor                           Novak
         2 Brane                          Novak
         3 Stanko                         Novak
         4 Miha                           Novak
         5 Peter                          Novak

select * from delavci_aud;

DELAVEC_ID TO_CHAR(TS,'DD.MM.Y
---------- -------------------
         1 07.10.2012 10:53:23
         2 07.10.2012 10:53:23
         3 07.10.2012 10:53:23
         4 07.10.2012 10:53:23
         5 07.10.2012 10:53:23
```

Označiti vse zapise-3

```
SQL> truncate table delavci_aud;
SQL> set arraysize 1
SQL> set pagesize 5
SQL> set pause on;
SQL> select delavec_id, ime, priimek from delavci_vw where priimek = 'Novak';

DELAVEC_ID IME                            PRIIMEK
---------- ------------------------------ ------------------------------
         1 Igor                           Novak
         2 Brane                          Novak


         3 Stanko                         Novak
ERROR:
ORA-01013: user requested cancel of current operation

SQL> select * from delavci_aud;

DELAVEC_ID TO_CHAR(TS,'DD.MM.Y
---------- --------------------
         1 07.10.2012 11:03:29
         2 07.10.2012 11:03:29
         3 07.10.2012 11:03:29
```

Označiti vse zapise-4

```
--
-- Kaj pa ta poizvedba? Zapiše v delavec_aud podatke?
--
select ime, priimek from delavci_vw where priimek = 'Novak';

CREATE OR REPLACE TYPE delavci_rec_t IS OBJECT (
   delavec_id INT,
   ime        VARCHAR2 (30),
   priimek    VARCHAR2 (30),
   davcna_st  VARCHAR2 (30),
   naslov VARCHAR2 (50),
   op_3 VARCHAR2 (10),
   op_4 VARCHAR2 (10),
   datum_kreiranja date
);

CREATE OR REPLACE TYPE delavci_tab_t AS TABLE OF delavci%ROWTYPE
/
```

Označiti vse zapise-5

```
CREATE OR REPLACE PACKAGE delavci_pkg AS
   FUNCTION get_delavci (p_delavec_id IN delavci.delavec_id%TYPE)
      RETURN delavci_tab_t PIPELINED;
END;

CREATE OR REPLACE PACKAGE  BODY delavci_pkg AS

   FUNCTION get_delavci (p_delavec_id IN delavci.delavec_id%TYPE)
      RETURN delavci_tab_t PIPELINED IS
      PRAGMA AUTONOMOUS_TRANSACTION;
      x_rec delavci%ROWTYPE;
   BEGIN
      SELECT * INTO x_rec from delavci WHERE delavec_id = p_delavec_id;
      INSERT INTO delavci_aud VALUES (x_rec.delavec_id, SYSTIMESTAMP);
      COMMIT;
      PIPE ROW (delavci_rec_t (
                  x_rec.delavec_id, x_rec.ime, x_rec.priimek,
                  x_rec.davcna_st, x_rec.naslov,
                  x_rec.op_3, x_rec.op_4, x_rec.datum_kreiranja));
   END;

END;
```

Označiti vse zapise-6

```
CREATE OR REPLACE VIEW delavci_all_vw AS
SELECT d.*
  FROM delavci d, TABLE (delavci_pkg.get_delavci (d.delavec_id)) p
   WHERE d.delavec_id = p.delavec_id;

SQL> SELECT ime, priimek from delavci_all_vw WHERE priimek = 'Novak';

IME                            PRIIMEK
------------------------------ ------------------------------
Igor                           Novak
Brane                          Novak
Stanko                         Novak
Miha                           Novak
Peter                          Novak

SQL> select delavec_id, to_char (ts, 'dd.mm.yyyy hh24:mi:ss') from delavci_aud;

DELAVEC_ID TO_CHAR(TS,'DD.MM.Y
---------- -------------------
         1 07.10.2012 11:20:41
         2 07.10.2012 11:20:41
         3 07.10.2012 11:20:41
         4 07.10.2012 11:20:41
         5 07.10.2012 11:20:41
```

Je možna prevara?

- administratorji še vedlo lahko vzamejo podatke direktno iz tabel
- select ime, priimek from ... še vedno označi, da so bili gledani OP posameznika

Šifriranje podatkov

- vse OP šifrirati
- vnos kode za vpogled v posameznikove OP
- zapis vpogleda v funkciji, ki dešifrira podatke
- dešifrirati podatek samo v eni seji in samo za en zapis naenkrat

Šifriranje podatkov-2

```
--
-- kriptirati podatke davcna_st, naslov, op_3, op_4
--
CREATE TABLE delavci_cr (
   delavec_id INT NOT NULL PRIMARY KEY,
   ime VARCHAR2 (30) NOT NULL,
   priimek VARCHAR2 (30) NOT NULL,
   davcna_st RAW (32) NOT NULL,
   naslov RAW (50) NOT NULL,
   op_3 RAW (32),
   op_4 RAW (32)
);


CREATE GLOBAL TEMPORARY TABLE gtt$delavci_cr (
   delavec_id INT NOT NULL PRIMARY KEY,
   decrypted NUMBER (1) DEFAULT 0 NOT NULL
) ON COMMIT PRESERVE ROWS;
```

Šifriranje podatkov-3

```
CREATE OR REPLACE PACKAGE mycrypt IS
   PROCEDURE setkey(p_key IN VARCHAR2);
   FUNCTION encrypt(p_data IN VARCHAR2) RETURN RAW DETERMINISTIC;
   FUNCTION decrypt(p_data IN RAW) RETURN VARCHAR2 DETERMINISTIC;
END;
/

CREATE OR REPLACE PACKAGE BODY mycrypt IS

   g_charset VARCHAR2(30) := 'AL32UTF8';
   ecryption_type PLS_INTEGER := sys.dbms_crypto.encrypt_aes128 +
                                 sys.dbms_crypto.chain_cbc +
                                 sys.dbms_crypto.pad_pkcs5;
   g_key          RAW(16) := NULL;
```

Šifriranje podatkov-4

```
PROCEDURE setkey(p_key IN VARCHAR2) IS
BEGIN
   IF p_key IS NOT NULL
   THEN
      g_key := utl_raw.cast_to_raw(p_key);
   END IF;
END setkey;

FUNCTION encrypt(p_data IN VARCHAR2) RETURN RAW DETERMINISTIC IS
   l_data      RAW(2048) := utl_i18n.string_to_raw(p_data, g_charset);
   l_encrypted RAW(2048);
BEGIN
   l_encrypted := sys.dbms_crypto.encrypt(
       src => l_data,
       typ => ecryption_type,
       key => g_key);
   RETURN l_encrypted;
EXCEPTION
   WHEN OTHERS THEN
      RAISE;
      -- for the security reason I want to completely silence the error
      -- stack that could reveal some technical details to imaginary attacker.
      -- Remember, such miss-use of WHEN OTHERS should be considered
      -- as a bug in almost all other situations.
      raise_application_error(-20001, 'Access denied!');
END encrypt;
```

Šifriranje podatkov-5

```
   FUNCTION decrypt(p_data IN RAW) RETURN VARCHAR2 DETERMINISTIC IS
      l_decrypted RAW(2048);
   BEGIN
      l_decrypted := sys.dbms_crypto.decrypt(
          src => p_data,
          typ => ecryption_type,
          key => g_key);
      RETURN utl_i18n.raw_to_char(l_decrypted, g_charset);
   EXCEPTION
      WHEN OTHERS THEN
         raise_application_error(-20001, 'Access denied!');
   END decrypt;
END;
```

Šifriranje podatkov-6

```
DECLARE
   dcr_rec delavci_cr%ROWTYPE;
BEGIN
   mycrypt.setkey (p_key => 'geslo67890123456');
   FOR x_rec IN (SELECT * FROM delavci)
   LOOP
      dcr_rec.delavec_id := x_rec.delavec_id;
      dcr_rec.ime := x_rec.ime;
      dcr_rec.priimek := x_rec.priimek;
      dcr_rec.davcna_st := mycrypt.encrypt (x_rec.davcna_st);
      dcr_rec.naslov := mycrypt.encrypt (x_rec.naslov);
      dcr_rec.op_3 := mycrypt.encrypt (x_rec.op_3);
      dcr_rec.op_4 := mycrypt.encrypt (x_rec.op_4);
      INSERT INTO delavci_cr VALUES dcr_rec;
   END LOOP;
   COMMIT;
END;
```

Šifriranje podatkov-6

```
SQL> select delavec_id, ime, priimek, davcna_st from delavci_cr;

DELAVEC_ID IME        PRIIMEK    DAVCNA_ST
---------- ---------- ---------- --------------------------------
         1 Igor       Novak      FC10953EC6943CE824566DE8ECAAD0E9
         2 Brane      Novak      E0EFB60B167569759CDFF87C3802C4EC
         3 Stanko     Novak      AAD78D2907D4BAD6B36BDE3BAF099DC6
         4 Miha       Novak      8BE28DE3CB285B8A93244F849CAF0F3F
         5 Peter      Novak      E23A5F083D45B0AA59DCE71758D473FB
         6 Igor       Kastelic   C8EC492D1F7581D95BED715F0A1917A1
         7 Roman      Mrak       B007A86DA121AA93D6074EFA20EE1CE5
         8 Sergej     Kovac      A5FBD477729506E063CA717ADE3488FC
         9 Boris      Grohar     0FFE22427F72C70D8E1FF0E405138D2B
        10 Jernej     Kavka      AD1FDC2032CBECB12F8F8579A37AC714
        11 Branko     Kopitar    82560DFC3758A6F506F9E4B5469BCA90
```

Dešifriranje podatkov

```
CREATE OR REPLACE PACKAGE delavci_crypt AS

   PROCEDURE decrypt(p_delavec_id IN delavci.delavec_id%TYPE,
                     p_key        IN VARCHAR2);
   FUNCTION get_delavci(p_delavec_id IN delavci.delavec_id%TYPE) RETURN
delavci_tab_t
      PIPELINED;

END;

CREATE OR REPLACE PACKAGE BODY delavci_crypt AS

   PROCEDURE decrypt(p_delavec_id IN delavci.delavec_id%TYPE,
                     p_key        IN VARCHAR2) IS
   BEGIN
      mycrypt.setkey(p_key => p_key);
      BEGIN
         INSERT INTO gtt$delavci_cr VALUES (p_delavec_id, 1);
      EXCEPTION
         WHEN dup_val_on_index THEN
            NULL;
      END;
   END;
```

Dešifriranje podatkov-2

```
PROCEDURE insert_delavci_vpogled (p_delavec_id IN delavci.delavec_id%TYPE) IS
   PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
   INSERT INTO delavci_aud
   VALUES (p_delavec_id, systimestamp);
   COMMIT;
END;
```

Dešifriranje podatkov-3

```
   FUNCTION get_delavci(p_delavec_id IN delavci.delavec_id%TYPE)
      RETURN delavci_tab_t PIPELINED IS
      d_rec delavci%ROWTYPE;
      l_decrypted gtt$delavci_cr.decrypted%TYPE := 0;
   BEGIN
      BEGIN
         SELECT decrypted INTO l_decrypted
           FROM gtt$delavci_cr WHERE delavec_id = p_delavec_id;
      EXCEPTION
         WHEN no_data_found THEN
            l_decrypted := 0;
      END;
      SELECT d.delavec_id, d.ime, d.priimek,
             decode (l_decrypted, 0, NULL, mycrypt.decrypt(d.davcna_st)) davcna_st,
             decode (l_decrypted, 0, NULL, mycrypt.decrypt(d.naslov)) naslov,
             decode (l_decrypted, 0, NULL, mycrypt.decrypt(d.op_3)) op_3,
             decode (l_decrypted, 0, NULL, mycrypt.decrypt(d.op_4)) op_4
        INTO d_rec
        FROM delavci_cr d, gtt$delavci_cr gttd
       WHERE d.delavec_id = p_delavec_id
         AND gttd.delavec_id (+) = d.delavec_id;
      IF l_decrypted = 1 THEN
         insert_delavci_vpogled (p_delavec_id);
      END IF;
      PIPE ROW(delavci_rec_t(d_rec.delavec_id,
                             d_rec.ime,
                             d_rec.priimek,
                             d_rec.davcna_st,
                             d_rec.naslov,
                             d_rec.op_3,
                             d_rec.op_4));
   END;
END;
```

Dešifriranje podatkov-4

```
CREATE OR REPLACE VIEW DELAVCI_ALL_VW AS
SELECT d."DELAVEC_ID",d."IME",d."PRIIMEK",p."DAVCNA_ST",p."NASLOV",p."OP_3",p."OP_4"
  FROM delavci d,
       TABLE (delavci_crypt.get_delavci (d.delavec_id)) p
   WHERE d.delavec_id = p.delavec_id;

SQL> select delavec_id, ime, priimek, davcna_st from delavci_all_vw;

  ID IME        PRIIMEK    DAVCNA_ST
---- ---------- ---------- ---------------------------------
   1 Igor       Novak
   2 Brane      Novak
   3 Stanko     Novak
   4 Miha       Novak
   5 Peter      Novak
   6 Igor       Kastelic
   7 Roman      Mrak
   8 Sergej     Kovac
   9 Boris      Grohar
  10 Jernej     Kavka
  11 Branko     Kopitar

11 rows selected.
```

Dešifriranje podatkov-5

```
SQL> BEGIN
   delavci_crypt.decrypt (1, 'geslo67890123456');
   delavci_crypt.decrypt (4, 'geslo67890123456');
   delavci_crypt.decrypt (6, 'geslo67890123456');
END;
/
PL/SQL procedure successfully completed.

SQL> select delavec_id, ime, priimek, davcna_st from delavci_all_vw;

  ID IME        PRIIMEK    DAVCNA_ST
---- ---------- ---------- --------------------------------
   1 Igor       Novak      99228811
   2 Brane      Novak
   3 Stanko     Novak
   4 Miha       Novak      99558815
   5 Peter      Novak
   6 Igor       Kastelic   99245811
   7 Roman      Mrak
   8 Sergej     Kovac
   9 Boris      Grohar
  10 Jernej     Kavka
  11 Branko     Kopitar

11 rows selected.
```

Kaj je prav?

- ni enostavne rešitve
- najbolje je kombinacija
  - neodvisna aplikacija za vodenje RS
  - šifriranje OP v tabelah
  - ločena baza RS za OP
- vse ostalo so delne rešitve

AS Abakus
As na disku.

ORA-03113: end-of-file on communication channel

Boris Oblak
Abakus plus d.o.o.

ORACLE CERTIFIED PROFESSIONAL

ORACLE Gold Partner

17. Strokovno srečanje
SIOUG 2012
Kongresni center Hotel Mons Ljubljana, 15. - 17. oktober

SIOUG Slovensko društvo Oracle uporabnikov

Kdo vse je videl podatke Marije Novak?