TWIN CITIES ORACLE Users Group

Oracle Security

Spring 2018

Daniel A. Morgan
email: vp@tcoug.org
mobile: +1 206-669-2949

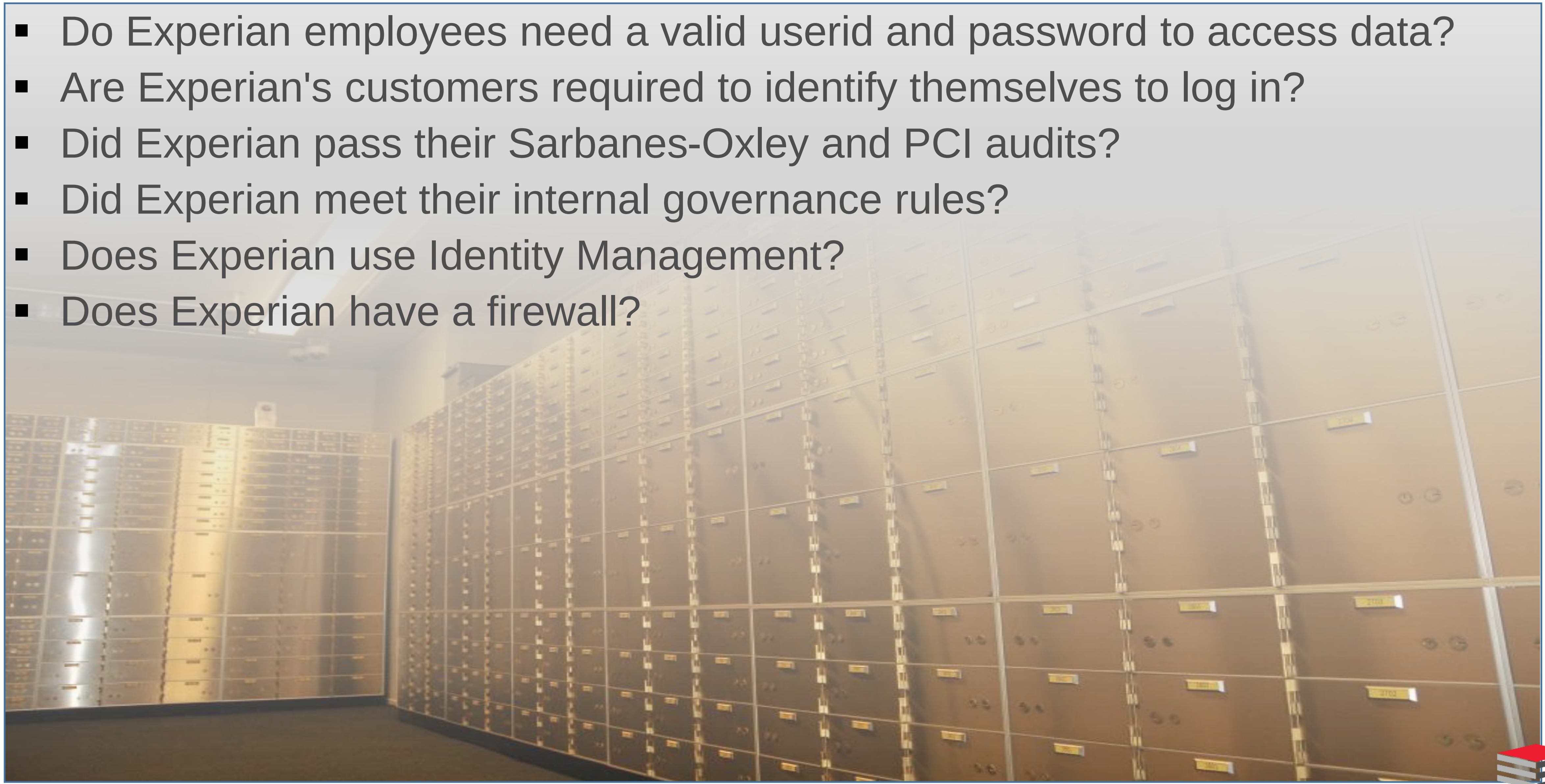
Wednesday: April 18, 2018

Oracle Fine Grained Security

Dan Morgan @ Sirius Meta7

TWIN CITIES ORACLE USERS GROUP

How The War Is Being Waged: Case 2 (1:3)



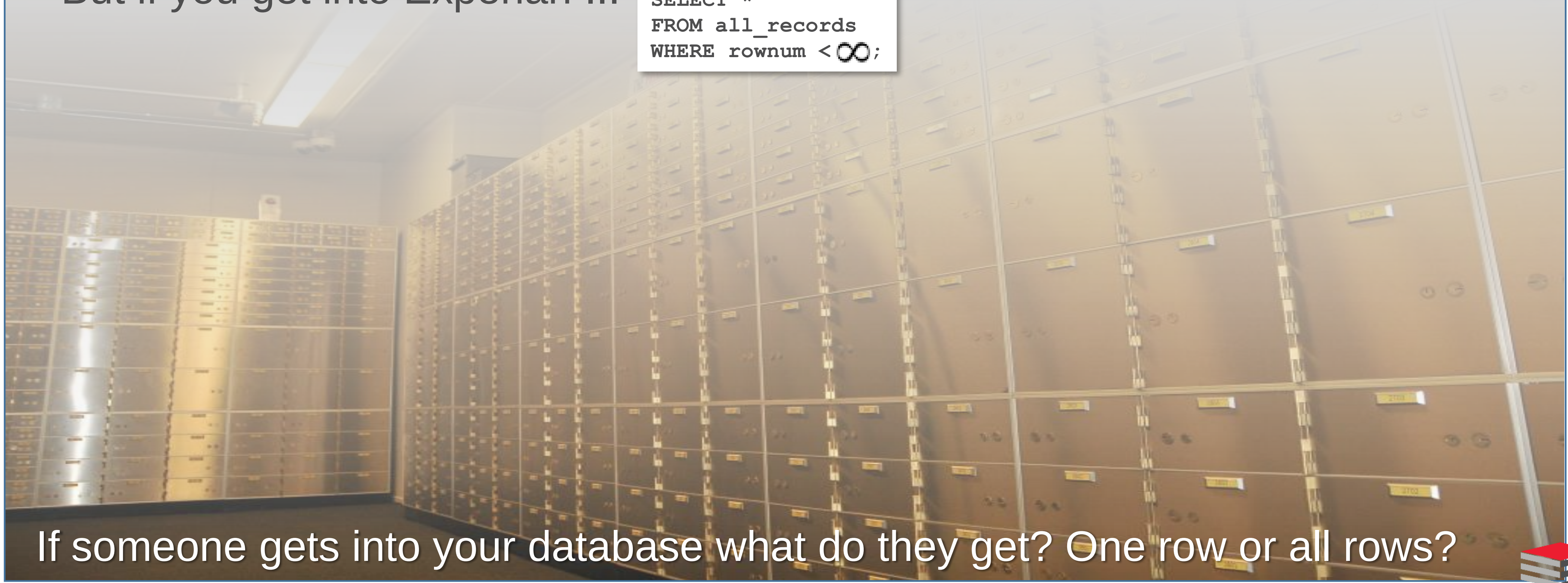
- Do Experian employees need a valid userid and password to access data?
- Are Experian's customers required to identify themselves to log in?
- Did Experian pass their Sarbanes-Oxley and PCI audits?
- Did Experian meet their internal governance rules?
- Does Experian use Identity Management?
- Does Experian have a firewall?

TWIN CITIES ORACLE USERS GROUP

How The War Is Being Waged: Case 2 (3:3)

- Every bank has a front door with a lock
- Every bank also has a vault with a separate door and its own lock
- If you get into a bank vault you don't get access to every safe deposit box
- But if you get into Experian ...

```
SELECT *
FROM all_records
WHERE rownum < ∞;
```



If someone gets into your database what do they get? One row or all rows?

TWIN CITIES ORACLE USERS GROUP

A Perry Mason Moment (1:2)

- Do you promise to tell the truth?
  The whole truth?
  And nothing but the truth?
  ~the bailiff

  I do
  ~ Dan Morgan

- If you are from Oracle Corp. you should be sitting down for the next 45 minutes or leave the room to make a phone call ... this will be the truth

A Perry Mason Moment (2:2)

- Most company's security products have very little, if anything, to do with security and not one of them, by itself, is capable of deploying real data security
- These products exist almost entirely for the purpose of making auditors go away
- The pressure is to pass the audit ... Sarbanes Oxley, Grahm Leach Bliley, HIPAA, PCI DSS, ...

A Perry Mason Moment

- Do you need to purchase these products and pass these audits?

Absolutely YES!

- Does passing these audits enhance your security?

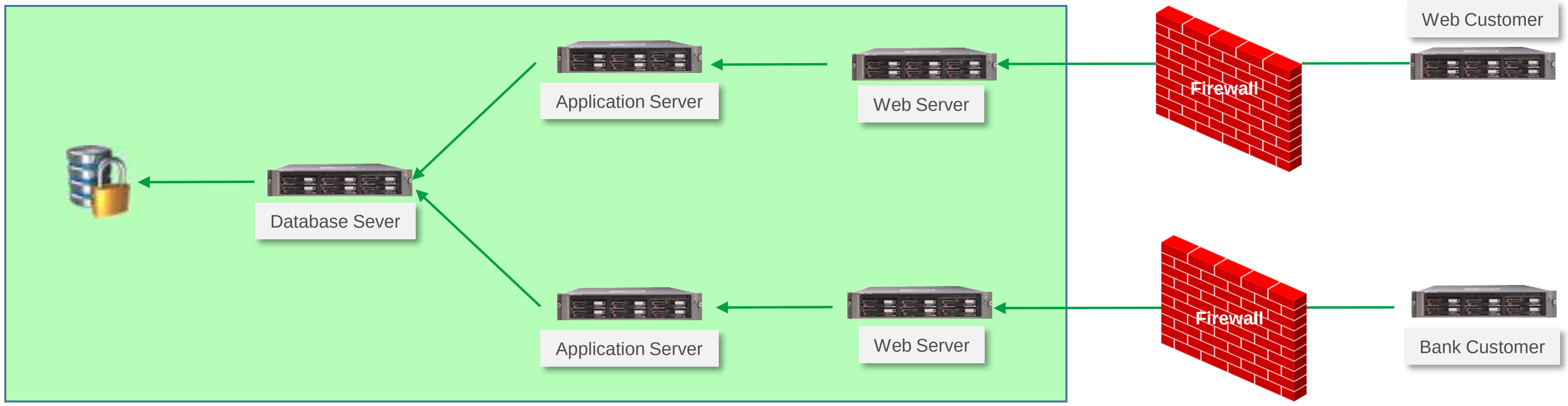
Marginally

Moving the Focus to Real Data Security

- Real data security assumes that bad actors are already in your database
  - They have a valid user-id
  - They have a valid password
  - They have a spot in your Active Director / Identity Management / LDAP system
  - They have a full access through your organization's VPN
- Real data security address the question of how do I/we protect data when the threat has a valid user-id and password?
- Here's one way to do it and one that can be made to work with almost all commercially available applications without changing a single line of code

The Architecture

- Let's assume we are doing security architecture for Experian and we have two types of customers
  - Individual consumers (web customer) who want to look at their own data
  - Corporate customers (bank customer) that wants to look at multiple consumer's data
- There are two distinct network paths to the database ... different subnets



Let's Set Up The Application Owners

- We will use two separate schemas ...
  - Experian to persistent credit card and other application sensitive data
    - It has the following privileges
      - CREATE SESSION
      - CREATE TABLE
  - SecAccess, which can be created without touching a single line of application code and provides a layer separating the data that needs to be protected from everyone and everything outside
    - It has the following privileges
      - CREATE SESSION
      - CREATE ANY CONTEXT
      - CREATE PROCEDURE
      - CREATE TYPE
      - CREATE VIEW

```
CREATE USER experian
IDENTIFIED BY E1x2p3e4r5i6a7n$
DEFAULT TABLESPACE uwdata
TEMPORARY TABLESPACE temp
PROFILE ora_stig_profile
QUOTA 0 ON system
QUOTA 0 ON sysaux
QUOTA 100M ON uwdata;

ALTER USER experian ENABLE EDITIONS;

GRANT create session TO experian;
GRANT create table TO experian;

CREATE USER secaccess
IDENTIFIED BY S1e2c3a4c5c6e7s8s$
PROFILE ora_stig_profile;
-- note secaccess gets no default or temporary tablespace

ALTER USER experian ENABLE EDITIONS;

GRANT create session TO secaccess;
GRANT create any context TO secaccess;
GRANT create procedure TO secaccess;
GRANT create type TO secaccess;
GRANT create view TO secaccess;
```

TWIN CITIES ORACLE USERS GROUP

Let's Set Up The Application Users

- There are two classes of users
  - Individuals (Web Customers) that need to look at their own data ... we will let them only see 3 lines of data no matter what SQL statement they write
  - Organizations (Bank Customers) that need to look at data belonging to groups of customers ... we will let them see only 12 lines of data no matter what SQL statement they write
  - Both user types get only one privilege
    - CREATE SESSION
- But you will see, if you look carefully that they are proxy users and they are audited for everything they do after they connect to the secure access layer
- There is almost no excuse for not making all connections as a proxy user

```
CREATE USER webcust
IDENTIFIED BY webcust
TEMPORARY TABLESPACE temp
PROFILE DEFAULT;

CREATE USER bankcust
IDENTIFIED BY bankcust
TEMPORARY TABLESPACE temp
PROFILE DEFAULT;

GRANT create session TO webcust;
GRANT create session TO bankcust;

-- the following is the proxy user auditing an connection
AUDIT CONNECT BY webcust ON BEHALF OF secaccess;
ALTER USER secaccess GRANT CONNECT THROUGH webcust;

AUDIT CONNECT BY bankcust ON BEHALF OF secaccess;
ALTER USER secaccess GRANT CONNECT THROUGH bankcust;
```

Let's Capture Login Information for Auditing

- First we create the audit table so that we can track
  - Application user login date+time
  - Database login name
  - Proxy login name
  - Database schema accessed
- The AFTER LOGON ON DATABASE trigger grabs this information and persists it
- In a non-demo environment capture
  - Client IP Address
  - Client Host Name
  - Application Name
  - and a lot more to help monitor usage

```
-- create login audit table
CREATE TABLE experian.app_audit (
login_date  TIMESTAMP WITH LOCAL TIME ZONE,
user_name   VARCHAR2(30),
proxy_name  VARCHAR2(30),
schema_name VARCHAR2(30));

GRANT insert ON experian.app_audit TO webcust, bankcust;

-- create after logon trigger
CREATE OR REPLACE TRIGGER experian.audit_app_cnx
AFTER LOGON ON DATABASE
DECLARE
 PRAGMA AUTONOMOUS_TRANSACTION;
 cur_user user_users.username%TYPE := sys_context('USERENV',
 'CURRENT_USER');
BEGIN
  dbms_application_info.set_client_info(cur_user);

  INSERT INTO app_audit
  (login_date, user_name, proxy_name, schema_name)
  VALUES
  (SYSTIMESTAMP, cur_user, sys_context('USERENV',
  'PROXY_USER'), sys_context('USERENV', 'CURRENT_SCHEMA'));
  COMMIT;
END audit_app_cnx;
/
```

TWIN CITIES ORACLE USERS GROUP

Then Build The Application

- The application for demo purposes consists of a single table that has PII and PCI data
  - Note that every column in the table contains sensitive data

```
CREATE TABLE credit_info_base (
ssn             VARCHAR2(11),
cc_number       VARCHAR2(19),
last_name       VARCHAR2(15),
first_name      VARCHAR2(15),
dob             DATE,
gender          VARCHAR2(1),
cc_exp_date     VARCHAR2(4),
cc_sec_code     VARCHAR2(4))
PCTFREE 0
TABLESPACE uwdata;

ALTER TABLE credit_info_base
ADD CONSTRAINT pk_credit_info_base
PRIMARY KEY (ssn, cc_number);
```

- And grant only a single read-only privilege to the data access layer

```
GRANT select ON experian.credit_info_base TO secaccess;
```

Security Layer Set-Up

- The ciprep_ctx package contains a single procedure that sets an Oracle object called a **context** in database memory

```
CREATE OR REPLACE PACKAGE ciprep_ctx AUTHID DEFINER IS
 PROCEDURE set_ctx(ssn_in IN VARCHAR2);
END ciprep_ctx;
/

CREATE OR REPLACE PACKAGE BODY ciprep_ctx IS
 PROCEDURE set_ctx(ssn_in IN VARCHAR2) IS
  BEGIN
    dbms_session.set_context('ci_env', 'ssn_ctx', ssn_in);
 END set_ctx;
END ciprep_ctx;
/

CREATE OR REPLACE CONTEXT ci_env USING secaccess.ciprep_ctx;
```

- The application will not be accessed by end users getting DML table access privs but rather through a view built upon a secure editioning view
- Oracle guarantees zero performance degradation when accessing an Editioning View

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "SECACCESS"."CREDIT_INFO" (
"SSN","CC_NUMBER","LAST_NAME","FIRST_NAME","DOB","GENDER","CC_EXP_DATE","CC_SEC_CODE") AS
SELECT "SSN","CC_NUMBER","LAST_NAME","FIRST_NAME","DOB","GENDER","CC_EXP_DATE","CC_SEC_CODE"
FROM experian.credit_info_base;
```

Application Data Access Objects (1:2)

- Application customers will never touch the application schema (Experian) nor will they touch the editioning view in the secure access layer (SecAccess) rather they will access data through a view built on top of not a statis SQL statement but rather an dynamic pipelined table function (PTF)
- To build the PTF we build data types and then a PL/SQL function and then a view built on top of the PTF

```
CREATE OR REPLACE TYPE credit_info_type AUTHID DEFINER AS
OBJECT(
ssn          VARCHAR2(11),
cc_number    VARCHAR2(19),
last_name    VARCHAR2(15),
first_name   VARCHAR2(15),
dob          DATE,
gender       VARCHAR2(1),
cc_exp_date  VARCHAR2(4),
cc_sec_code  VARCHAR2(4));
/

CREATE OR REPLACE TYPE credit_info_TypeSet AS TABLE OF
credit_info_type;
/
```

Application Data Access Objects (2:2)

```
CREATE OR REPLACE PACKAGE refcur_pkg AUTHID DEFINER IS
 TYPE refcur_t IS REF CURSOR RETURN credit_info%ROWTYPE;
END refcur_pkg;
/
```

```
CREATE OR REPLACE VIEW rciv AS SELECT * FROM TABLE(rci(CURSOR(SELECT * FROM credit_info)));
```

Application Data Access Objects (2:2)

```
CREATE OR
 TYPE ref
END refcu
/
```

```
CREATE OR REPLACE FUNCTION rci(p refcur_pkg.refcur_t) RETURN credit_info_TypeSet PIPELINED AUTHID DEFINER IS
 in_rec    p%ROWTYPE;
 out_rec   credit_info_type := credit_info_type(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);
 cntr      PLS_INTEGER := 0;
 cur_limit PLS_INTEGER;
 cur_match VARCHAR2(19);
BEGIN
  cur_match := (sys_context('ci_env', 'ssn_ctx'));
  IF sys_context('USERENV', 'PROXY_USER') = 'WEBCUST' THEN cur_limit := 3;
  ELSIF sys_context('USERENV', 'PROXY_USER') = 'BANKCUST' THEN cur_limit := 12;
  ELSIF sys_context('USERENV', 'CURRENT_USER') = 'SECACCESS' THEN cur_limit := 999999999;
  ELSE cur_limit := 0;
  END IF;
  LOOP
    FETCH p INTO in_rec;
    EXIT WHEN p%NOTFOUND;
    IF in_rec.ssn = cur_match THEN
      cntr := cntr + 1;
      out_rec.ssn := in_rec.ssn;
      out_rec.cc_number := in_rec.cc_number;
      ...
      PIPE ROW(out_rec);
    END IF;
    IF cntr >= cur_limit THEN EXIT; END IF;
  END LOOP;
  CLOSE p;
  RETURN;
END rci;
/
```

```
CREATE OR
```

Let The End-Users Attack

- The application now log in proxy users to the secaccess schema and gain access to the rciv view (in the real world I would use a third schema layer to separate rciv from the schema with real access to the Experian application
- To get 100% of the source code for this application, and you can build it in10gR2 or above, join our TCOUG Slack group and I will post the URL there
- If you run it you will see that webcust, no matter what query it writes, can only view data associated with a single SSN and never more than 3 rows
- Bankcust, no matter what SQL it writes can only see a maximum of 12 rows associated, again, with a single SSN
- It is the rules written into the RCI function that controls what data is available in the RCIV view

```
conn webcust[SECACCESS]/webcust@pdbdev

exec ciprep_ctx.set_ctx('545-98-1234');
SELECT * FROM rciv;

exec ciprep_ctx.set_ctx('618-45-2345');
SELECT * FROM rciv;

exec ciprep_ctx.set_ctx('795-61-3457');
SELECT * FROM rciv;

SELECT * FROM rciv;

conn bankcust[SECACCESS]/bankcust@pdbdev

exec ciprep_ctx.set_ctx('545-98-1234');
SELECT * FROM rciv;

exec ciprep_ctx.set_ctx('618-45-2345');
SELECT * FROM rciv;

exec ciprep_ctx.set_ctx('795-61-3457');
SELECT * FROM rciv;
```

TWIN CITIES ORACLE USERS GROUP

An IT Terrorist Attack (7:7)

- Would you be interested in attending a one day hand's on security workshop?
  (not part of a quarterly meeting ... with a laptop preloaded with Oracle 12c)
- Show of hands please

- Thank you