One Time Passwords with the Oracle Database

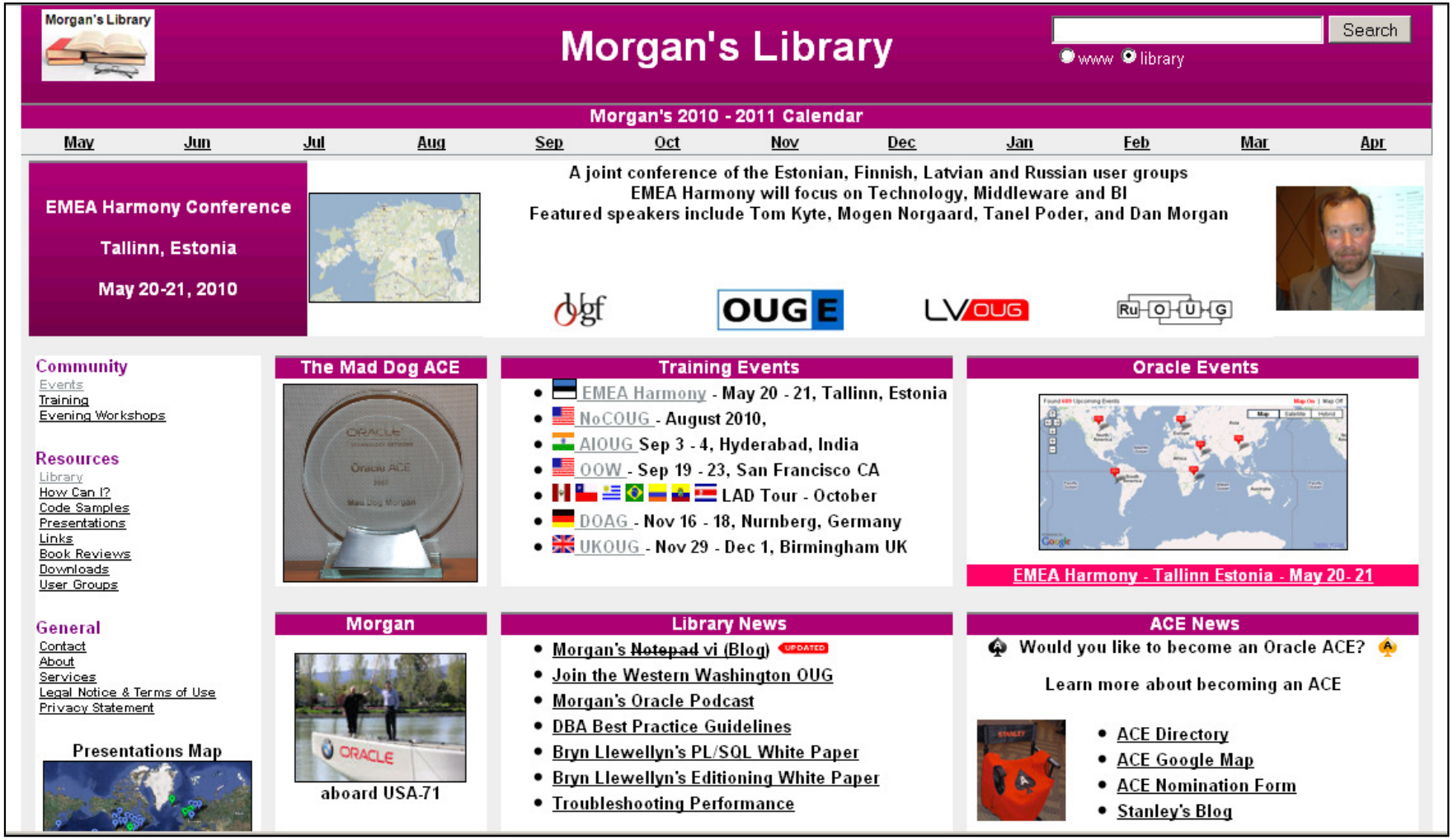
Daniel A. Morgan: Oracle ACE Director

Daniel Morgan

Daniel Morgan

- More than 45 years technology experience
  - First computer was an IBM 360/40 mainframe in 1970
  - Fortran IV and Punch Cards
- Oracle ACE Director
- Curriculum author and primary Oracle instructor at University of Washington
- Guest lecturer on Oracle at Harvard University
- Decades of hands-on SQL, PL/SQL, and DBA experience
- The "Morgan" behind Morgan's Library on the web
  www.morganslibrary.org
- 10g, 11g, and 12c Beta tester
- Co-founder International GoldenGate Oracle Users Group

The Morgan's Library Web Site



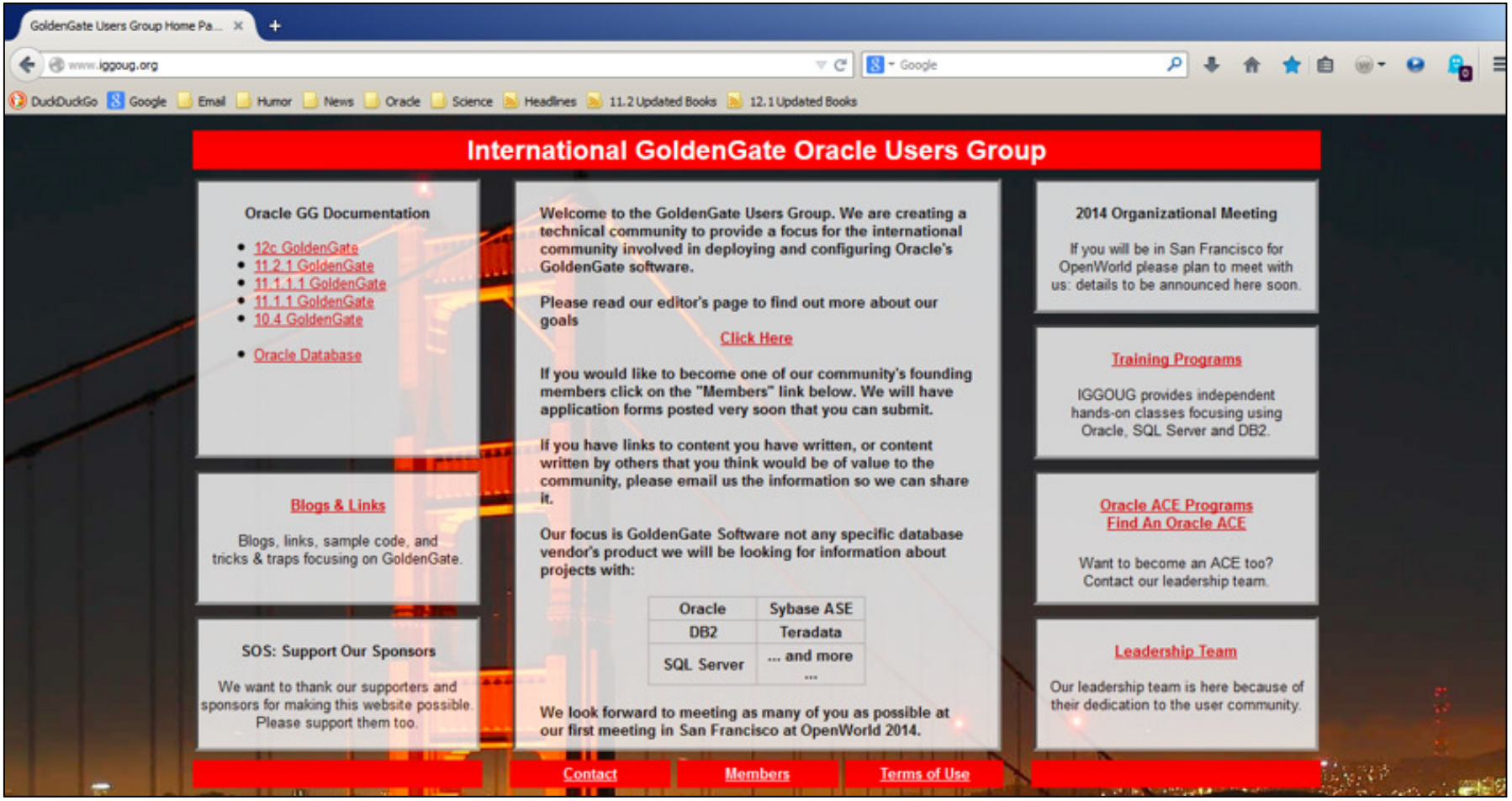
My Oracle Travels

cd $MORGAN_BASE/Seattle

cd $MORGAN_BASE/San_Francisco

Under the GoldenGate Bridge

IGGOUG: The International GoldenGate OUG



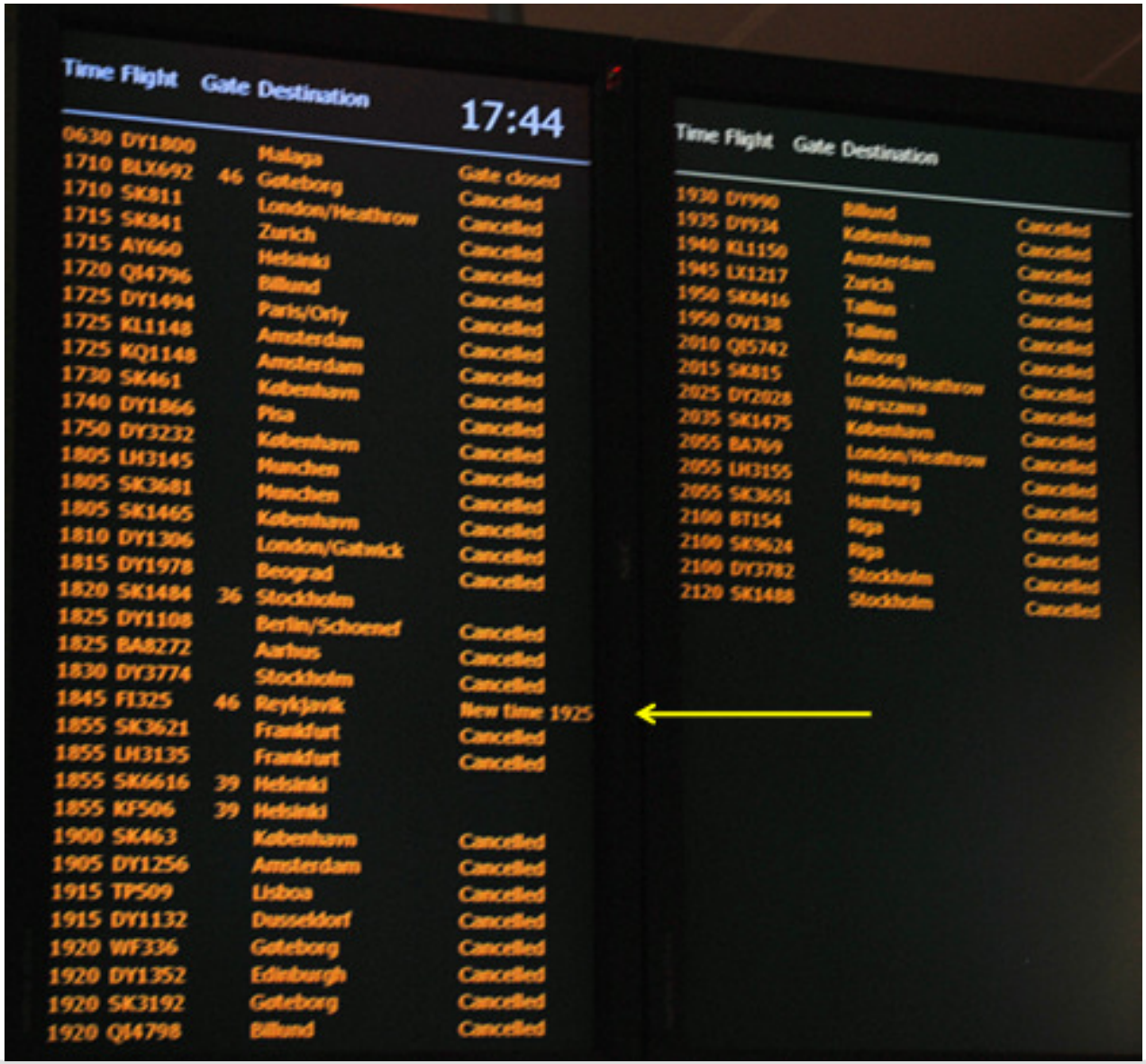
www.iggoug.org

San Francisco

To Watch Larry's AC72

Travel Log: Amsterdam & Cuzco

Travel Log: Thank You SAS



Travel Log: Global Warming Is A Myth

Travel Log: With Jonathan Lewis & Tim Gorman

Why Am I Doing This Presentation? (1:2)

- If you buy the Oracle Database just to have columns and rows ... Oracle will gladly sell you columns and rows
- If you buy the Oracle Database and pay extra for RAC, Active DataGuard, Advanced Compression, Multi-tenant, and In-Memory DB ... Oracle will gladly sell you that too
- But the Oracle Database licenses you already own include a staggeringly large range of tools that you have already paid for but are probably not using
- This presentation's sole purpose is to remind Oracle's customers that there is tremendous value in what they already own and that they owe it to themselves, and their employers, to maximize the value received
- This presentation will show how to secure a database with passwords no one can hack

Why Am I Doing This Presentation? (2:2)

- Is this a solution everyone should adopt?
  - No
- Is this a solution that some of you should adopt?
  - Perhaps
- This is an example of thinking outside of the box
- Of combining Oracle Database capabilities to create something that Oracle does not offer as a feature
- It is my sincere hope that you not only consider what I am presenting here today but also what you might create and share with the community

Do You Know About UTL_LMS?

```
DECLARE
 s VARCHAR2(200);
 i PLS_INTEGER;
 BEGIN
   i := utl_lms.get_message(601, 'rdbms', 'oci', 'English', s);
   dbms_output.put_line('English: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'Spanish', s);
   dbms_output.put_line('Spanish: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'German', s);
   dbms_output.put_line('German: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'French', s);
   dbms_output.put_line('French: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'Danish', s);
   dbms_output.put_line('Danish: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'Turkish', s);
   dbms_output.put_line('Turkish: OCI--00601 is: ' || s);

   i := utl_lms.get_message(601, 'rdbms', 'oci', 'Swedish', s);
   dbms_output.put_line('Swedish: OCI--00601 is: ' || s);
 END;
 /
```

The Other Reason For This Presentation (1:2)

RELATED NEWS

U.S. says busts largest-ever identity theft scheme

RPT-FEATURE-Chinese learn credit card perils the hard way

Chinese learn credit card perils the hard way

Visa, MasterCard seek growth abroad

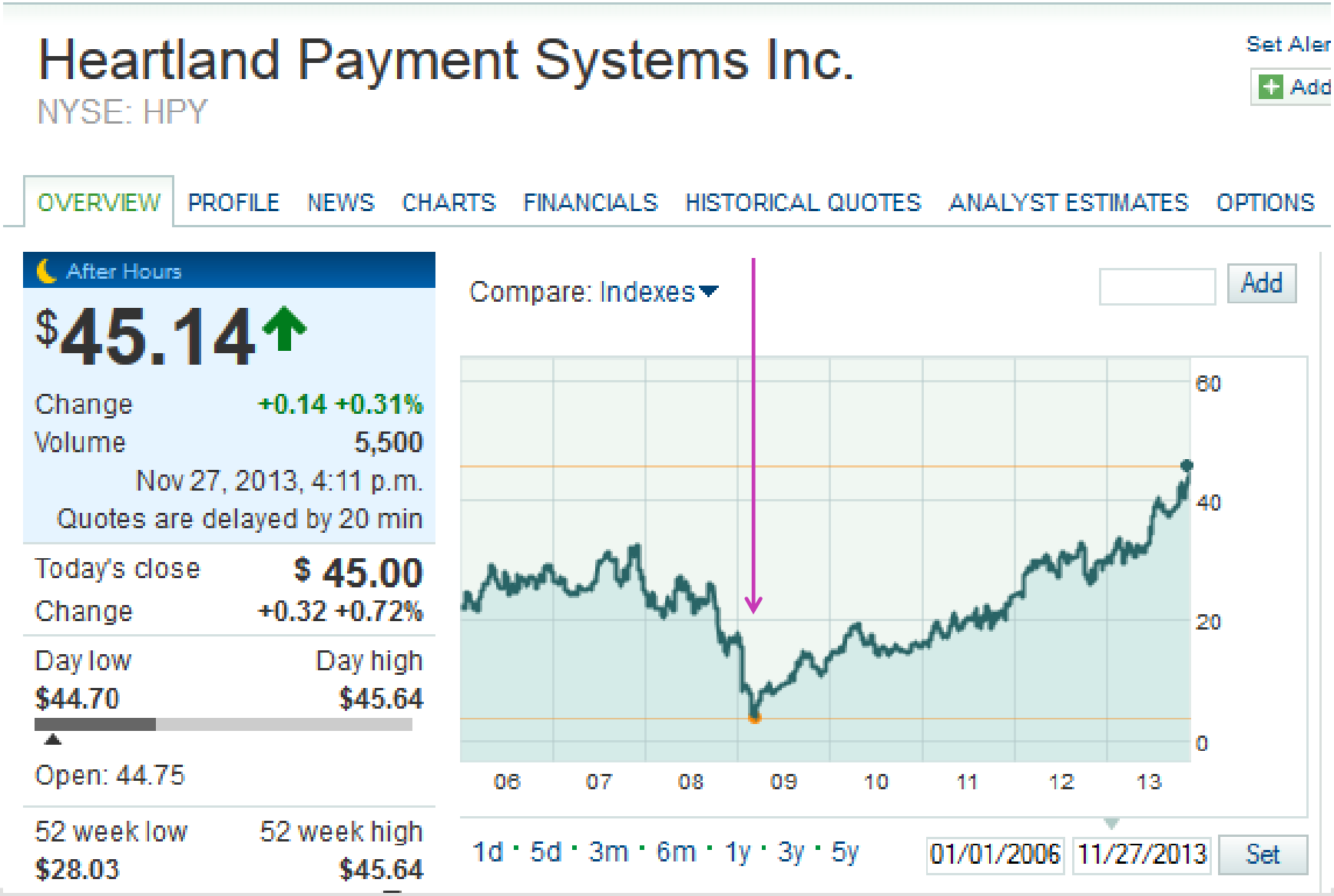
UPDATE 1-Heartland Payment posts Q2 net loss, lowers '09 outlook

(Reuters) - Three men were indicted on Monday for allegedly stealing more than 130 million credit and debit card numbers in what U.S. authorities said they believe is the largest hacking and identity theft case ever prosecuted.

Albert Gonzalez, a former government informant already in jail in connection with hacking cases, and two unnamed Russians were indicted on charges related to five corporate data breaches from 2006 to 2008.

Card numbers were stolen in those breaches from credit-card processor Heartland Payment Systems and retail chains 7-Eleven Inc and Hannaford Brothers Co, prosecutors said.

The Other Reason For This Presentation (2:2)



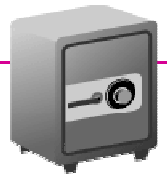
Disclaimer

- This room is an unsafe harbour
- No one from Oracle has previewed this presentation
- No one from Oracle knows what I'm going to say
- No one from Oracle has supplied any of my materials

- This presentation is about a capability of the Oracle database that, to the best of my knowledge, Oracle Corp. is not aware of

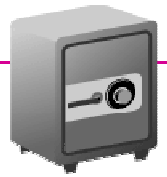
- But which I hope they learn about today and build as a formal feature into future versions to make our computing environment safer

What Is Database Security?



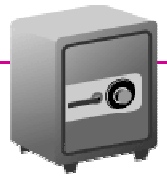
- Database security is the tools and techniques that can be utilized to make sure that they only people, processes, or environments that can access your data are those approved to do so
- You secure your data when everyone can access to what they need to access but can not access what they do not need to access
- Auditing is not security ... auditing tells you , after the fact, when a security breach has already occurred ... and then it is too late

What Data Do We Need To Secure?



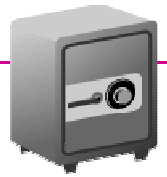
- Personal information
  - names
  - addresses
  - phone number
  - health-related
- Financial information
  - tax identification number
  - credit card numbers
  - bank account numbers
- Business and trade secrets
- Government secrets

What Objects Do We Need To Secure?



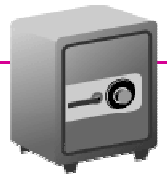
- Segment data
  - tables and materialized views
  - indexes
- Transaction data
  - v$sql
  - v$sqlarea
  - v$sql_bind...
- Redo logs
- Archived redo logs
- Flashback logs
- Operating system files

What Infrastructure Do We Need To Secure?



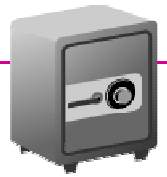
- Primary Databases
- Standby Databases
- Storage Snapshot
- Network communications
  - SQL*Net
  - Data Guard Replication network
- Storage Appliances
  - Disk
  - Tape
- Operating System Disks
  - Shell scripts
  - DataPump Files
  - SQL*Loader Files
  - External Tables

Database Security Technologies (1:3)



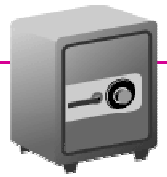
- Active Directory & LDAP
- Password Protection
  - enable case sensitive passwords
  - use the profile to expire passwords and prevent reuse
  - use a verify_password function to force password complexity
- Access Control Lists (DBMS_NETWORK_ACL_ADMIN)
- Backup Encryption
- Database Links (DBMS_DISTRIBUTED_TRUST_ADMIN)
- DBMS_CRYPTO
  - enhanced in 12cR1 with SHA-2
- DDL Event Triggers
- Fine Grained Access Control (DBMS_RLS)
  - also known as Virtual Private Database

Database Security Technologies (2:3)



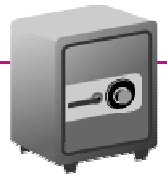
- Label Security
- Net Services (SQL*Net)
  - encryption
- Object Privilege Controls
- Parameterized Views
- Real Application Security
- Roles
  - password protected
  - package protected
- SecureFiles for LOBs
- SQL Injection Prevention
  - bind variables
  - DBMS_ASSERT built-in package

Database Security Technologies (3:3)



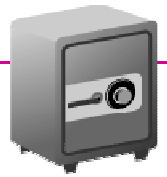
- System Privilege Controls
- Tablespace Encryption
- Transparent Data Encryption (TDE)
- Wallets

Security Related Packages



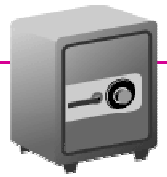
- DBMS_ASSERT (protection against SQL injection)
- DBMS_CRYPTO
- DBMS_DISTRIBUTED_TRUST_ADMIN (secure db_links)
- DBMS_NETWORK_ACL_ADMIN (secure network access)
- DBMS_RANDOM
- DBMS_RLS (virtual private database)

And We Know The Oracle Database Has Weaknesses



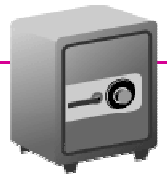
- Execute on UTL_FILE is granted to public which means anyone that has DB access can write data to a file
- And anyone with DB access can read O/S files
- DBMS_ADVISOR can be used to write data to a file
- DBMS_XSLPROCESSOR can be used to write data to a file
- DBMS_LOB can be used to read O/S files
- External tables can be used to read O/S files
- Cron jobs often contain clear-text passwords
- DataPump scripts often contain clear-text passwords
- Shell scripts often contain clear-text passwords
- DBAs that do not use the available tools to lock down the system for fear out of lack of knowledge or fear of complaints

Let's Focus On One Of Those Weaknesses: Passwords



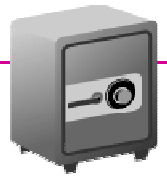
- Most passwords are not especially complex
- Most passwords are not especially unpredictable
- Most passwords are not expired regularly
- Most passwords that are expired can be reused
- Most applications use a single password across all application servers and the password is known to many people

Consider The Impact On Security If ...



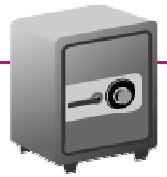
- No human knew the password to your database?
- No server knew the password to your database?
- No process knew the password to your database?
- All passwords were long, complex, totally random, and unpredictable?
- All passwords were invalid as soon as they were used?
- All passwords became invalid within a fraction of a second even if they were never used?

What Is A One-Time Password?



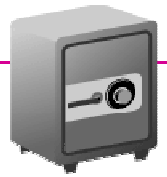
- A one-time password is a password valid for only one use: A single login session or transaction
- As soon as a one-time password is used it becomes invalid
- Eliminates the most common weaknesses of traditional passwords
  - brute force attack
  - replay attack
- No persistent record is kept so if the passwords are not truly random it is impossible to utilize previous passwords to guess at future passwords

How Do Oracle Passwords Work?



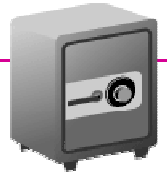
- When you attempt to log onto an Oracle database the password is used for authentication
- If a password is changed after logon existing connected sessions are not affected
- An attempt to logon subsequent to an existing logon requires re-authentication

The Challenge (1:2)



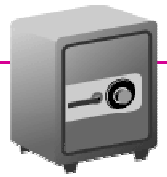
- The biggest single problem with passwords: Database passwords, operating system passwords, PIN Number, etc. is that some human chooses it, knows it, writes it down, and may pass it along to others
- Take the human out of the equation and you get a lot closer to secure access
- On many assignments I see DOT NET and Java developers hard-coding passwords into Web Application servers: Security by appearance only
- So here's a simple implementation of a solution to the problem. My actual production implementation is far more complex and if not unbreakable, nothing is, a lot closer to the goal than where 99% of database implementations are today

The Challenge (2:2)



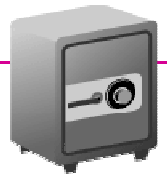
- The heart of the solution is that any application, or user, wishing to connect to the database must first obtain a valid password by calling an "open" schema that provides access to only one function
- The password received must then be used to log into the application schema, very quickly, because the password is again changed to an unknown random password

One Time Password Design



- Step 1: User requests a one-time password
- Step 2: User authentication
  - If authenticated
    - *a complex one-time password is generated*
    - *the current schema password is changed*
    - *a scheduler job, with a very short delay, is created to again change the password*
    - *the valid password is returned to the user*
  - If not authenticated
    - *the application schema is locked (optional)*
    - *an audit record is created*
    - *DBA team is notified*
    - *security team is notified*
- Step 3: User logs onto the application schema
- Step 4: An AFTER_LOGON trigger changes the password

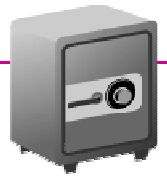
Step 1: One-Time Password Requested



- User connects to a non-privileged schema and requests a password

```
SELECT gen_random(schema_name)
FROM dual;
```

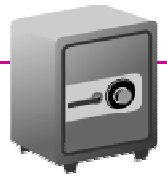
Step 2a: Validate The Request For The Password



- User requesting the password is validated

```
CREATE OR REPLACE FUNCTION gen_random(access_to_schema IN VARCHAR2)
AUTHID CURRENT_USER
RETURN VARCHAR2 IS
 uname  VARCHAR2(30) := sys_context('USERENV', 'authenticated_identify');
 ameth  VARCHAR2(30) := sys_context('USERENV', 'authentication_method');
 cpname VARCHAR2(30) := sys_context('USERENV', 'client_program_name');
 cuname VARCHAR2(30) := sys_context('USERENV', 'current_user_name');
 dblink VARCHAR2(30) := sys_context('USERENV', 'dblink_info');
 eident VARCHAR2(30) := sys_context('USERENV', 'enterprise_identity');
 itype  VARCHAR2(30) := sys_context('USERENV', 'identification_type');
 ipaddr VARCHAR2(30) := sys_context('USERENV', 'ip_address');
 isdba  VARCHAR2(30) := sys_context('USERENV', 'isdba');
 netpro VARCHAR2(30) := sys_context('USERENV', 'network_protocol');
 osuer  VARCHAR2(30) := sys_context('USERENV', 'osuser');
 proxyu VARCHAR2(30) := sys_context('USERENV', 'proxy_user');
 srvc   VARCHAR2(30) := sys_context('USERENV', 'service');
 term   VARCHAR2(30) := sys_context('USERENV', 'terminal');

 PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  -- verify values here for the schema access being requested
  -- validating criteria can include day of the week and hours of operation
END;
/
```

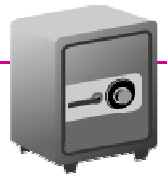
Step 2b: Authentication Succeeds



- A complex one-time password is generated and returned

```
CREATE OR REPLACE FUNCTION gen_random(access_to_schema IN VARCHAR2)
AUTHID CURRENT_USER
RETURN VARCHAR2 IS
 rStr VARCHAR2(50) := dbms_crypto.randombytes(25);
 sVal POSITIVE := dbms_random.value...
 PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  execute immediate 'ALTER USER ' || acess_to_schema ||
                    ' IDENTIFIED BY "' || SUBSTRB(rStr, sVal, 30) || '"';

  dbms_scheduler.create_job(
    job_name=>'A' || TO_CHAR(btLen),
    start_date=>SYSDATE+10/86400, -- this is 10 sec. for demo purposes only
    enabled=>TRUE,
    auto_drop=>TRUE,
    job_type=>'PLSQL_BLOCK',
    job_action=>'DECLARE x VARCHAR2(30); BEGIN x :=
                 gen_random(access_to_schema); END; ');

  RETURN SUBSTRB(rStr, sVal, 30);
END;
/
```

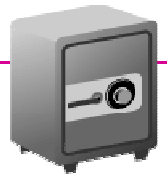
Step 2c: Authentication Failure



- An invalid access is identified
- Return something that looks valid but is not

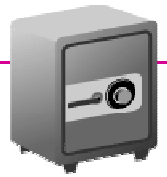
```
CREATE OR REPLACE FUNCTION gen_random(access_to_schema IN VARCHAR2)
AUTHID CURRENT_USER
RETURN VARCHAR2 IS
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  -- insert into audit_table date-time and validation info.
  COMMIT;

 c := utl_smtp.open_connection('smtp-server.morganslibrary.org');
  utl_smtp.helo(c, 'morganslibrary.org');
  utl_smtp.mail(c, 'mailsys@morganslibrary.com');
  utl_smtp.rcpt(c, 'recipient@oracle.com');
  utl_smtp.open_data(c);
  send_header('From', '"Sender" <database_name@morganslibrary.org>');
  send_header('To', '"Recipient" <security_team@morganslibrary.org>');
  send_header('Subject', 'Database Security Threat');
  utl_smtp.write_data(c, UTL_TCP.CRLF || 'An invalid attempt was made to ...');
  utl_smtp.close_data(c);
  utl_smtp.quit(c);

  RETURN dbms_random.string('X', 30);
END;
/
```

Step 3: Application Performs A Valid Logon



- Application code takes the string returned by the call to get_random and logs into the application schema in a fraction of a second so that the password does not become invalid before it is used

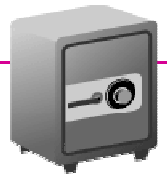
Step 4: After Login Automatic Password Reset



- AFTER LOGON Trigger Resets The Password

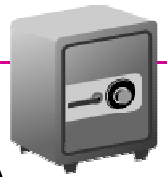
```
CREATE OR REPLACE TRIGGER logintrig
AFTER LOGON ON DATABASE
CALL gen_random('INTERNAL_RESET') -- pseudocode - trigger must make function call
/
```

Implementation Risk Factors



- A flood of connection attempts caused by an application server restart would overwhelm the system unless spaced out such that, for example, there is one connection attempt every 0.25 second (or something similar)
- A DDoS attack could be initiated by flooding the system with bogus password requests

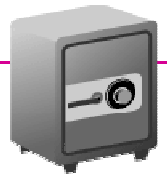
Risk Mitigation



- Rate limit connections in LISTENER.ORA or TNSNAMES.ORA
- CONNECTION_RATE_<listener_name>
  - Specify a global rate that is enforced across all listening endpoints that are rate-limited. When this parameter is specified, it overrides any endpoint-level numeric rate values that might be specified in connections per second

```
LISTENER= (ADDRESS_LIST=
   (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE_LIMIT=yes))
   )
CONNECTION_RATE_LISTENER=4
```

Conclusion



- The user community can create new functionality that extends the value of the Oracle Database
- There is a need for moving beyond the simple, but insecure, userid/password connection methodology
- The risks to our organizations and our personal data are demonstrably so high that we need to do better and need to encourage Oracle Corp. to provide enhanced tools

ERROR at line 1:
ORA-00028: your session has been killed

Thank you

Feel free to ask questions now or contact me at PTC
daniel.morgan@perftuning.com / +1 206-669-2949
Skype: damorgan11g