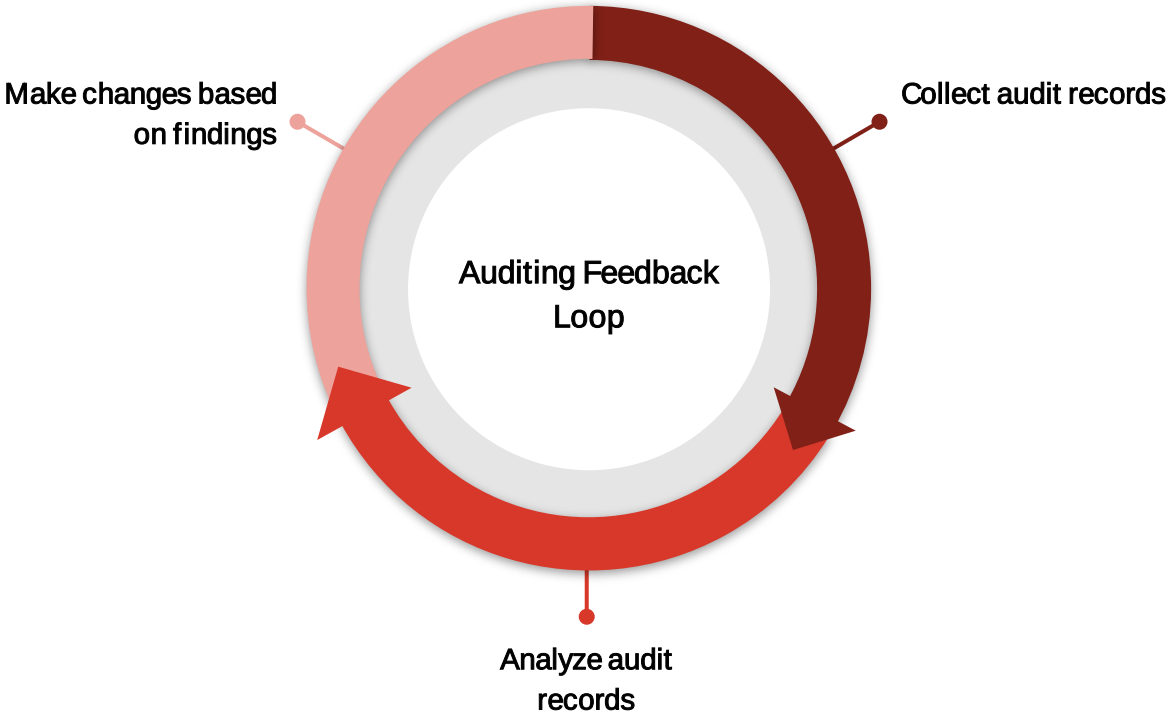
Auditing Voter Registration Databases

Background

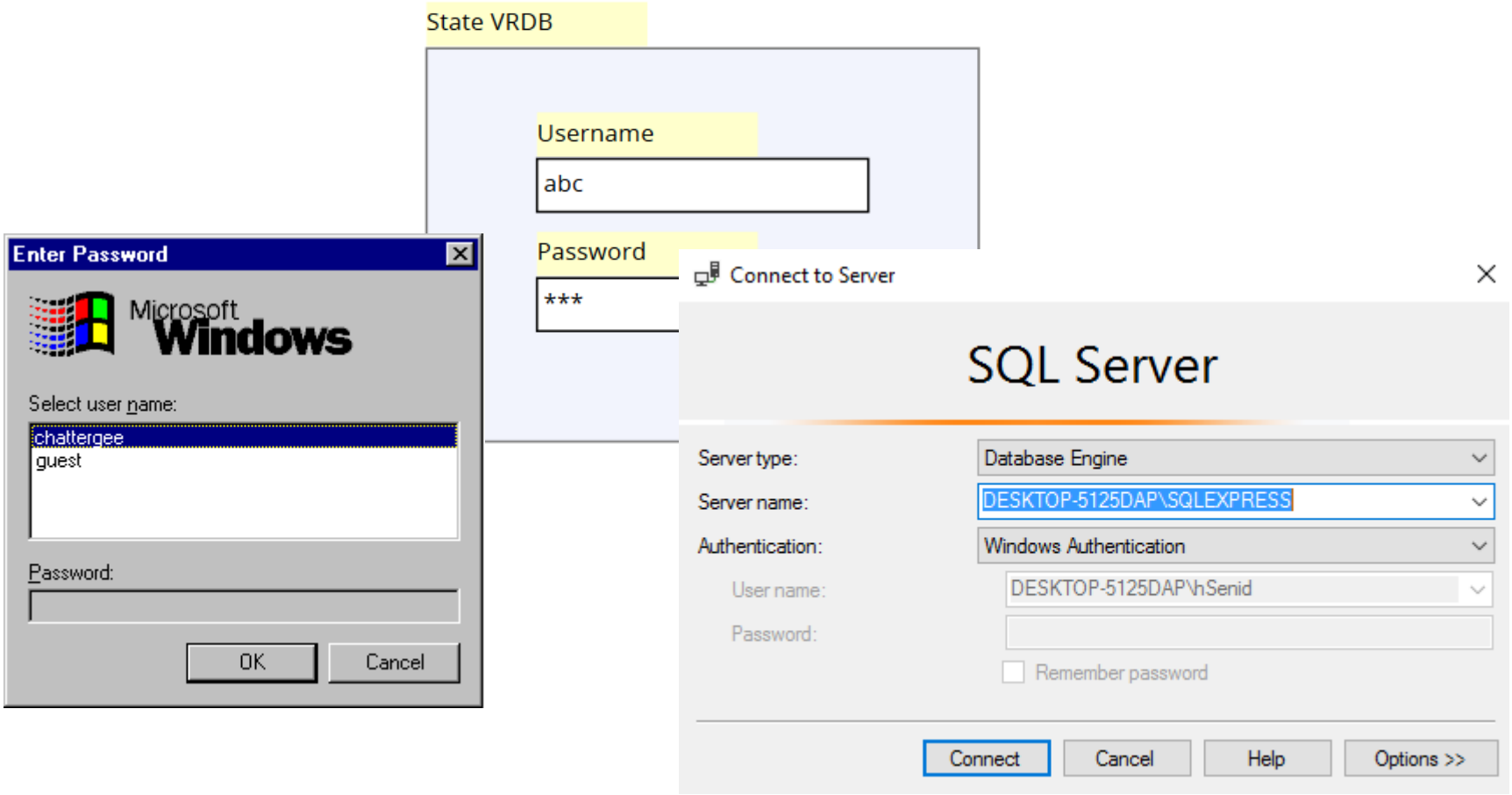
- 10+ Years Election Administration
- Lead Developer, Statewide Voter Registration DB (2012-2017)
- Oracle Certified Professional...
- ...But talk will be DB vendor neutral

Why audit?

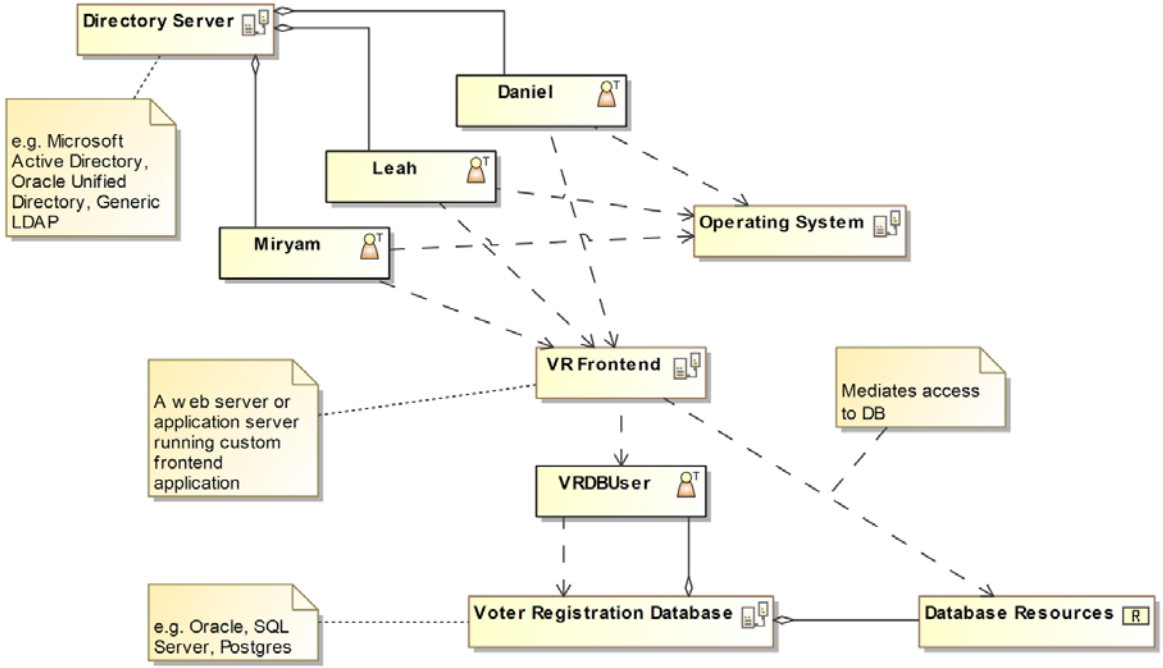
- Enable accountability for actions
- Deter users from inappropriate activity
- Support investigation of suspicious activity
- Monitor general activity in the database
- Additional layer of detection when prevention isn’t possible
  - Security Holes
  - Zero-day attacks


Make changes based on findings
Collect audit records
Auditing Feedback Loop
Analyze audit records

Part I: Auditing Database Access


State VRDB
Username
abc
Password
***
Enter Password
Microsoft Windows
Select user name:
chattergee
guest
Password:
OK
Cancel
Connect to Server
SQL Server
Server type:
Database Engine
Server name:
DESKTOP-5125DAP\SQLEXPRESS
Authentication:
Windows Authentication
User name:
DESKTOP-5125DAP\hSenid
Password:
Remember password
Connect
Cancel
Help
Options >>

Example Authentication Configuration



Auditing accounts for least privilege

- Users should have only the rights they need in the applications they use.
- Applications should have only the rights they need to the database!
  - Applications ARE users! (non-person entities)

Practical Account Auditing

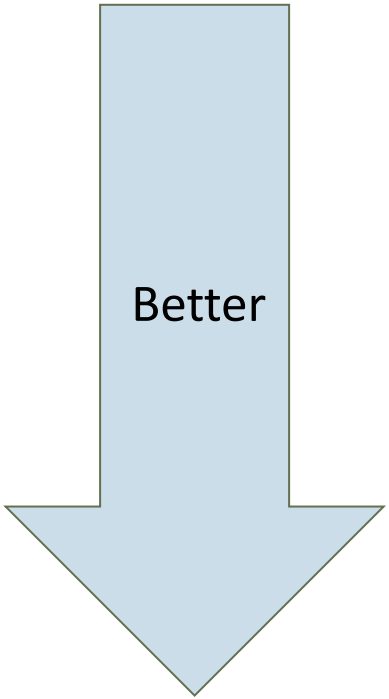
- Get list of users
- Map users to people, applications
  - Do they still work here?
  - Are they in the same position?
- Deactivate, delete, adjust accounts as necessary
- Grant rights to roles/groups rather than users
  - Users have roles, which inherit the rights of those roles
  - Database roles can map to actual business roles (Registration Clerk, Campaign Finance, etc.)
- But what about application “users”?

Audit Applications using VRDB

ORA-31050: Access denied

ORA-31050: Access denied

- How auditable application is will depend on how it was written. DBA + Devs working together!
- Spaghetti Code
  - Database calls scattered across codebase
- Database Access Layer
  - Database calls in a separate code module
- Application can make query
  - Application sends query to DB
  - Less Securable
- Application can make procedure call
  - Application sends request to procedure in DB
  - More securable
  - **Much** more auditable
  - https://bit.ly/2CcyAks



Part II: Auditing Database Activity

Building a Database Audit Trail

- Know what is being included!
  - Logins
  - Data Access
  - Data Manipulation
- And at what level of granularity!
  - Database Level (**bob** logged in)
  - Table Level (**bob** looked at the **voter** list)
  - Query Level (**bob** looked at the voter record for **John Smith**)
- Look at the audit trail. What questions can they answer?
- How long do you need to keep them? (not forever!)

Building a Database Audit Trail

- Discover and locate the sensitive data
- Assess the security risks around the data
  - Is it personally identifiable information?
  - Is it vital to the system (e.g. a voter's eligibility status)
- Do the logs contain everything you want to know?

Don't store PII or other sensitive information in your log files!

Performing an Audit

- Need a methodology
  - Continuous Monitoring
    - Look for out of normal activity
  - Orange County Voter Registration Audit Report, Version 3 (Alvarez, et al.)
    - interquartile range (IQR) method
    - Identify *events* that warrant further examination
    - Used daily “snapshots” of VRDB
    - Paper at https://bit.ly/2UA3gDy

Thank You!

John Dziurłaj
Systems Architect
The Turnout LLC
john@turnout.rocks
Work/Fax 234-706-6434