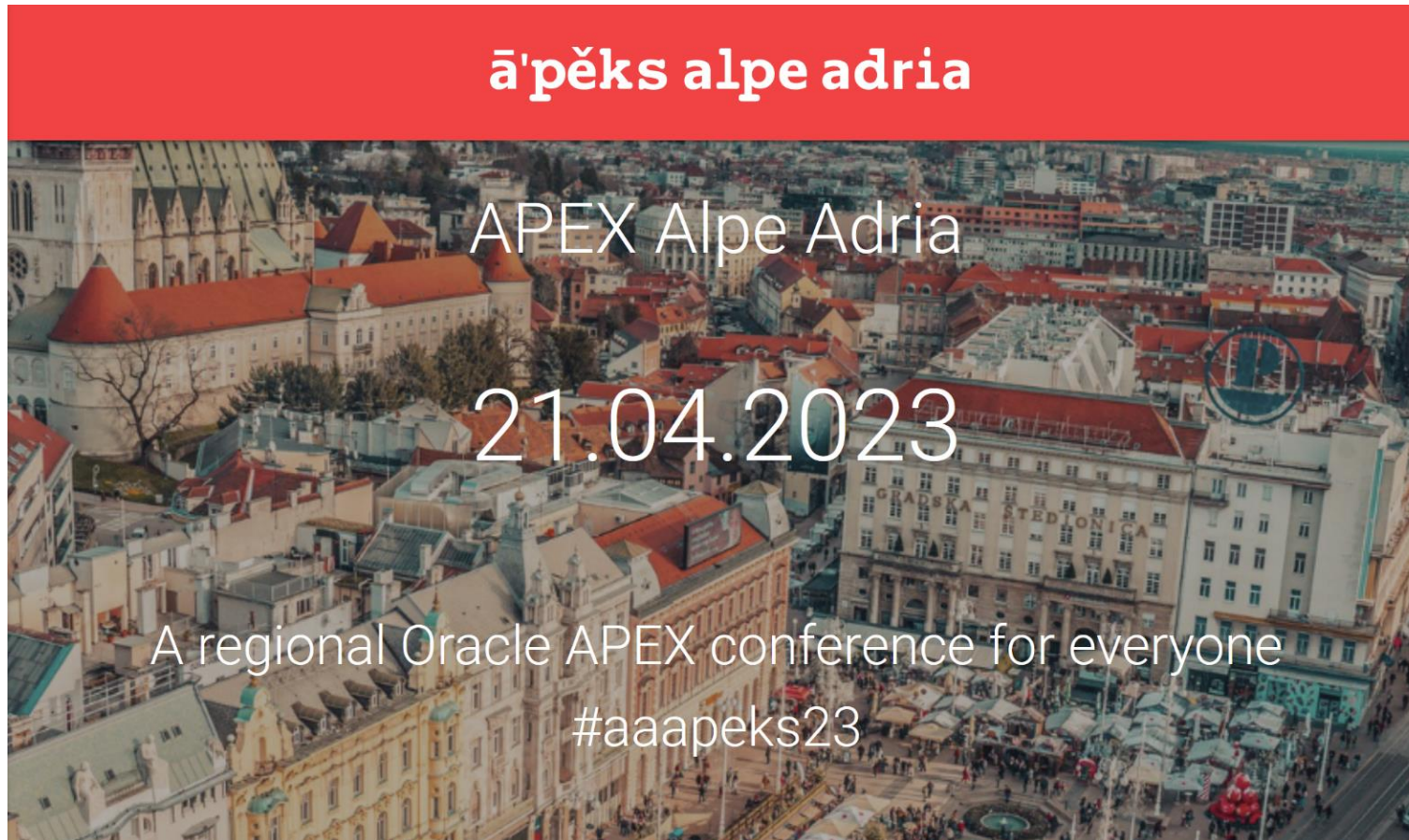


APEX security scanning – what have we learned



Lino Schildenfeld

@LinoSchilde

lschilde.blogspot.com

Oracle Database 23c Free Version Now Available to Developers

Oracle Database 23^c Free Developer Release

Easy, early access to application development features.



23^c



AUSOUG APEX News



<https://www.ausoug.org.au/news/apex-special-series-2023/>



Mentor and Speaker Hub

- Our goal is to *connect* speakers with mentors to assist in *preparing* technical sessions and *improving* presentation skills

Interested? Read more and get in touch

<https://mashprogram.wordpress.com>

My story



AUSOUG



FOEX



APEX RD
research & development

- AUSOUG APEX webinars
- NZ APEX meetup organizer

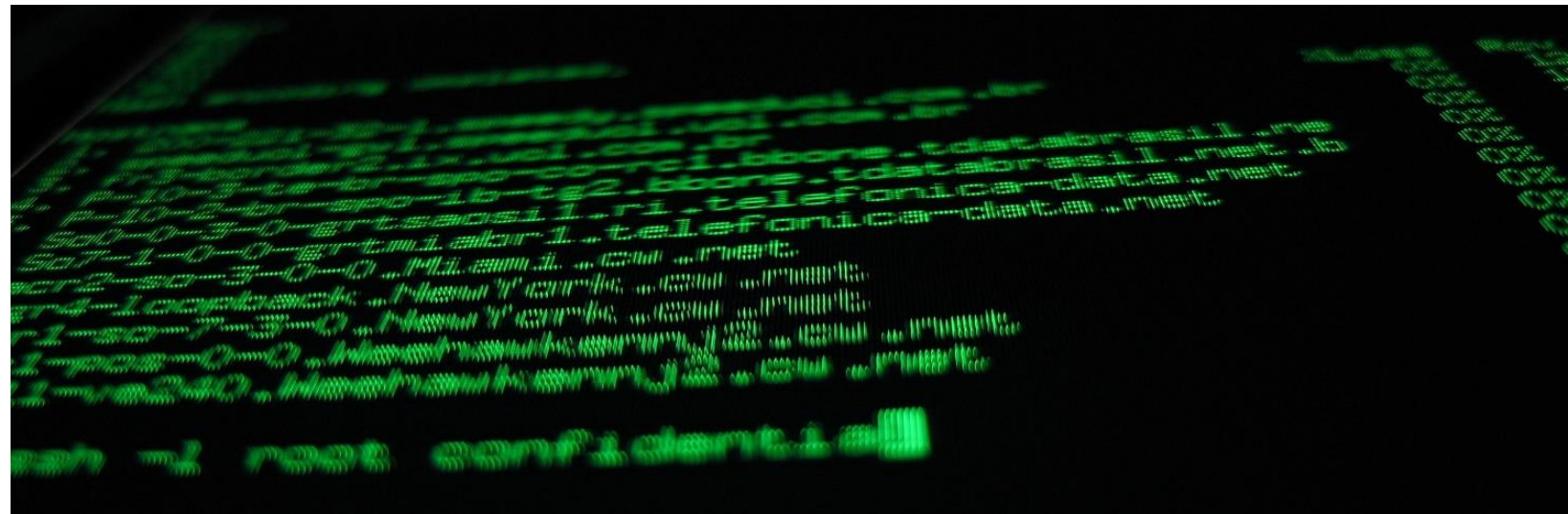
- APEX World Member of the Month
- Conference speaker
- Blogger

@LinoSchild



Intro

- There is a hacker attack each 39 seconds
- On average 180 days to detect a breach
- 95% of breaches are due to human error
- 53% of successful are not even detected



Technology

Google Chrome's latest update has a security fix you should install ASAP

Luckily updating Chrome is usually as simple as restarting

By [Justine Calma](#) | [@justcalma](#) | Sep 5, 2022, 9:51am EDT

[f](#) [t](#) [s](#) SHARE

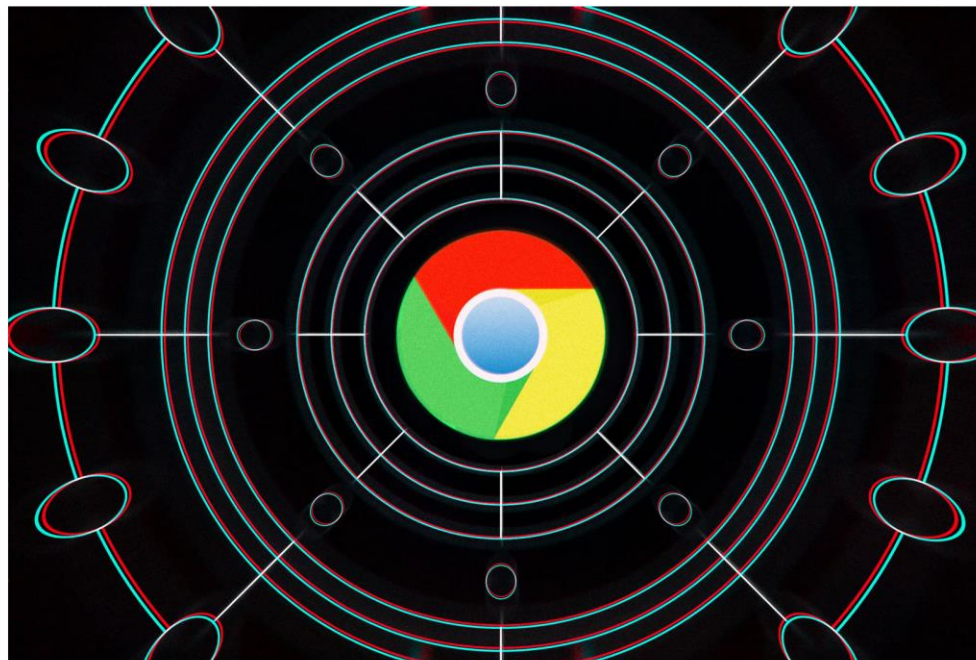


Illustration by Alex Castro / The Verge

Free Browser Extension

Coupert: The tool searches for codes and applies them automatically at the checkout.

Coupert

Install


**verge
deals**

Subscribe to get the best Verge-approved tech deals of the week.

Email (required)



Optus <noreply@e.optus.com.au>

to me ▼

OPTUS

Urgent update about your personal information

Dear Lino,

Tue, Sep 27, 4:53 PM (6 months ago)

It is with great disappointment I'm writing to let you know that Optus has been a victim of a cyberattack. As a former Optus customer this has resulted in the disclosure of some of your personal information.

No financial information or passwords have been accessed. The information which has been exposed is a combination of your name, date of birth, email, phone number and/or address associated with your former account. **No ID document numbers or details**

Intro

- It works best if it is applied in layers
- Poor or no security practices/standards
- Staying ahead of the game is hard
- We developers are first line of defence



ā'pěks
(#orclapex)

is
SAFE

APEX features

Out of the box features

- Features we have available:
 - APEX URL
 - Authentication/Authorisation
 - Security settings
 - Session handling
 - Session state protection
 - Encoding APIs
 - Advisor tools
 - External tools

APEX provides mechanisms to secure your apps!



Typical examples

- URL tempering
- Access control inconsistency
- Hidden or Display only items
- JavaScript submit page
- Unprotected application processes or items

XSS examples

- Reports Column Display Type
- Disabling Escape special characters
- Report Column Formatting - HTML Expressions
- Report Column Formatting - Column Link
- Report Column - List of Values
- HTP.p

SQL injection examples

- Input changes the expected results
- Function returning Query
- Most things with Substitution variables
- Htp.p
- DYNAMIC SQL
 - Cursors
 - EXECUTE IMMEDIATE
 - APEX API

We need Security tools

APEX Advisor

The screenshot displays the APEX Advisor interface within the Oracle APEX Utilities section. The left sidebar contains a list of utility tools: Application, Upgrade APEX, Attribute, Debug, Export Report, Embedded, and Font APEX. The main content area shows the 'Advisor' tool, which is highlighted with a red box. The 'Advisor' tool description states: 'Perform various checks on this application, including programming errors and best practices.' Below this, there are links for 'Recently Updated Pages' and 'Change History'. The 'Check Page(s)' dialog box is open, showing a list of checks to perform. The dialog box includes a 'Perform Check' button and a list of checks categorized by type: Error (22), Security (47), Warning (5), Performance (2), and Quality Assurance (13). The checks are listed with their respective counts and descriptions. The 'Checks to Perform' section is expanded, showing a list of checks with green checkmarks indicating they are selected. The checks are grouped by category: Errors, Performance, Usability, Quality Assurance, Security, and Accessibility.

Application 109 \ Utilities

Utilities

Advisor - Google Chrome

intapex04.ama.at:8443/ords/entwentman1/f?p=4000:8101:6939559996221:::

Check Page(s) Perform Check

Advisor

Perform various checks on this application, including programming errors and best practices.

Recently Updated Pages

View a history of the page updates made on this application.

Change History

Check Page(s) Perform Check

The Advisor performs various checks on your application, including programming errors, security issues, quality assurance, and other best practices. For the list of checks, see [Checks to Perform](#). Once executed, your previous settings will be recalled for the next use.

Specify a comma separated list of pages to check, or leave blank to check all pages.

Page(s) ?

Pages last edited within the last 7 days

Checks to Perform

Select All Deselect All

Errors: ?

- References with Substitution Syntax
- References with Column Syntax
- References with Bind Variable Syntax
- Declarative References of Application Items, Page Items, Columns or Interactive Report Filters
- Referenced Page Number Exists
- Is Valid SQL or PL/SQL Code
- Fetch, DML, MR* Processes are Valid
- Unconditional Branch before other Branches
- Referenced Button in When Button Pressed exists
- Button is not compatible with Dynamic Actions

Performance: ?

- V Function used in SQL Statements
- User Interface includes compatibility JavaScript

Usability: ?

- Target Page Authorization is also set for Current Component
- Associated Item or Column of Validations

Quality Assurance: ?

- Hardcoded Application ID
- Report has Default Order
- Page Item has Help Text
- Deprecated attribute values

Security: ?

- Inappropriate use of Substitution Syntax
- Application attributes that can be locked down
- Authorization
- Session State Protection
- Browser Security Settings

Accessibility: ?

- Theme Style tested for accessibility
- Page has page title
- Region has Row Header
- Page Item has label

Warnings: ?

Security tools - APEX Advisor

- Takes time getting used to
- It is not a professional tool
- **App Builder -> Application X -> Utilities**
- It checks for errors, security issues, usability and quality assurance

APEXSec

The screenshot displays the APEXSec desktop application interface. The main window, titled 'Welcome', features the heading 'Getting started with APEXSEC' and instructions: 'To begin securing your Oracle APEX application, create a new project and specify how APEXSEC can access your application.' Below this, there is a link 'Click here to create a new project' and a section titled 'Alternatively you can create a project by:' with four radio button options: 'Retrieve Application from the APEX Application Builder URL' (selected), 'Retrieve Application from Database Schema(s)', 'Import Application from an APEX export SQL File or ZIP File', and 'Import Application and Additional SQL Sources from a Directory'. A 'New Project...' dialog box is open in the foreground, mirroring these options. It includes input fields for 'Apex Web URL' (https://apex.oracle.com/pls/apex/), 'Workspace' (lino), 'User Name' (lschilde@gmail.com), and 'Password'. There is also a 'Messages' text area and a checkbox for 'Dynamically Scan Packages and Calls'. The background shows a sidebar with 'ApexSec Desktop' and 'File Report Tools Help' menus, and a file explorer with 'VEMACS Electronic Storage' and 'TT_BASELINE' folders. The bottom of the window shows the license 'Licensed to: Lino Schilde, Skillbuilders' and the browser version 'Internet Explorer 11.0'. The APEXSEC desktop logo and version 'v3.1.24.142' are in the bottom right corner.

ApexSec Desktop

File Report Tools Help

Getting started with **APEXSEC**

To begin securing your Oracle APEX application, create a new project and specify how **APEXSEC** can access your application.

[Click here to create a new project](#)

Alternatively you can create a project by:

- ☒ Retrieve Application from the APEX Application Builder URL
- ☐ Retrieve Application from Database Schema(s)
- ☐ Import Application from an APEX export SQL File or ZIP File
- ☐ Import Application and Additional SQL Sources from a Directory

Apex Web URL: https://apex.oracle.com/pls/apex/

Workspace: lino

User Name: lschilde@gmail.com

Password:

Messages:

☐ Dynamically Scan Packages and Calls

Cancel Ok

Drag and Drop Project File

Licensed to: Lino Schilde, Skillbuilders
Browser Version: Internet Explorer 11.0

APEXSEC desktop v3.1.24.142

Security tools - APEXSec

- Licensing cost
- Ability to scan from live apps vs uploaded files
- Tests your PLSQL code too
- Available as desktop version
- Fairly simple to run
- With detailed instruction how to sanitize security flaws found

APEX SERT – free tool



XSS: Unescaped Output - Items

SEV 1 Submit All

Dashboard Settings PageRpt SQLi XSS URL Reports

View As: Approved As

96.8% Approved 96.8% Pending 96.8% Raw

1,301 of 1,344 Points

Entire Score

Page Score

Attribute Set: AMA Security Vorgab

Defects: 4

Approximate Time to Fix: 47 hours

Approved: 3 out of 31 possible points (9.7%)

Pending: 3 out of 31 possible points (9.7%)

Raw: 3 out of 31 possible points (9.7%)

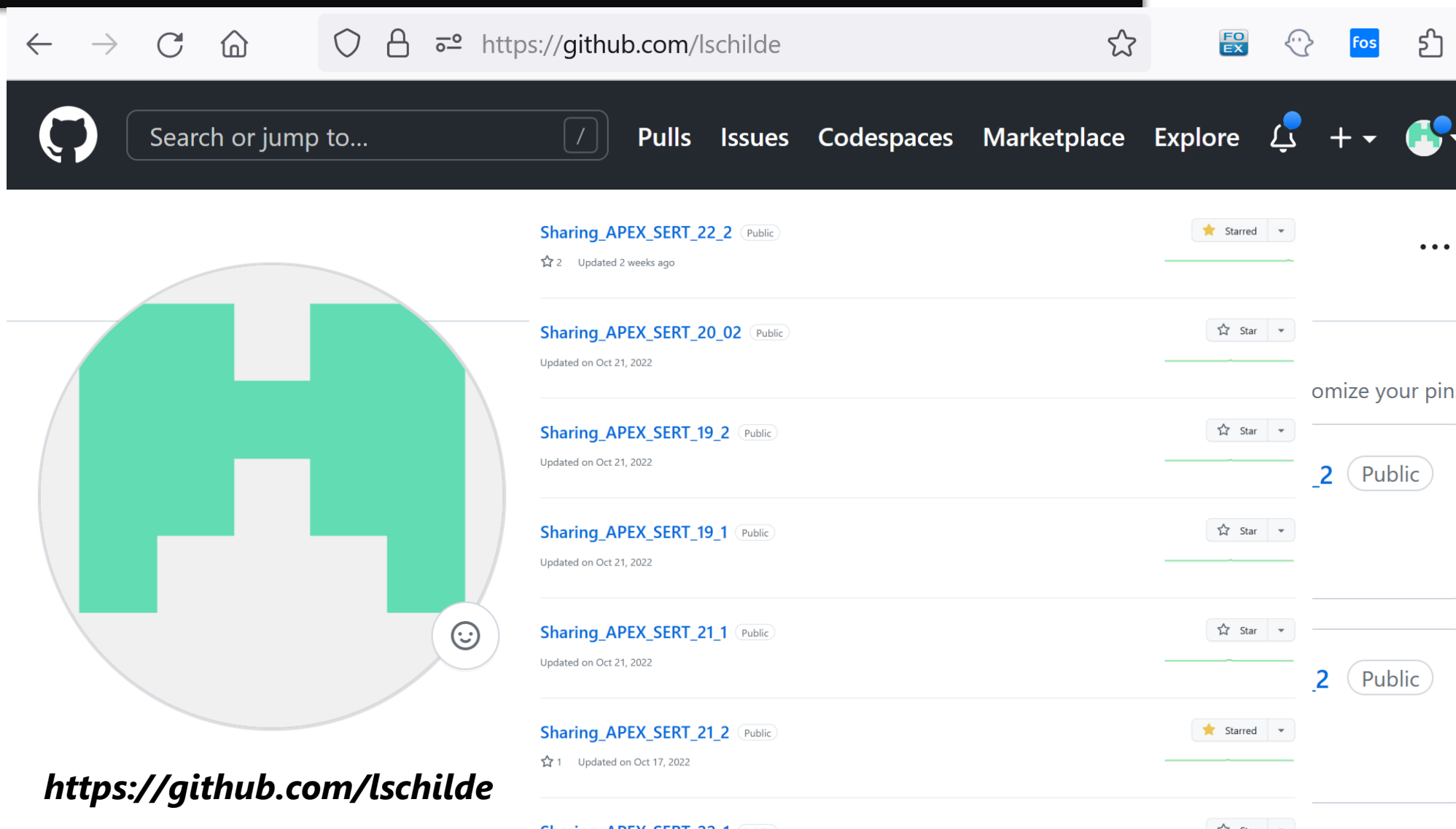
	Page Name	Page ID	Item Name	Escape	Updated By	Updated On	Result		
	Partitionen und Vorhaefzeilen	5	PS_MESSAGE	-	-	-	FAIL		
	Partitionen und Vorhaefzeilen	5	PS_MESSAGE_PROD_INSTANZ	-	-	-	FAIL		
	Session Control	14	P14_SID	-	-	-	FAIL		
	Session Control	14	P14_SERVER	-	-	-	FAIL		

<https://github.com/OraOpenSource/apex-sert>

Security tools - APEX SERT

- No licencing cost
- Ability to scan only 'live'
- Will not tests your PLSQL code
- Community edition available? Original was outdated
- OpenSource tool needed on the market

Security tools - APEX SERT



The screenshot shows a web browser displaying the GitHub repository page for 'lschilde'. The browser's address bar shows the URL 'https://github.com/lschilde'. The GitHub navigation bar includes the search bar, 'Pulls', 'Issues', 'Codespaces', 'Marketplace', 'Explore', and a user profile icon. The repository page features a large green 'A' logo on the left. The main content area lists several repositories, each with a title, a 'Public' badge, and a star count. The repositories are: 'Sharing_APEX_SERT_22_2' (2 stars, updated 2 weeks ago), 'Sharing_APEX_SERT_20_02' (updated Oct 21, 2022), 'Sharing_APEX_SERT_19_2' (updated Oct 21, 2022), 'Sharing_APEX_SERT_19_1' (updated Oct 21, 2022), 'Sharing_APEX_SERT_21_1' (updated Oct 21, 2022), 'Sharing_APEX_SERT_21_2' (1 star, updated Oct 17, 2022), and 'Sharing_APEX_SERT_22_1' (partially visible). Each repository entry has a 'Star' button and a 'Public' badge. On the right side, there is a section titled 'omize your pins' with a '2' and a 'Public' badge.

← → ↻ 🏠 🔒 🔑 🔗 https://github.com/lschilde ☆

FO EX 👤 Fos 🗑️

🐱 Search or jump to... / Pulls Issues Codespaces Marketplace Explore 🔔 + ▾ 🌐 ▾

Sharing_APEX_SERT_22_2 Public ☆ 2 Updated 2 weeks ago

Sharing_APEX_SERT_20_02 Public Updated on Oct 21, 2022

Sharing_APEX_SERT_19_2 Public Updated on Oct 21, 2022

Sharing_APEX_SERT_19_1 Public Updated on Oct 21, 2022

Sharing_APEX_SERT_21_1 Public Updated on Oct 21, 2022

Sharing_APEX_SERT_21_2 Public ☆ 1 Updated on Oct 17, 2022

Sharing_APEX_SERT_22_1 Public

omize your pins 2 Public

<https://github.com/lschilde>

Demo - Security tools

What does it look for?

Security scans

- Various settings
- Page and Item SSP
- Potential XSS and SQL Injections
- Button and page processes
- Authentications and authorizations

It is all about inconsistency

TIP #1

Keep your data clean

User inputs checks

- Sanitize and validate it
- Do not trust it
- >50% of security risk
- Normally never done and ignored

TIP #2
Know your items

Items tips

- Not all items are the same
- Use SSP appropriately
- Store encrypted in session state
- Maintain Session State
- Use
 - Value protected vs escape special character or (Format Settings)
- Restricted characters

Name	P5_ITEM
Type	Hidden
Settings	
Value Protected	<input checked="" type="checkbox"/>
Layout	
Appearance	
Advanced	
Source	
Form Region	- Select -
Type	Null
Used	Only when current value in session state is null
Maintain Session State	Per Session (Disk)
Default	
Server-side Condition	
Security	
Authorization Scheme	- Select -
Session State Protection	Unrestricted
Store value encrypted in session state	<input checked="" type="checkbox"/>
Restricted Characters	All characters can be saved.

Type	Display Only
Label	
Label	New
Settings	
Format	Plain Text
Based On	Plain Text
Show Line Breaks	HTML

Items

Application items

- Used for Server-side logic items
- Set SSP to Restricted
- Very often forgotten and left exposed

User editable items

- Page - Set SSP to Checksum
- Validate and sanitize before submitting it to DB
- Do not trust user inputs

Hidden items

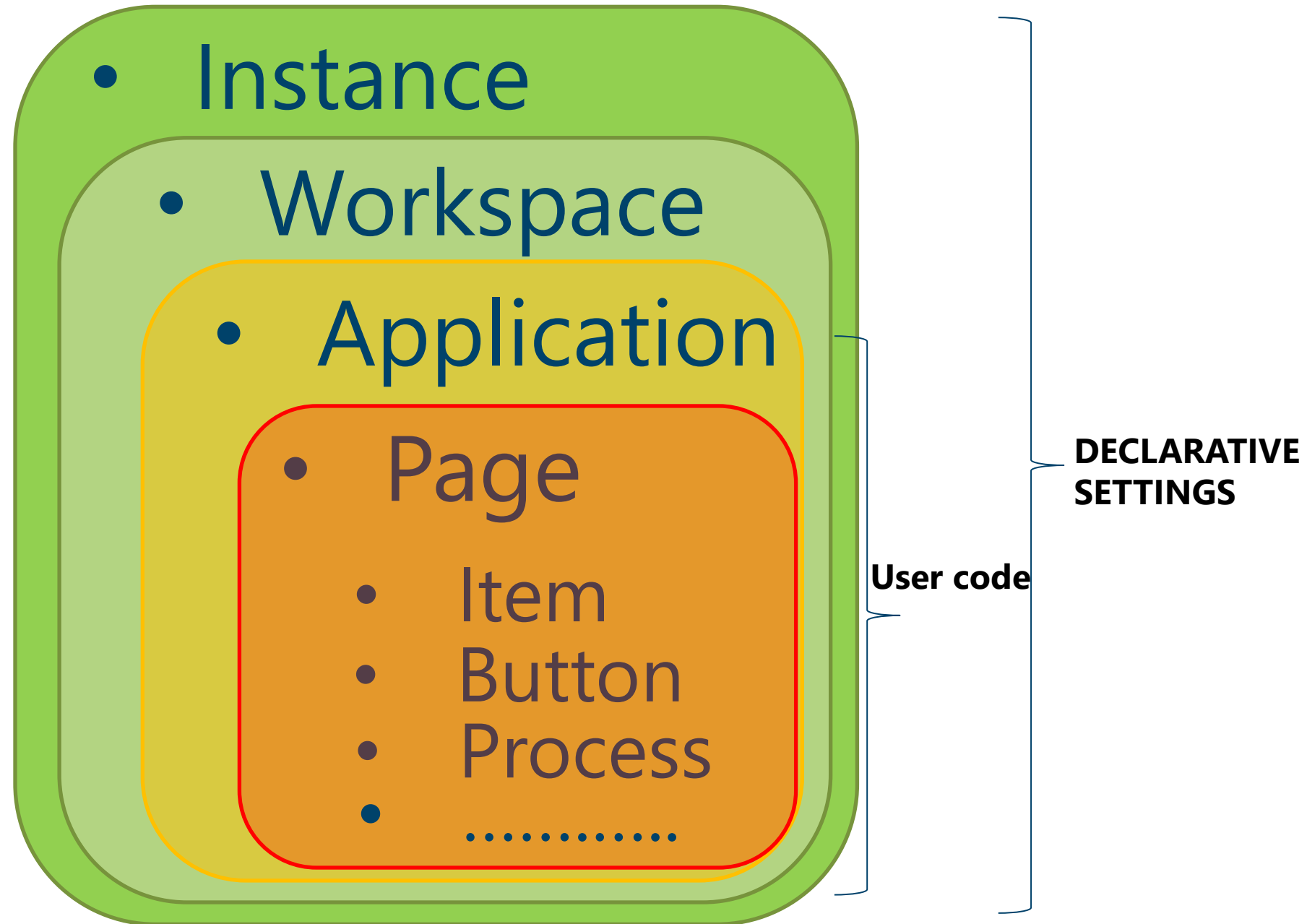
- Not displayed but rendered in an HTML
- Hidden item can be modified
- If JavaScript modifies the value then it cannot be fully protected
- For Items passed as data between pages:
 - Set Page SSP to Checksum
 - Set PAP of receiving page to Checksum
 - Set Value Protected to Yes

Display items

- Non-enterable text item
- Values can still be changed
- Restricted - May not be set from browser
- Use escape special character or Format Settings like plain/HTML/markdown
- Make sure to check:
 - if it can be set by end users or if it used in JS as substitution string

TIP #3

Important settings



Settings – Application Definition

- 150+ APEX settings in total
- Rejoin session & Deep Linking
- Session Timeout & Session idle in seconds
- Error handling function
- Session State Protection (SSP)
- Authorization & Authentication

Settings – Page Security

- Authorization Scheme
- Authentication
- Rejoin Sessions
- Deep Linking
- Page Access Protection (PAP)
 - PAP does not provide any security but is required when SSP is enabled
 - Unrestricted
 - Arguments Must Have Checksum
 - No Arguments Supported
 - No URL Access
- Form Auto Complete & Double submission

Error handling function

- We can easily define one
- Minimize amount of information we give back to the users
- Use it to improve user experience

TIP #4

Where (should) we use HTML?
like substitution variables

APEX and built in HTML

- Everywhere where APEX lets us?
- Watch out for Substitution variables!

OK

- Page title, region title and process success message

NOT OK

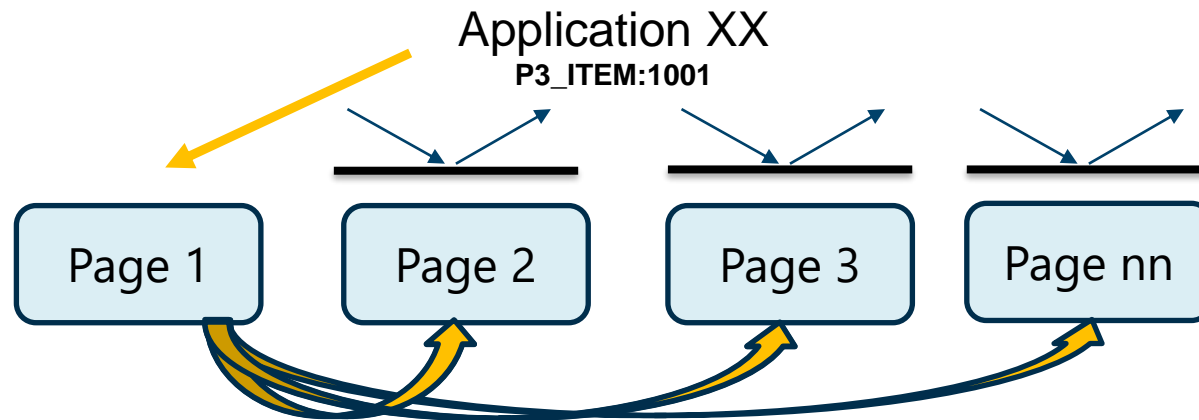
- Substitution variable in JavaScript
- Report columns title
- Breadcrumbs entries

TIP #5 APEX URL

Remember

`f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues`

- Tampering is the first way
- Changing URL parameters or changing the values using JavaScript



URL is powerful

- It can run your processes too
 - Page or Application process

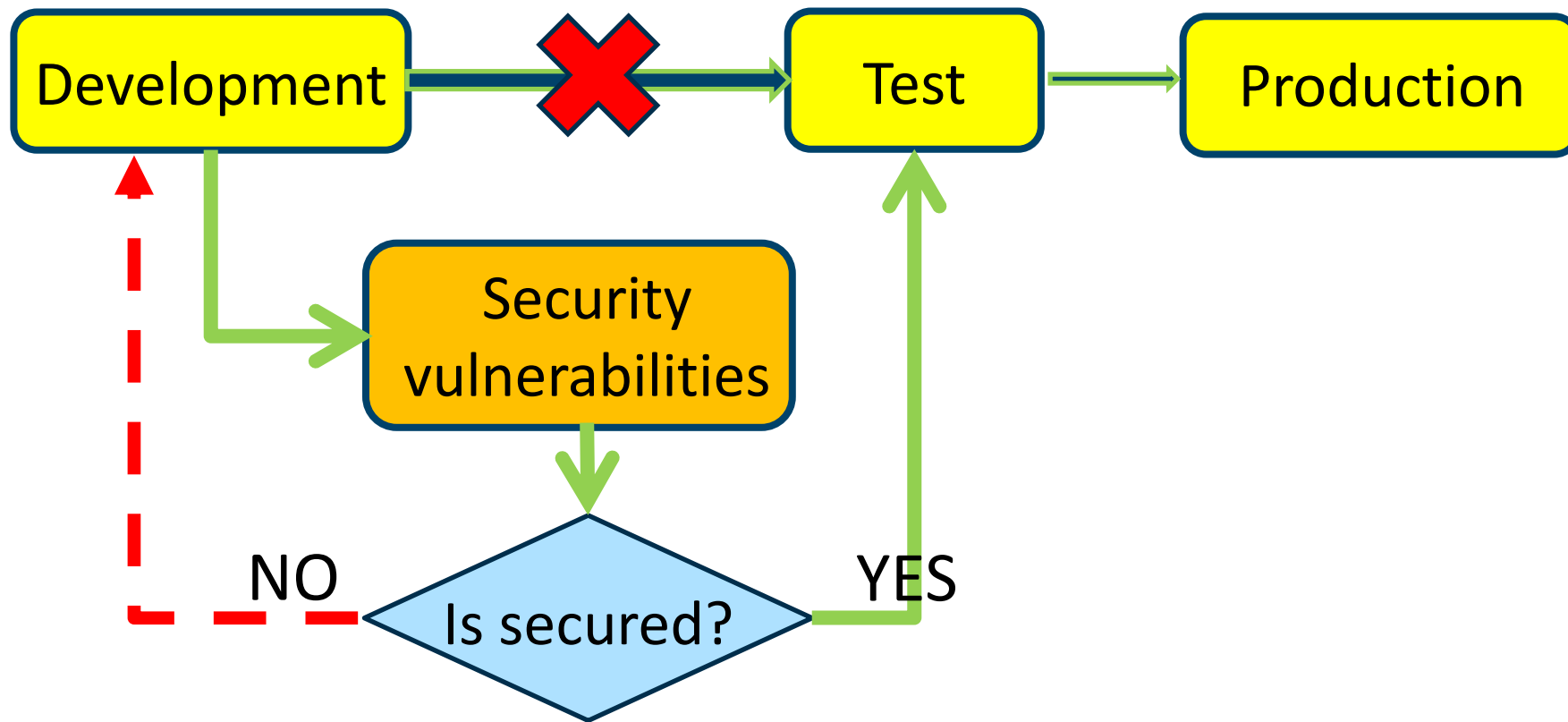
▼ Demo 2 - Application process - On Demand

There is Application process - On Demand.
what happens if I do now

f?p=96792:9999:106983718520082:APPLICATION_PROCESS=DUMMYPROCESS:::

Improvements

Idea?



Automated testing

WHY Automated testing?!

- **Improves software quality**
- **Helps with APEX security**
- Shorter test cycles
- Saves Time and Money
- **Faster APEX Upgrading**
- Return on investment is high

WHY Automated testing?!

- Reduces manual errors
- Improves Standards
- Faster time to market
- Faster Feedback
- Improved Productivity
- Faster Debugging

Testing tools

- Selenium
- Cypress
- Ghost Inspector
- Playwright?!!!!
- utPLSQL
- LCT

DIY – part 2

Monitor and logging

- All invalid authentication are logged

```
select * from apex_workspace_access_log
```

OR

```
select * from apex_workspace_activity_log
```

- Attackers will try to break into the system
- Logs are kept for 2 weeks so consider backing them up

Search

Total Row Count 52

Q

Search...

Go

Language

- ☐ PL/SQL (22)
- ☐ SQL (16)
- ☐ JavaScript (14)

Scope

- ☐ Shared Component (2)
- ☐ Page 4 - Access-control vulnerability (1)
- ☐ Page 7 - Error handling function (1)
- ☐ Page 11 - XSS Cross Site Scripting (3)
- ☐ Page 12 - SQL Injection (1)
- ☐ Page 15 - Display and hidden Items (13)
- ☐ Page 16 - Report, toggle, card (12)
- ☐ Page 17 - Line charts (15)
- ☐ Page 9999 - Login Page (4)

Component Type

- ☐ Region (16)
- ☐ Action (15)
- ☐ Process (10)

Search Results

Shared Component
Application Process: DUMMYPROCESS

Begin
 htp.p('Hello there!!');
End;

Shared Component
Authorization: ADMINISTRATION RIGHTS

if :APP_USER != 'LSCHILDE@GMAIL.COM' then return true; else return false; end if;

Page 15 - Display and hidden Items
Page: DISPLAY AND HIDDEN ITEMS

```
function myFunction(p1) {  
    alert (p1);  
}
```

Page 11 - XSS Cross Site Scripting
Region: REPORT COLUMN HTML EXPRESSION LINK

```
select "ID",  
      "USERNAME",  
      "FIRSTNAME",  
      "SURNAME"  
      , apex_escape.js_literal('Full name is ' || firstname || ' ' || surname)  
      as jsfullname  
      from "APEX_210200"."DEMO_USERS_XSS"
```

Application x / Utilities / Embedded Code

APEX Views - Information

- We can do lots using APEX views:
 - **apex_applications**
 - apex_application_static_files
 - apex_application_processes
 - apex_application_pages
 - **apex_application_page_regions**
 - **apex_application_page_proc**
 - apex_application_page_items
 - apex_application_page_buttons

...is in meta data

- apex_application_page_reg_cols
- **apex_application_page_rpt_cols**
- apex_appl_page_igs
- **apex_appl_page_ig_columns**
- apex_application_page_ir
- **apex_application_page_ir_col**
- apex_application_page_da_acts
- apex_application_list_entries
- apex_application_lists
- apex_application_lovs
- apex_application_page_map_layers

APEX Views - We can find

- **Application settings**
 - Application processes with no authorizations
 - Application processes for Dynamic SQL
 - Application computations for Dynamic SQL
- **Page settings**
 - Page regions with no authorizations
 - Page processes with no authorizations
 - Page regions with Dynamic SQL
 - Page regions with substitution variables
 - Buttons with no authorizations

...and check

- Reports with JS links
- Reports with non escaped items
- Global and on page JS
- **Region sources for SQL injection**
- **Dynamic PLSQL region**
- Deprecated APIs
- Deprecated APEX features (**tabular forms**, date pickers)

...and address

- List items with no authorizations
- Items with invalid settings (hidden or display)
- Items default values (source) checks for Dynamic SQL
- **Editable IGs with no authorizations**
- Page Computations
- Page Validations
- **Dynamic actions with substitution variables**
- **Page JS with substitution variable**

DEV Processes

- Export apps
 - With No debug and as Run Application Only
- Run your APEX Advisors/testing tools
- Have internally built tool to monitor common settings

DEV Apps

Details	Current		Should be
Application	1111 - SB demo		
Embedded in Frames	Allow from same origin		
Application version	1.5.0.2		
Compatibility	19.2		
Default Theme	Universal Theme		
Theme Subscribed From	99999. Style		
Theme Style	SB Style		
Global Page Regions Without Server Side Condition	2	Details	
Global Page Dynamic Actions Without Server Side Condition	2	Details	
Authentication Scheme	SB Authentication		
Button Region Position "Top and Bottom"	3	Details	DEPRECATED (ab APEX 21.2)
Interactive Grids	13	Details	
JavaScript other	6	Details	
Error Handling Functionn			apex_exc_error_handling.apex_error_handling



Quick test

- Item conditions are safe?
- APEX JS alert example
- Commented code?
- APEX API can not be abused?
- APEX built-in substitution variable?

Code Editor - Code



```
1  alert("Entered value &P10_ITEM.");
2
3  alert("Entered value &P10_ITEM!JS.");
4
5  alert("Entered value " + $v('P10_ITEM'));
6
7  alert("Entered value " + apex.item("P10_ITEM").getValue());
8
9  alert("Entered value " + apex.items.P10_ITEM.value );
10
```



Demo 7 - Condition injection demo

Display Only
Showing

Submit



Server-side Condition

Type

Expression



Language

PL/SQL



PL/SQL Expression



```
length('&P9_MSG.') > 8
```



```
1  DECLARE
2  l_text CLOB;
3  l_email VARCHAR2(50);
4  l_name VARCHAR2(50);
5  BEGIN
6      SELECT 'dummy@demo.com' email, ' there!!!!!!' name into l_email, l_name
7      from dual;
8
9      --l_text := 'LAST MESSAGE: &P9_COMMENT..' || utl_tcp.crlf;
10     l_text := 'LAST MESSAGE: ' || :P9_COMMENT || utl_tcp.crlf;
11
12     l_text := l_text || ' Changed by &APP_USER.' || utl_tcp.crlf;
13     l_text := l_text || ' Hello ' || l_name || utl_tcp.crlf;
14
15     htp.p(l_text);
16 END;
```

```
8 | l_text := 'LAST MESSAGE: ' || apex_escape.html(:P9_COMMENT) || utl_tcp.crlf;
```

Source

Form Region

- Select -

Type

Expression

Language

PL/SQL

PL/SQL Expression



```
apex_util.prepare_url('f?p=&APP_ID.:&APP_PAGE_ID.:&APP_SESSION.::::P9_MSG:&P9_MSG.')
```



ā'pěks
(#orclapex)

is
SAFE

Summary

- Do not trust user inputs
 - Sanitize & validate
 - Make sure it gets escaped when used
- Protect your items
 - We can not change values using URL or JavaScript
 - Use SSP, restricted characters and value protected
- Be careful with using &PX_ITEM. syntax
 - Region, error & success messages are safe places
- Use security & auto-testing tools
- Build your own checks based on APEX views

Summary

- Minimum is to utilize authentication and authorization policies as minimum
- Authentication - “best practices”
- Error handling function
- Inconsistencies can lead to **vulnerabilities**
- Keep APEX up-to date

Summary SQL Injection

- Do use bind variables with care:
 - ! If used in Dynamic SQL bind variable needs to be embedded in the string
- If you are forced to use &ITEM. syntax:
 - Check where data is coming from
 - Use appropriate escaping methods available
- DBMS_ASSERT and APEX_ESCAPE
- Reduce the impact surface

Summary SQL Injection

- APEX_COLLECTION
 - CREATE_COLLECTION_FROM_QUERY
 - CREATE_COLLECTION_FROM_QUERY2
 - CREATE_COLLECTION_FROM_QUERY_B
 - CREATE_COLLECTION_FROM_QUERY_B2
 - MERGE_MEMBERS
- APEX_ITEM
 - POPUP_FROM_QUERY
 - POPUPKEY_FROM_QUERY
 - SELECT_LIST_FROM_QUERY
 - SELECT_LIST_FROM_QUERY_XL
 - TEXT_FROM_LOV_QUERY
- DBMS_SQL.PARSE

Summary XSS

- Restrict, sanitize, validate and escape user inputs per context
- APEX_ESCAPE
- APEX_JAVASCRIPT.ESCAPE
- APEX_UTIL.URL_ENCODE
- HTF.ESCAPE_SC
- apex.util
 - escapeCSS, escapeHTML, escapeHTMLAttr

Summary XSS

- &PX_ITEM!**HTML**.
- &PX_ITEM!**JS**.
- &PX_ITEM!**ATTR**.
- &PX_ITEM!**RAW**.
- &PX_ITEM!**STRIPHTML**.

How much can we do ourself?

Yes we can do most!

Q&A

Thank you for attending

