Help Me – I am Running Out Of Ideas!

Can Threat Modeling Facilitate Pen-Testing?

Joern Freydank

https://www.linkedin.com/in/joernfre/

Intro/Bio

Bio

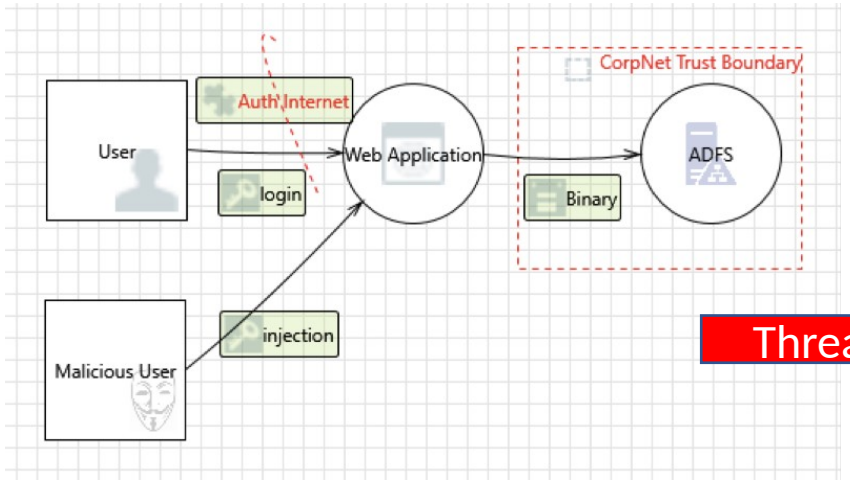
- TU-Berlin, Germany (MSc. Computer Engineering)
- Started in security by dongle cracking
- Software Developer embedded and Enterprise Systems
- 20+ years Experience, CISSP
- Bank/ATM/Payment Security
- Splunk (Cisco) Principal Product Security Engineer/Technical Leader

Experience in both Pen-Testing and Threat Modeling!

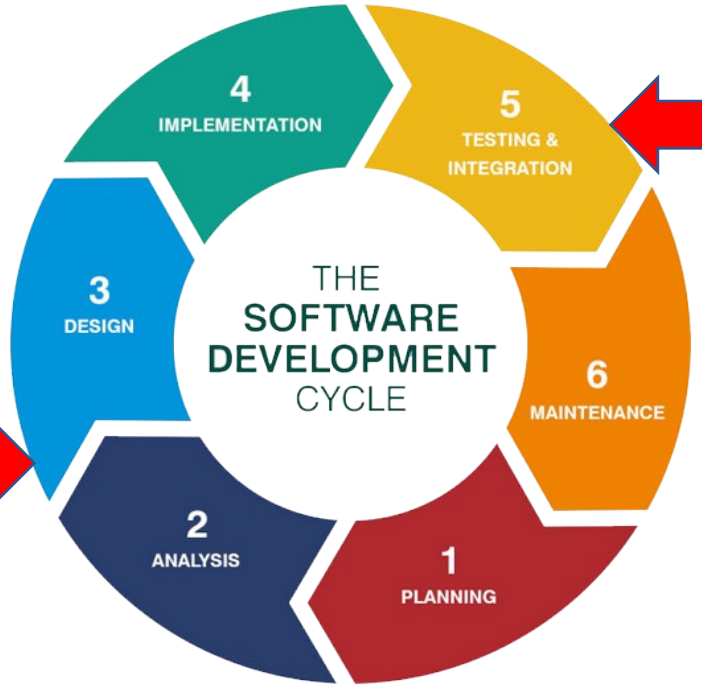
Email: joernfre@yahoo.com
LinkedIn: https://www.linkedin.com/in/joernfre/

Threat Model and Pen-testing in the SDLC

- Identified Threats to Custom Systems
- Created early during Software Design Phase
- Identifies (missing) controls
- Provides Security Requirements



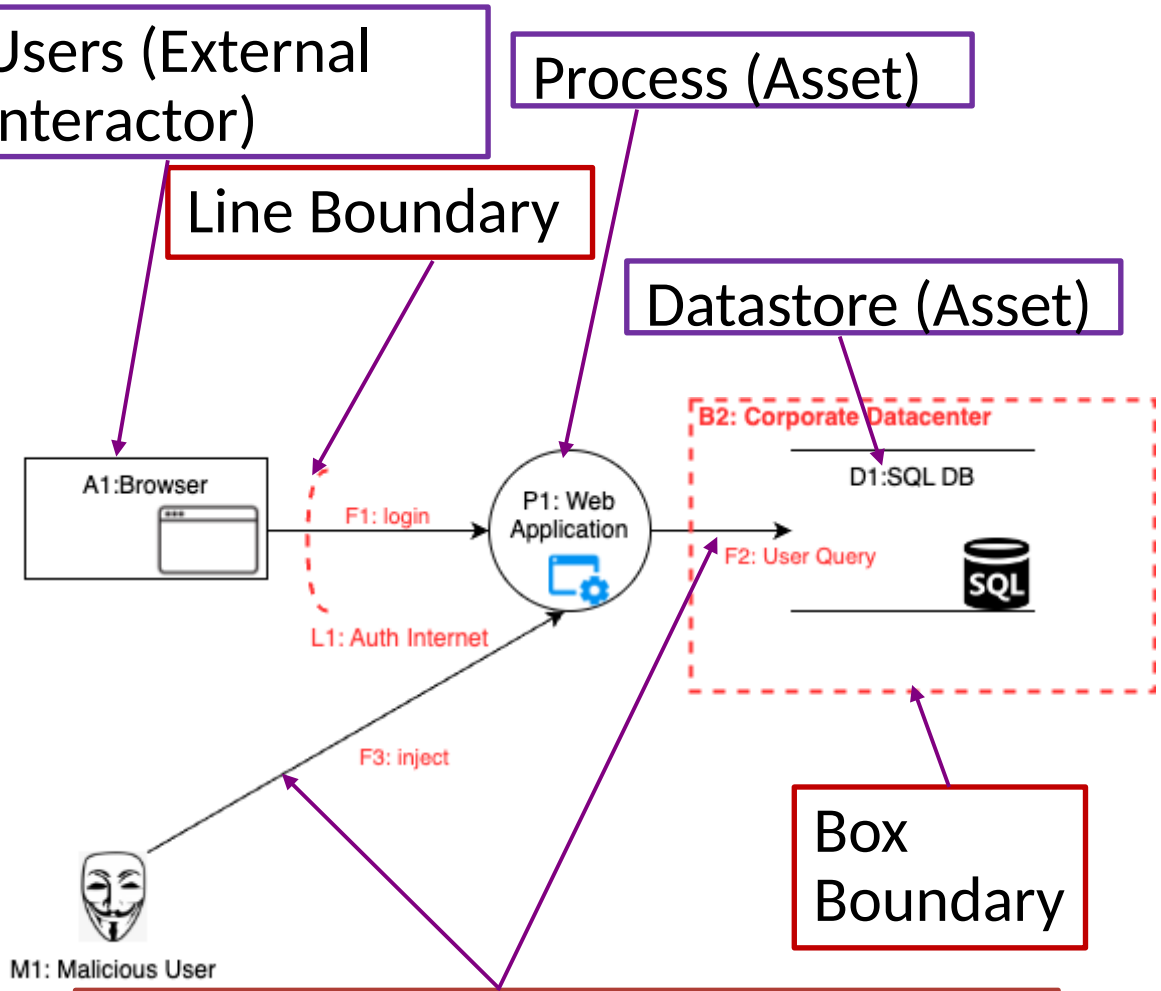
Threat Model



Pen-testing

- Consumed by Pen-testers
- Provides Reference

Threat Model Example



No Control: SQL Injection - Threat becomes Vulnerability

Identified Threats (Web Application)

(S) Spoofing of Identity of User

✓ Login Control

(I) Information Disclosure of PII

✓ TLS 1.3 (Control)

(E) Elevation of Privilege of Admin role

✓ Authorization Token (Control)

Identified Threats (DB)

(S) Spoofing of Identity of Web App

✓ Client Certificate (Control)

(I) Information Disclosure of PII

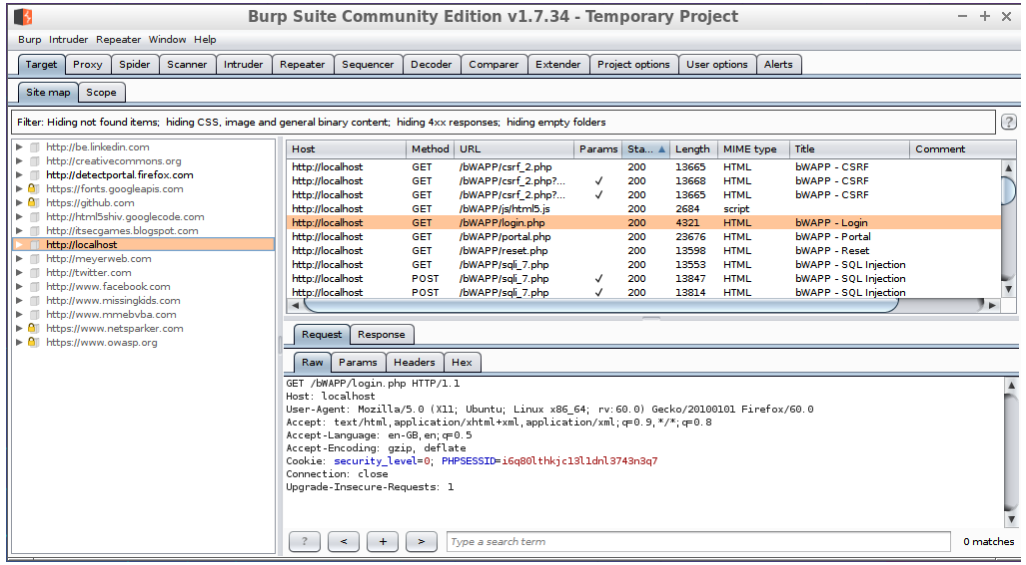
✓ VPN Protocol

(T) Tampering SQL Injection

X No Sanitization Control

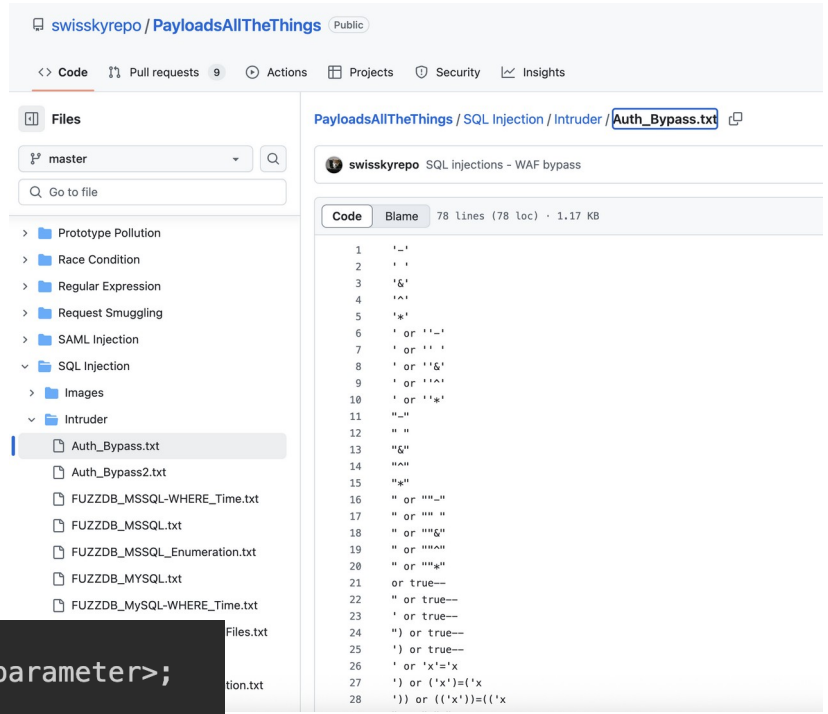
Pen-Test Example – SQL Injection Bypass

1. Select Tools: Burp Suite



2. Select Payload: SQL Injection List

Source: https://github.com/swisskyrepo/PayloadsAllTheThings/



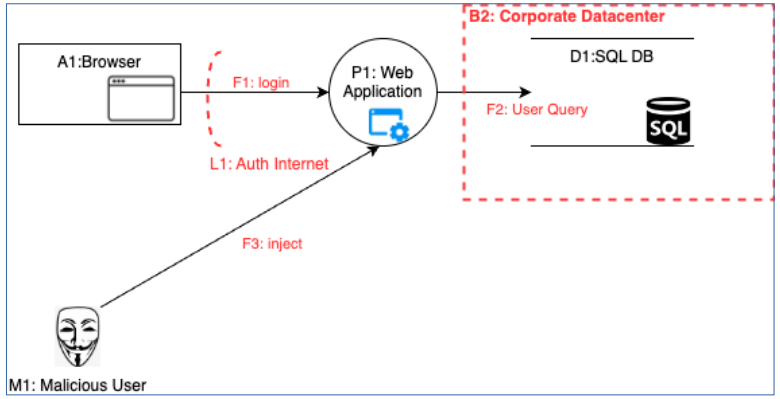
3. Inject SQL into logout parameter

```
SELECT user_session FROM users WHERE username = <username_url_parameter>;
```

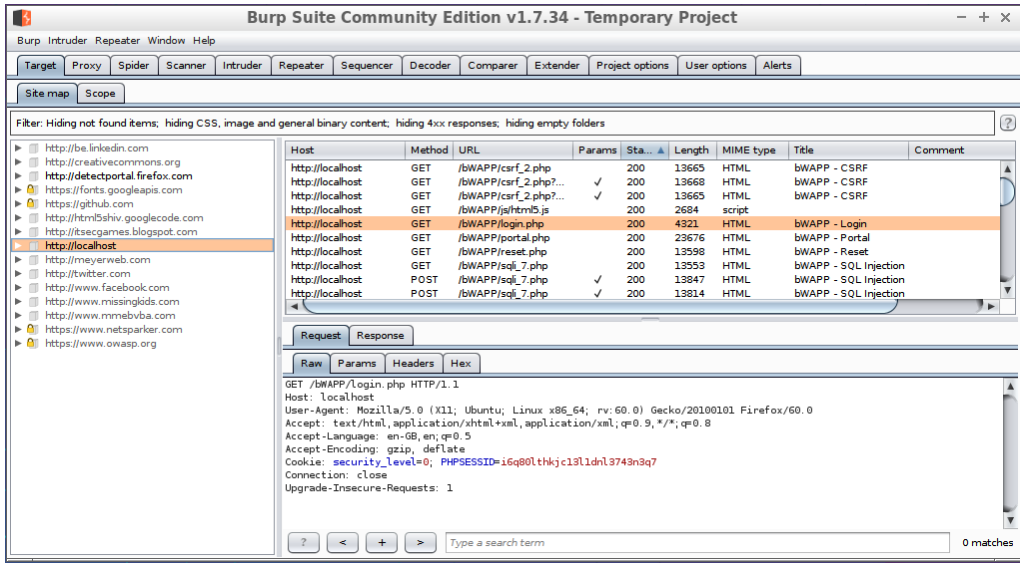
```
1' UNION SELECT username, password FROM users --
```

```
SELECT user_session FROM users WHERE username = '1' UNION SELECT
username, password FROM users --';
```

Threat Modeling as Input to Pen-Testing



?

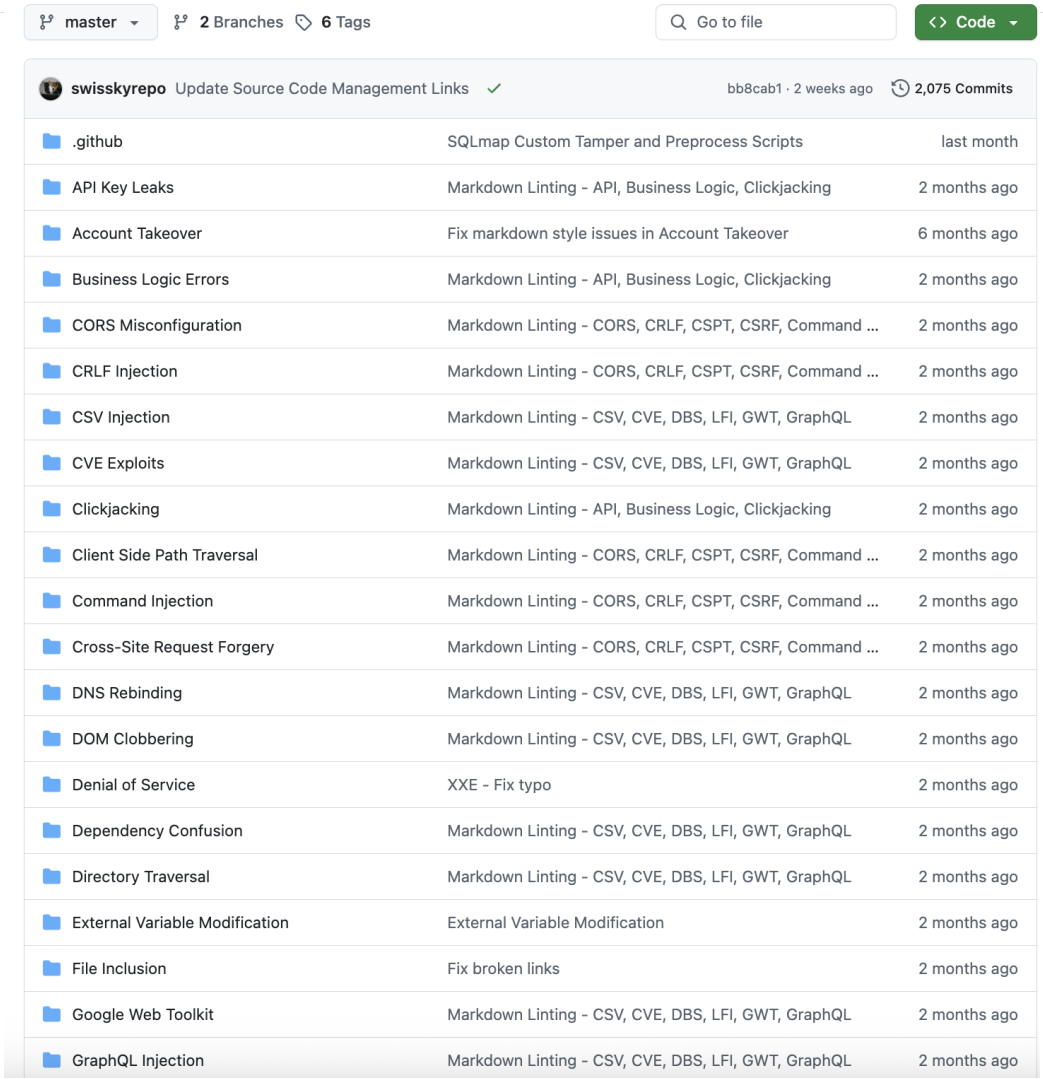



```
1' UNION SELECT username, password FROM users --
```

Threat Model

Pen-Test

Threat Modeling => Pen-Testing — Lay of the Land/Reconnaissance



master · 2 Branches · 6 Tags · Go to file · Code

swisskyrepo Update Source Code Management Links ✓ bb8cab1 · 2 weeks ago · 2,075 Commits

.github	SQLmap Custom Tamper and Preprocess Scripts	last month
API Key Leaks	Markdown Linting - API, Business Logic, Clickjacking	2 months ago
Account Takeover	Fix markdown style issues in Account Takeover	6 months ago
Business Logic Errors	Markdown Linting - API, Business Logic, Clickjacking	2 months ago
CORS Misconfiguration	Markdown Linting - CORS, CRLF, CSPT, CSRF, Command ...	2 months ago
CRLF Injection	Markdown Linting - CORS, CRLF, CSPT, CSRF, Command ...	2 months ago
CSV Injection	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
CVE Exploits	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
Clickjacking	Markdown Linting - API, Business Logic, Clickjacking	2 months ago
Client Side Path Traversal	Markdown Linting - CORS, CRLF, CSPT, CSRF, Command ...	2 months ago
Command Injection	Markdown Linting - CORS, CRLF, CSPT, CSRF, Command ...	2 months ago
Cross-Site Request Forgery	Markdown Linting - CORS, CRLF, CSPT, CSRF, Command ...	2 months ago
DNS Rebinding	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
DOM Clobbering	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
Denial of Service	XXE - Fix typo	2 months ago
Dependency Confusion	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
Directory Traversal	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
External Variable Modification	External Variable Modification	2 months ago
File Inclusion	Fix broken links	2 months ago
Google Web Toolkit	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago
GraphQL Injection	Markdown Linting - CSV, CVE, DBS, LFI, GWT, GraphQL	2 months ago

Gray Box Testing: Lay of the Land

- Tool selection
- Payload Selection

Clues from Inventory of Threat Model (Example):

Webapp→ Burp Suite

Webapp → JSON Deserialization:

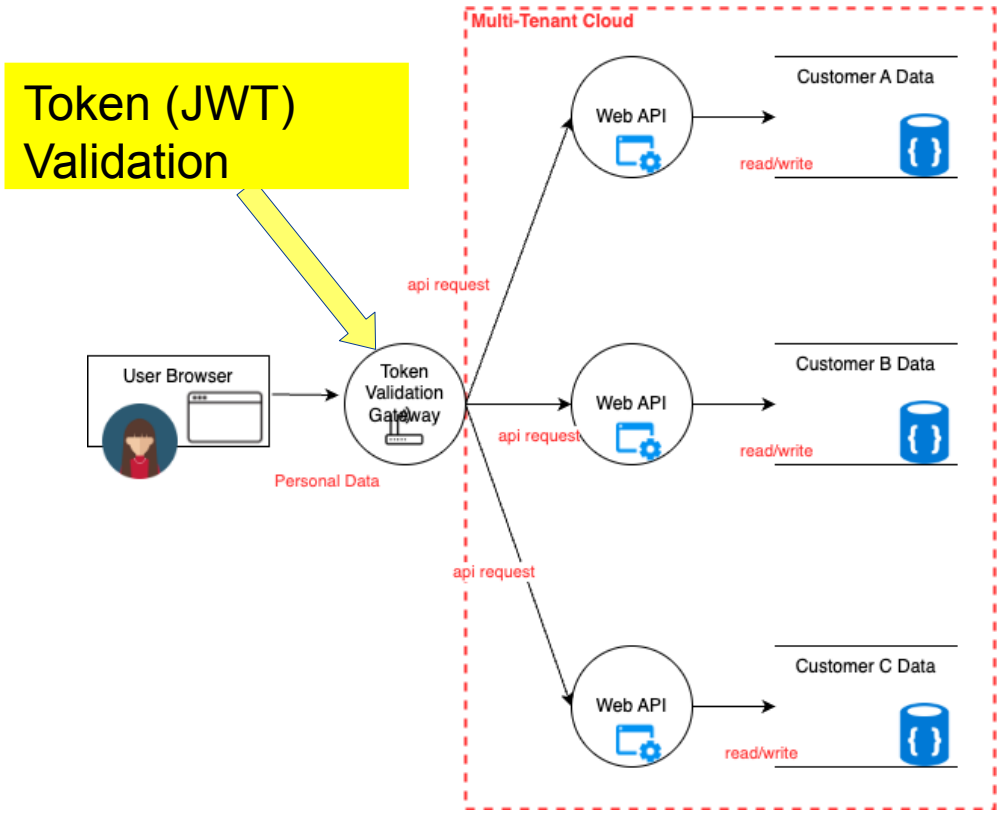
Ysoserial (https://github.com/frohoff/ysoserial)

Database →SQL Injection Payload

Black Box Testing:

- Look for clues (web application code) and build a preliminary model
- Use typical high-level deployment patterns,e.g. webapp, storage, authentication service

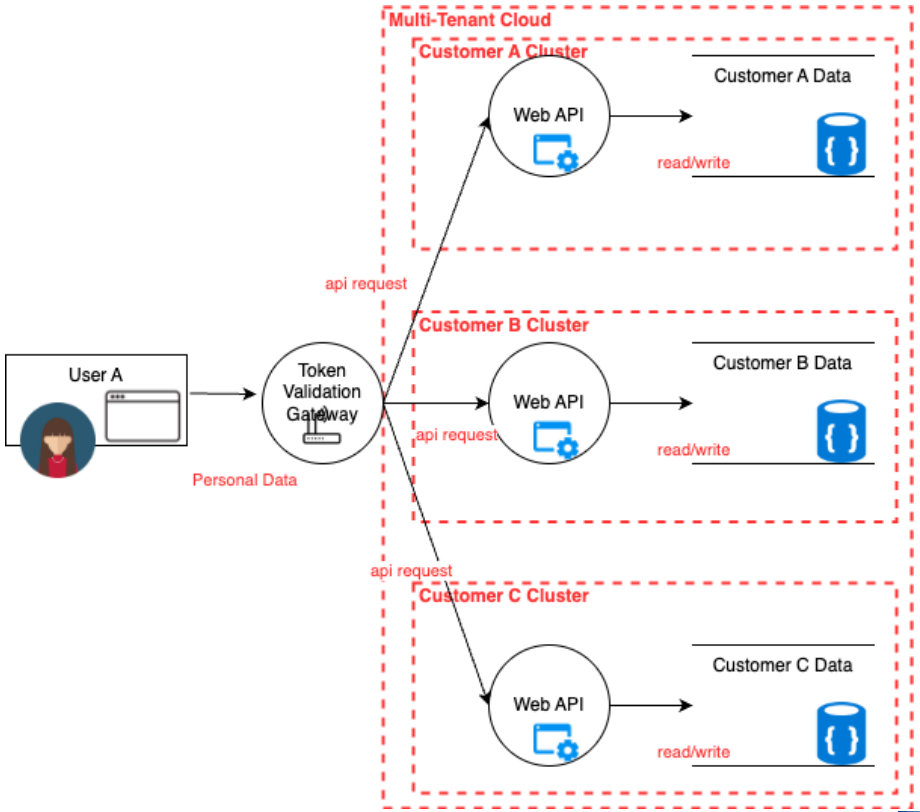
Threat Modeling => Pen-Testing - Achilles Heels



Point out Critical Controls:

Note to Pen-Testing:
Please thoroughly test JWT Validation by Gateway
Examples:
- JWT **NONE** Algorithm (Tampering)
- JSON Deserialization (Injection)
- Flooding/Fail Open? (Denial of Service)

Threat Modeling => Pen-Testing – Authorization Boundaries



Note to Pen-Testing:
Please switch out 2 (valid) tenant authorizations in a test:

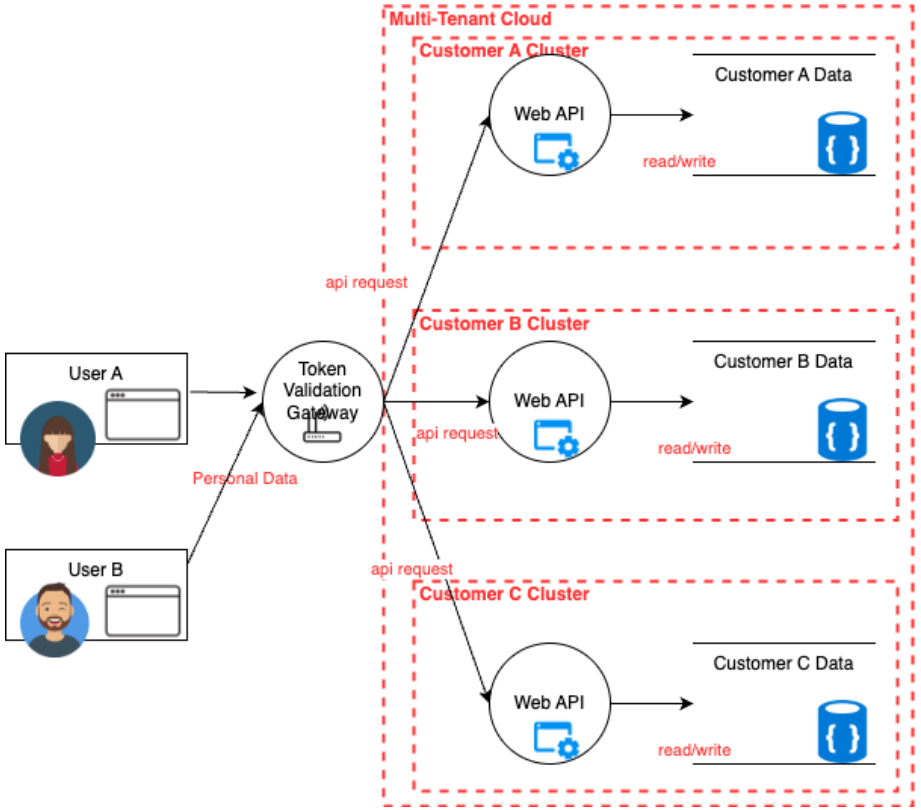
Testing tenantID in URI

https://example.com/api/<tenantID>/users-address

Authorization in JWT Custom Claims

```
{
  "email": "jane@example.com",
  "email_verified": true,
  "user_id": "customer123",
  "tenantId": "customerA"
}
```

Threat Modeling => Pen-Testing – Identify Roles

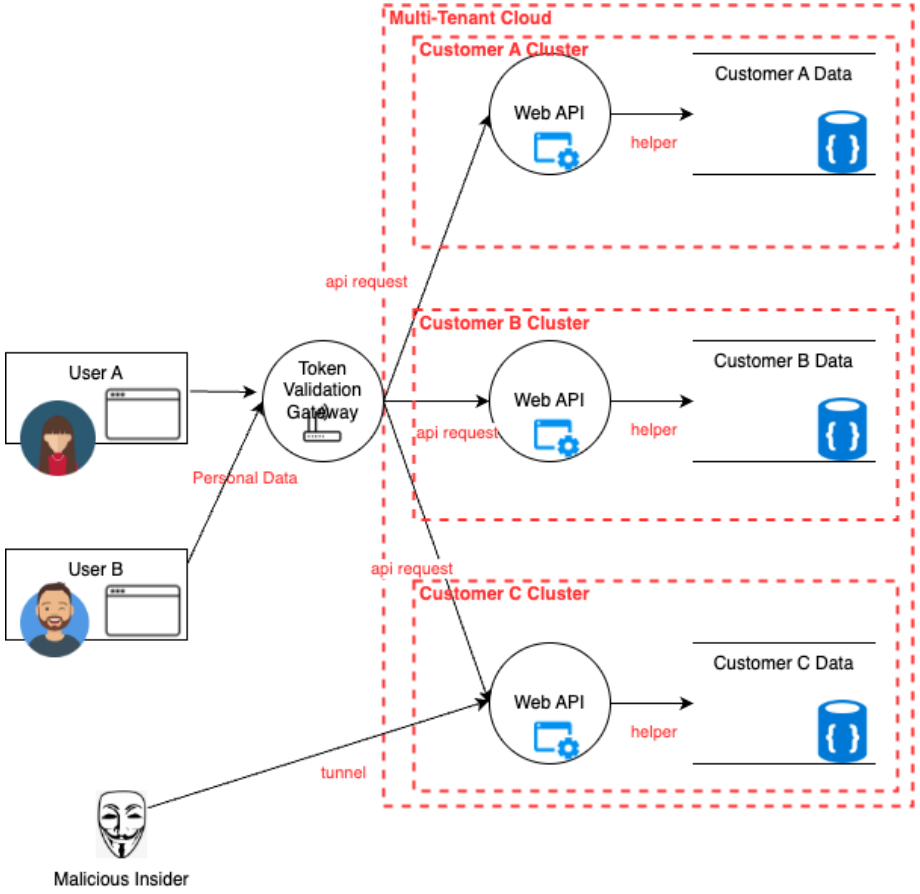


Note to Pen-Testing for setup:
- We need at least 2 different tenants
- We have 3 different roles: User and Customer's Admin and Site admin

Authorization differences between business users are frequently not obvious!

Example:
Data Authorization: Regional sales people

Threat Modeling => Pen-Testing – Insiders Threats



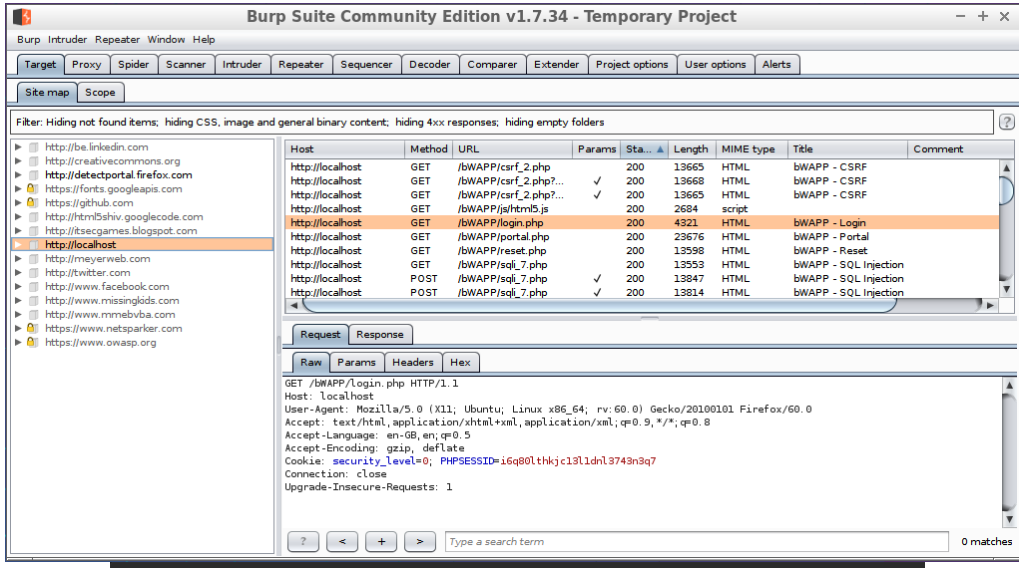
Note to Pen-Testing (purple test):

- Please scan inside one cluster and see what is reachable on other clusters
  - -- network listeners

- There are vaulted Credentials for DB access
  - -- Visibility of mapped Credentials in environment?

- There are Namespace conventions (tampering),e.g. each cluster starts with variable: '**<tenant-id>**_production'

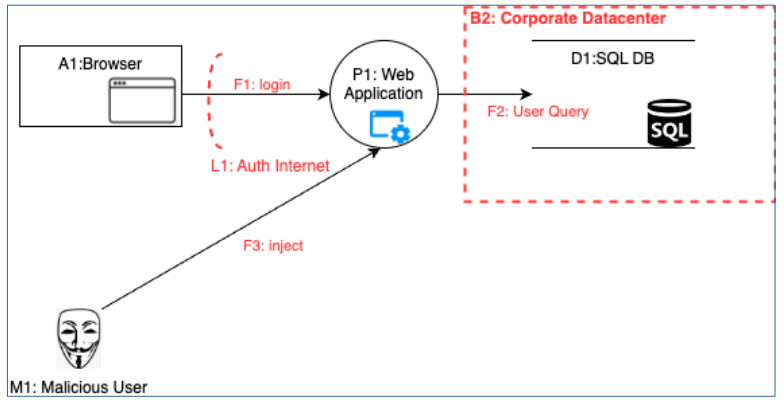
Pen-Testing as Feedback to Threat Modeling




```
1' UNION SELECT username, password FROM users --
```

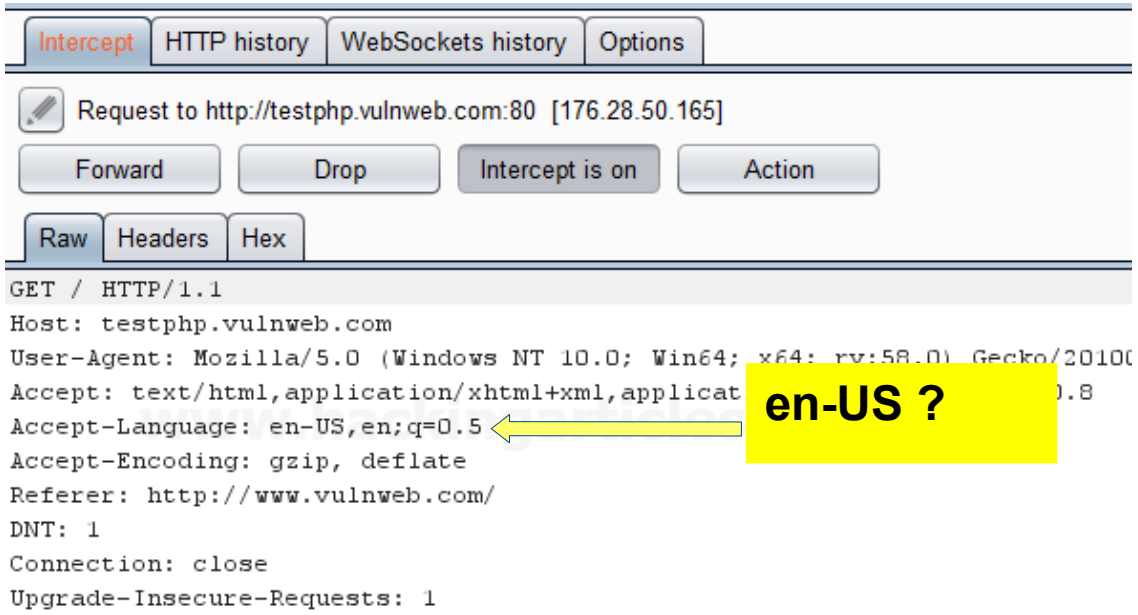
Pen-Test

?

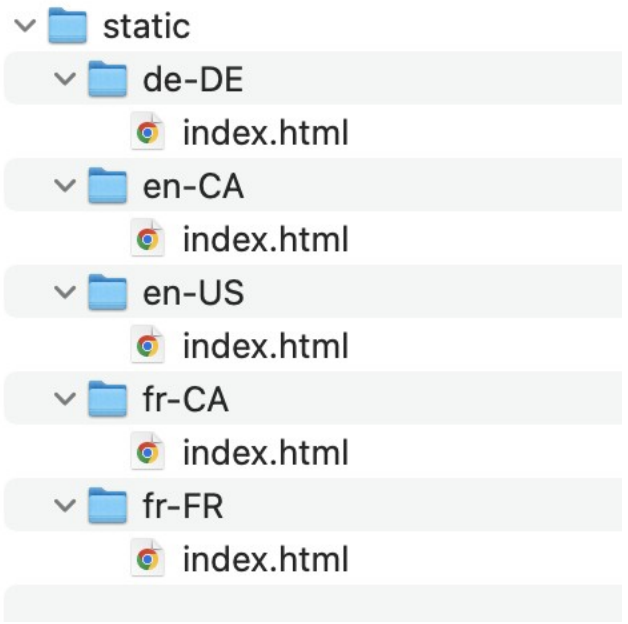


Threat Model

Pen-Testing => Threat Modeling – Exotic Inputs



https://www.hackingarticles.in/engagement-tools-tutorial-burp-suite/



Path injection if not sanitized

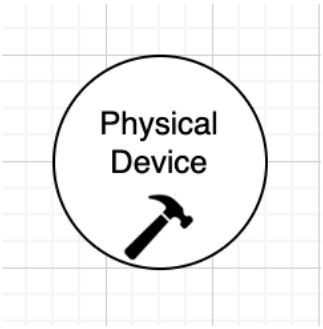
Feedback to Threat Modeling : Threat Locale injection - Input Sanitization for locale.

Pen-Testing => Threat Modeling – Missing Controls

Pen-test: Web Application running in Rack

Stencil/Library:

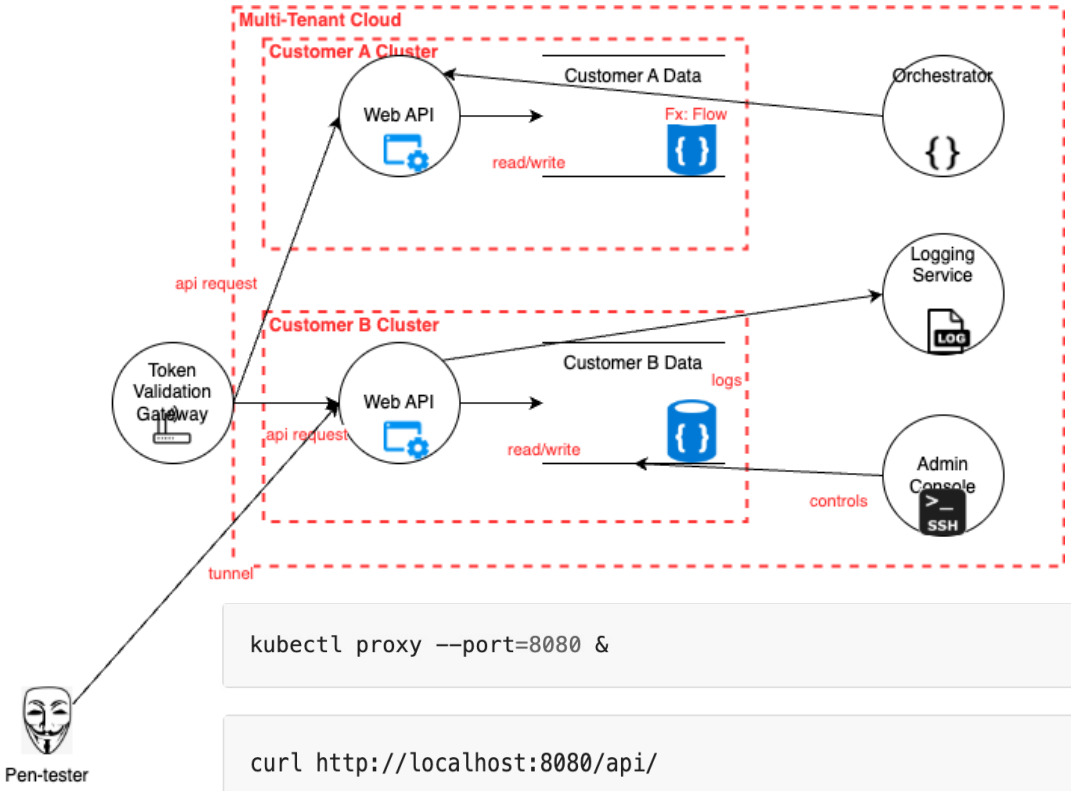
Missing Bluetooth Control



Other Listeners: e.g.
Streaming Services
(Zookeeper)

Feedback to Threat Modeling:
Add ‘Unsecured Bluetooth’ as Threat to Library of ‘Physical Device’.
Add Streaming admin service ,e.g. ‘Zookeeper’

Pen-Testing => Threat Modeling – Lateral Movement & Pivot




```
kubectl proxy --port=8080 &
```

```
curl http://localhost:8080/api/
```

Feedback to Threat Modeling:

Do the network boundaries (controls) hold up?

Admin Console visibility?

Logging Service data accessible for everyone?

Internal System API's => Authentication ?

Operational Synergy

- **Pen-test Ticket Issued** - Development team ‘failing the smell-test’ during Threat Modeling
  - Custom implemented authentication controls
  - Developers do not understand critical controls
  - HIGH or Critical Vulnerabilities discovered (oversight)
- **Scoping Call** between Pen-tester and Threat Modeler
  - Time Gap issues: Threat Model as a reference 3-6 month later
  - Model becomes instrument to remember
- **Time-Boxed** (limited time) exercises
  - What should the pen-test mostly focus on?

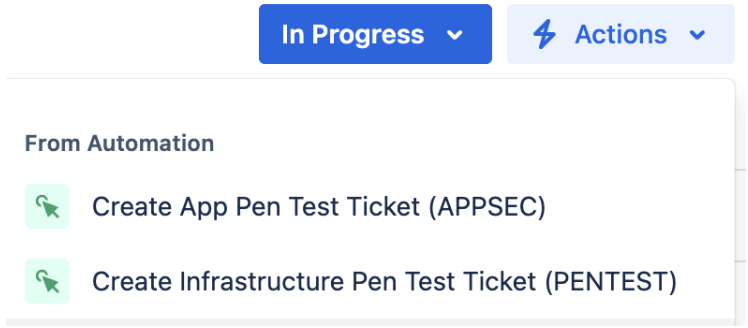
Practical Feedback Loop

- Initiate practice to **red flag or tag areas of interest** for pen-testing
  - Include a dedicated section in threat model for notes to pen-testing
  - Comment in pen-test ticket to bridge time-gap

Pen-Testing Notes

- <Notes for Pen-testing,e.g. areas of interest that are suspect or achilles heels, especially if a ticket was issued.>

Jira Actions automatically create tickets from Threat Model ticket

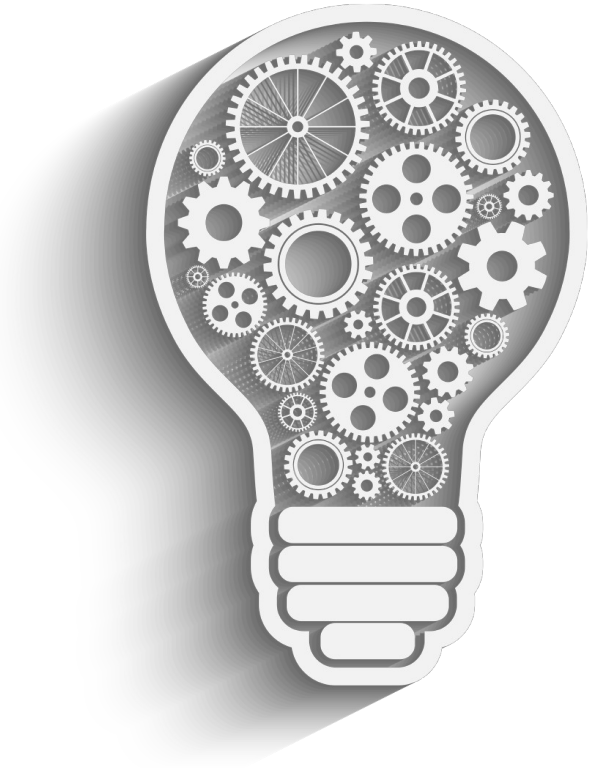


Collect Artifacts for Pen-test:
- API Specification: Swagger File
- Sample Auth Token Content

Help Me –
I Am Running Out Of Ideas!
Joern Freydank

Summary and Outlook

IDEA GENERATOR



OR

IDEA FILTER

?

RELEVANT ITEMS OF INTEREST

Establishing Communication is Key

Q&A & Discussion

Email: jfreydan@splunk.com
LinkedIn: https://www.linkedin.com/in/joernfre/