Injecting Security Controls into Software Applications

Katy Anton
Principal Application Security Consultant

About me

Katy Anton

- Software development background
- Project co-leader for OWASP Top 10 Proactive Controls (@OWASPControls)
- Principal Application Security Consultant @Veracode

@KatyAnton

@KatyAnton

Common Developer Questions

“My website is behind the firewall.
Why do I have to fix the SQL injection ?“

Common Developer Questions

"I validated the input.
Isn't this enough to prevent SQL Injection ?"

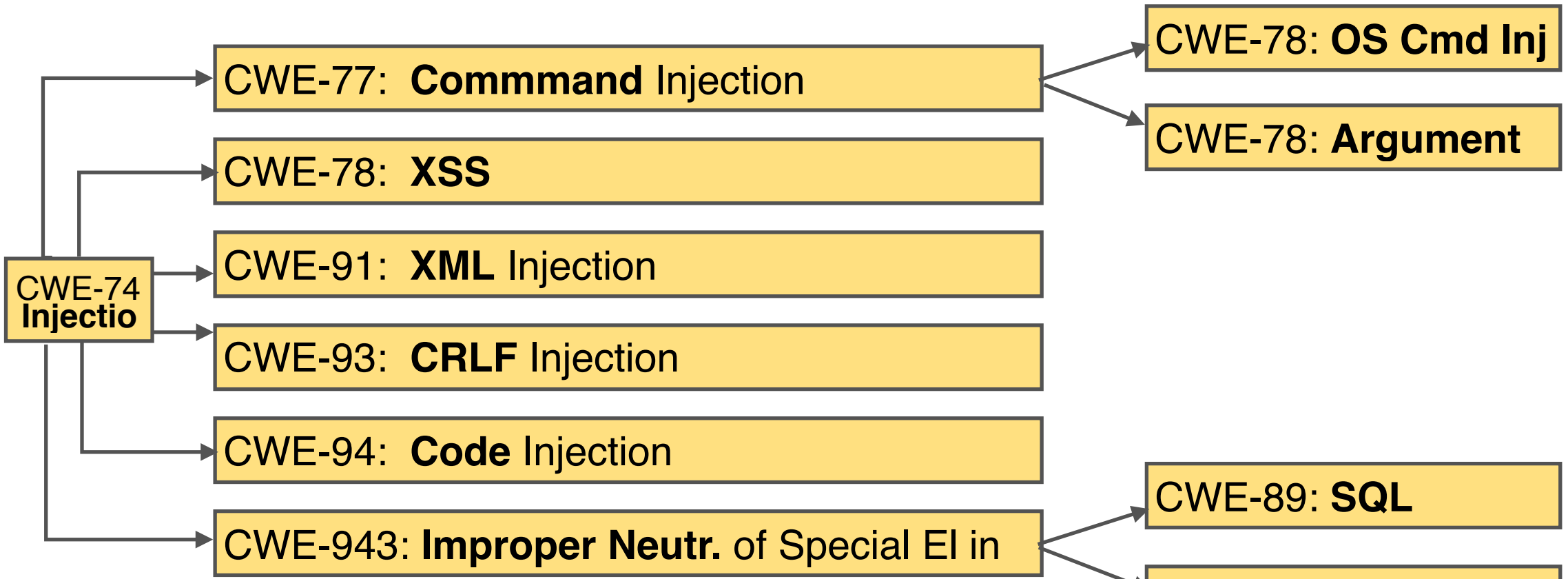
Common Developer Questions

“I have parameterized.
Look I use preparedStatement - why is not correct ?”

```
result = preparedStatement.executeQuery("SELECT * FROM users WHERE uid=':1'".replace(":1",userID));
```

Injection

CWEs in Injection Category



Source: NVD

Types of SQL Injection

- In-Band SQLi
  - Error based SQLi
  - Union based SQLi
- Blind SQL injection
  - Boolean
  - Time based
- Out-of-Band SQLi
  - Compounded SQLi (SQL + XSS)
  - Second Order SQL Injection

@KatyAnton

Injection

First mentioned in Phrack magazine in 1998

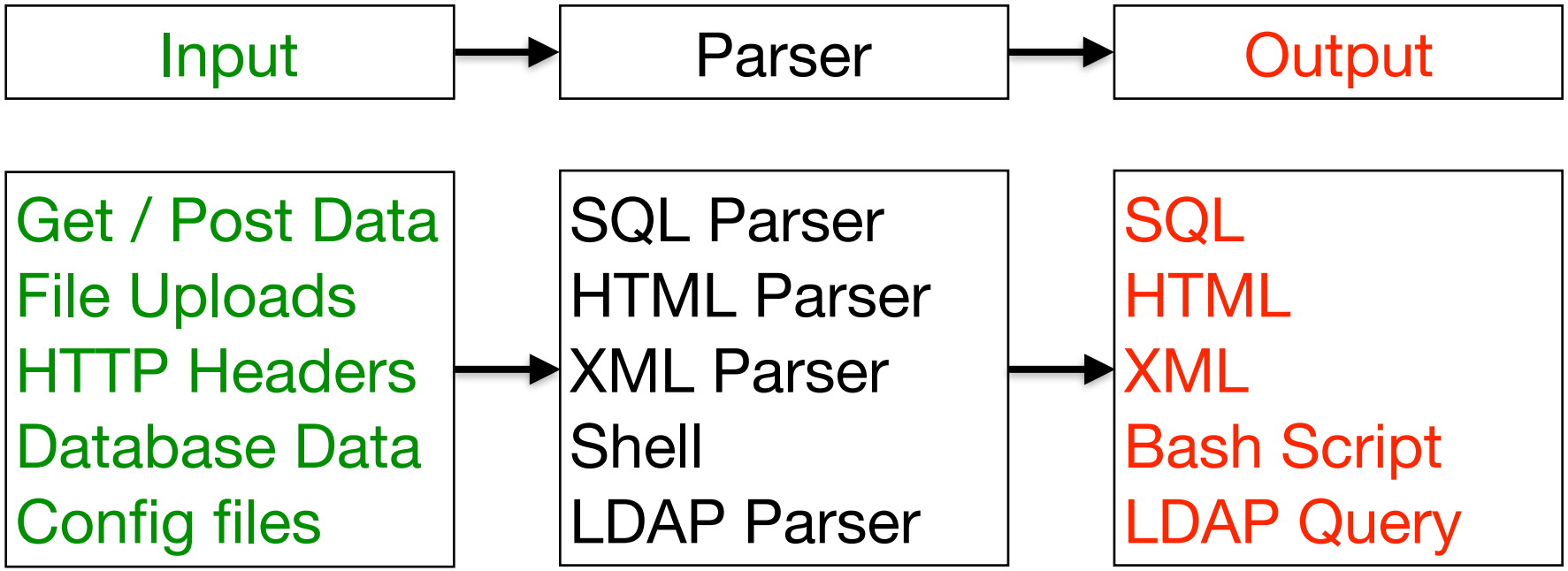
20 years anniversary

	2004	2009	2010	2013	2017
Injection	A6	A2	A1	A1	A1

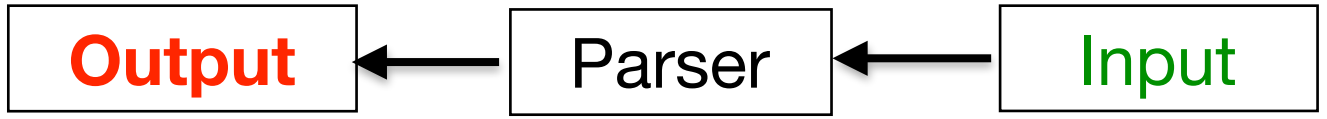
Is there another way to look at it?

Decompose the Injection

Data interpreted as Code



Extract Security Controls



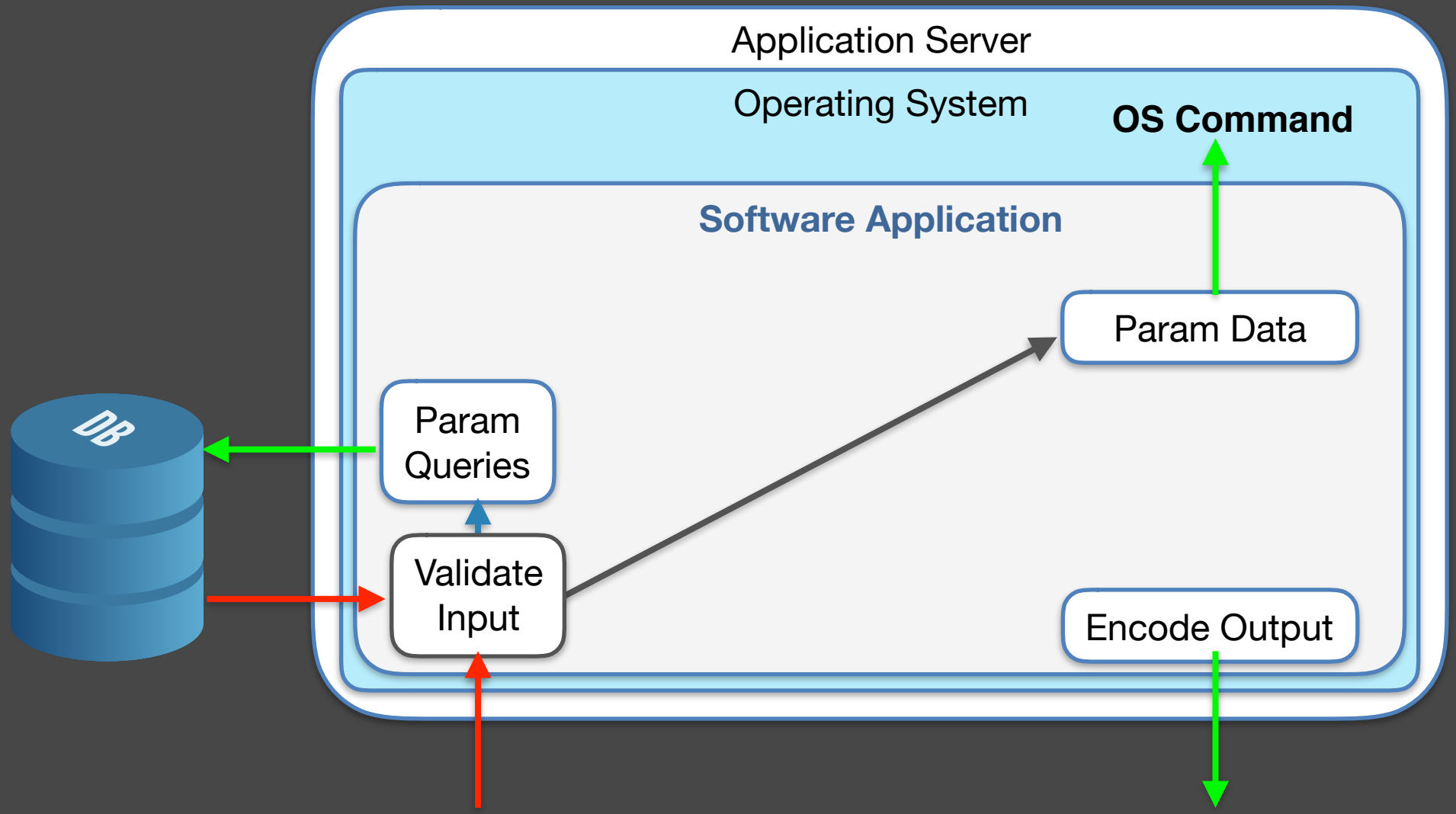
Vulnerability	Encode Output	Parameterize	Validate Input
SQL Injection		☑	☑
XSS	☑		☑
XML Injection	☑		☑
Code Injection	☑		☑
LDAP Injection	☑		☑
Cmd Injection	☑	☑	☑

Primary Controls

Defence in depth

@KatyAnton

Security Controls Recap



Intrusions
(or lack of Intrusion Detection)

“If a pen tester is able to get into a system without being detected, then there is insufficient logging and monitoring in place“

@KatyAnton

Security Controls: Security Logging

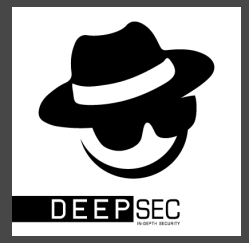
The security control developers can use to log security information during the runtime operation of an application.

The 6 Best Types of Detection Points

Good attack identifiers:

1. Authorisation failures
2. Authentication failures
3. Client-side input validation bypass
4. Whitelist input validation failures
5. Obvious code injection attack
6. High rate of function use

Source: https://www.owasp.org/index.php/AppSensor_DetectionPoints



Examples of Intrusion Detection Points

Request Exceptions

- Application receives GET when expecting POST
- Additional form or URL parameters submitted with request

Authentication Exceptions

- The user submits a POST request which only contains the username variable. The password variable has been removed.
- Additional variables received during an authentication request (like ‘admin=true’)

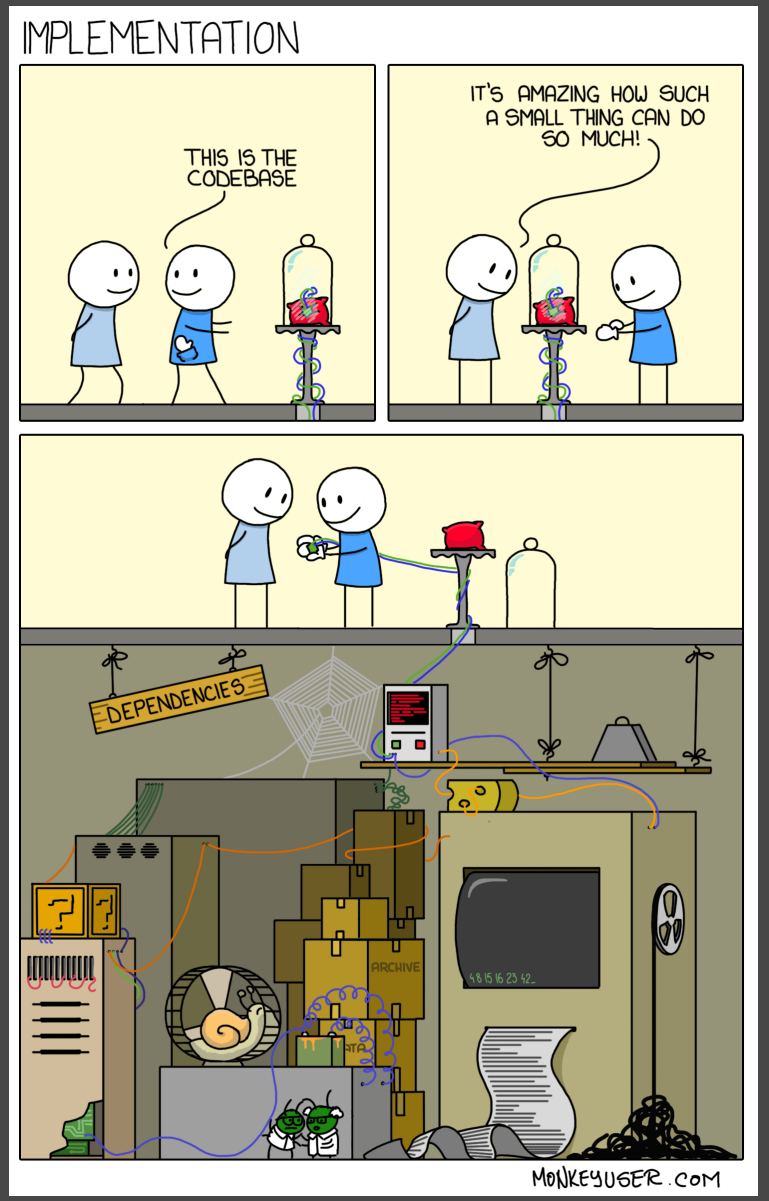
Input Exceptions

- Input validation failure on server despite client side validation
- Input validation failure on server side on non-user editable parameters (hidden fields, checkboxes, radio buttons, etc)

Source: https://www.owasp.org/index.php/AppSensor_DetectionPoints

Vulnerable Components

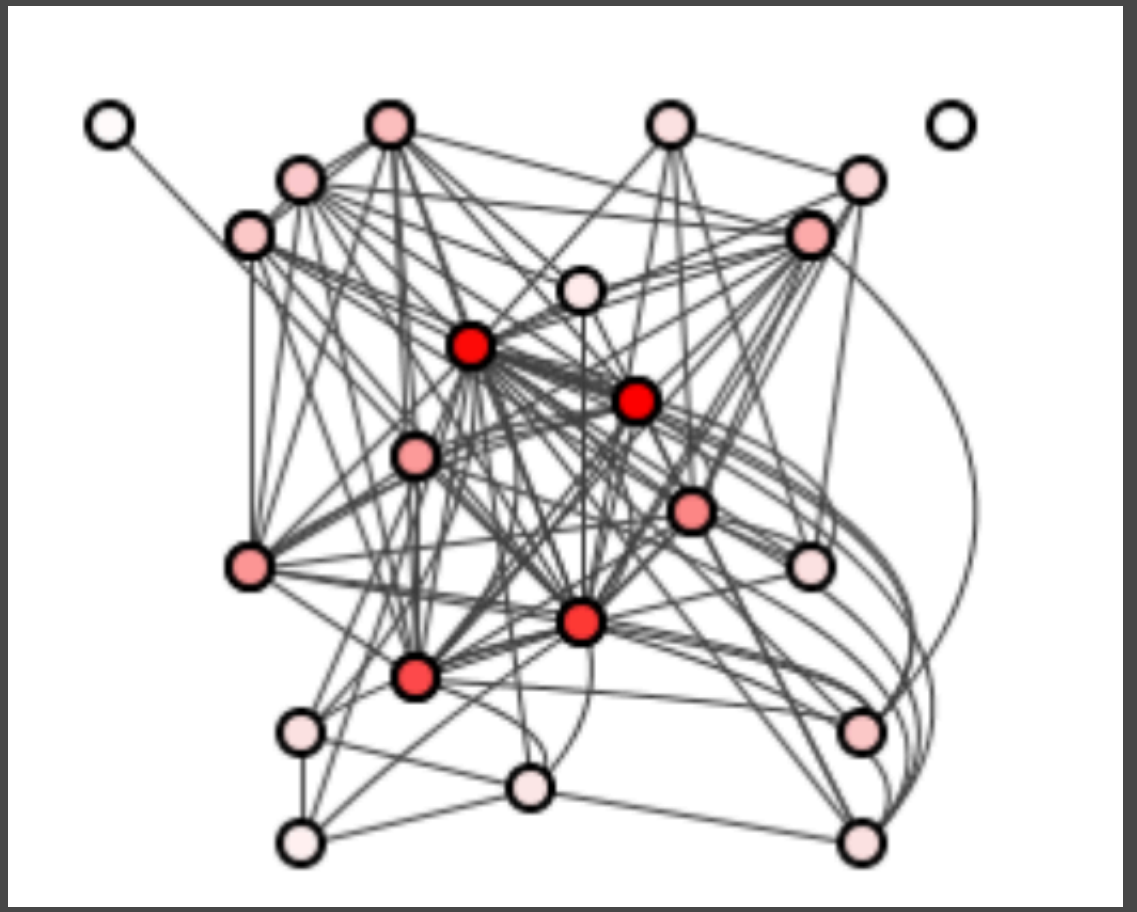
Using Software Components with Known Vulnerabilities



@KatyAnton

Root Cause

- Difficult to understand
- Easy to break
- Difficult to test
- Difficult to upgrade
- Increase technical debt



Components Examples

Example of external components:

- Open source libraries - for example: a logging library
- APIs - for example: vendor APIs
- Libraries / packages by another team within same company

Example 1: Implement Logging Library

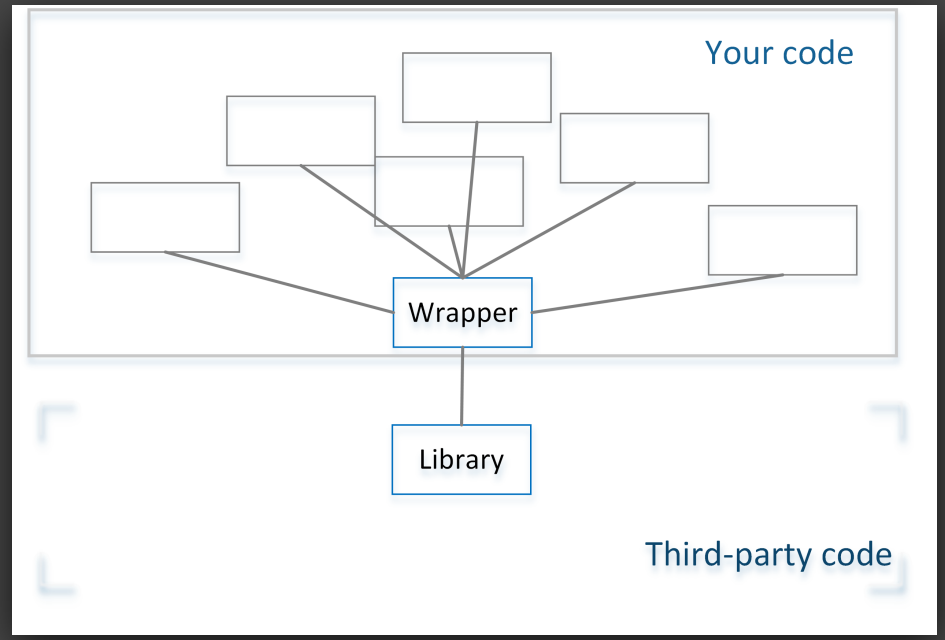
- Third-party - provides logging levels:
- FATAL, ERROR, WARN, INFO, DEBUG.

- We need only:
- DEBUG, WARN, INFO.

Simple Wrapper

Helps to:

- Expose only the functionality required.
- Hide unwanted behaviour.
- Reduce the attack surface area.
- Update or replace libraries.
- Reduce the technical debt.



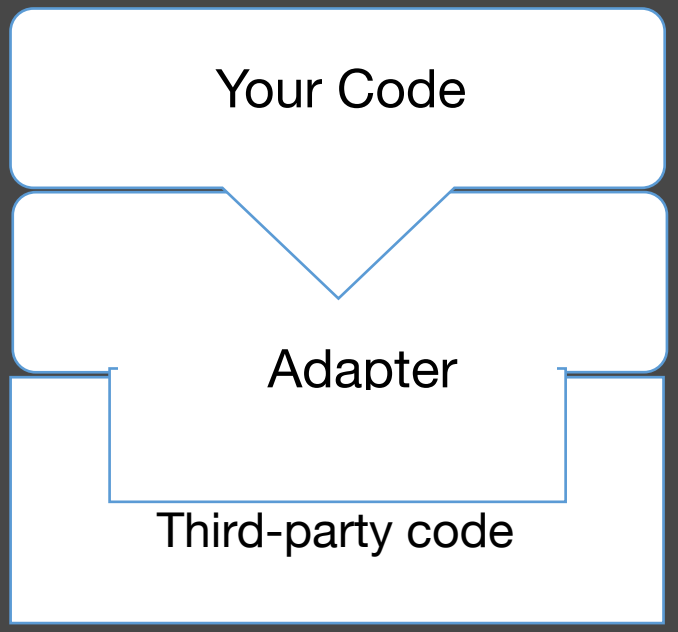
Example 2: Implement a Payment Gateway

Scenario:

- Vendor APIs - like payment gateways
- Can have more than payment gateway one in application
- Require to be inter-changed

Adapter Design Pattern

- Converts from provided interface to the required interface.
- A single Adapter interface can work with many Adaptees.
- Easy to maintain.

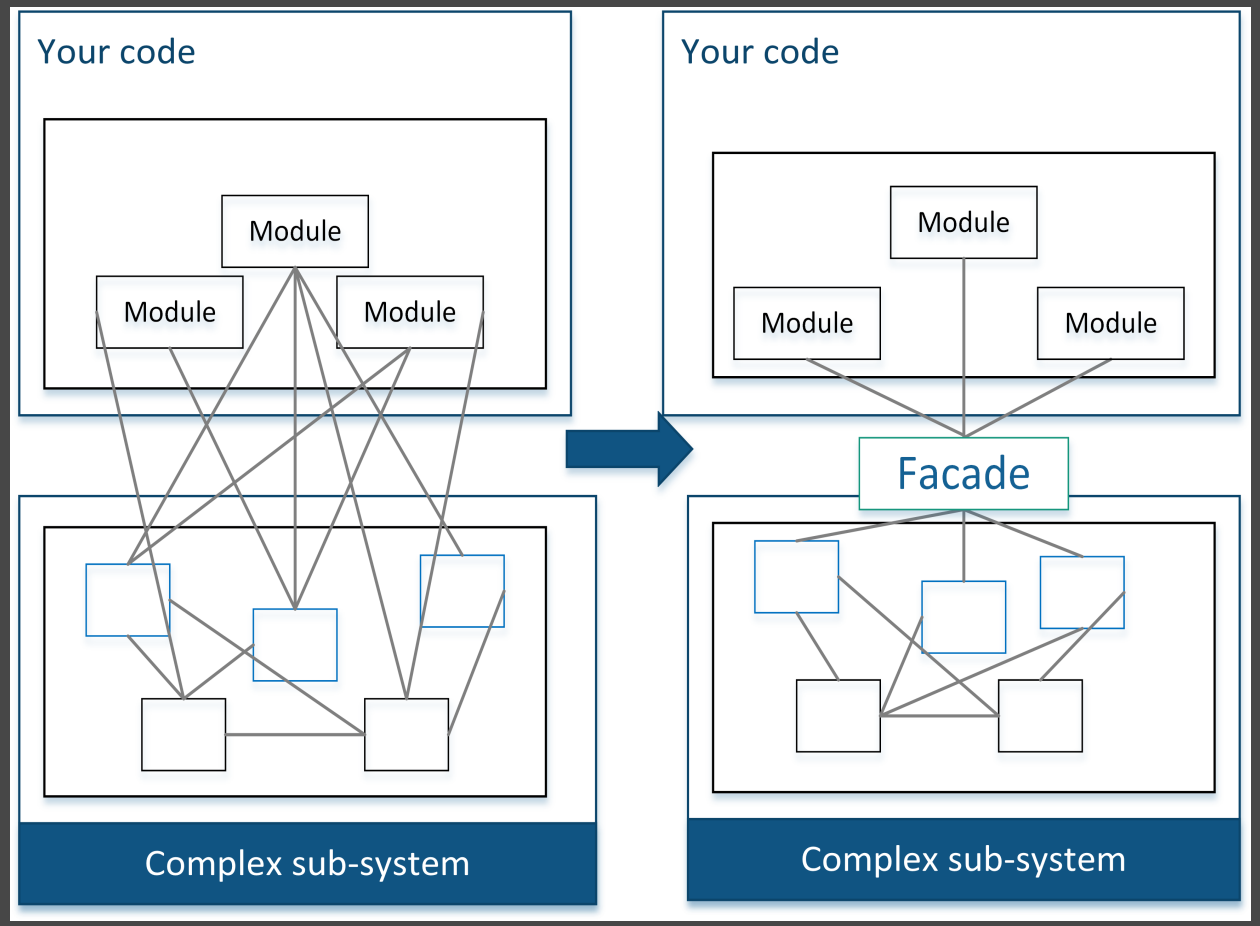


Example 3: Implement a Single Sign-On

- Libraries / packages created by another team within same company
- Re-used by multiple applications
- Common practice in large companies

Façade Design Pattern

- Simplifies the interaction with a complex sub-system
- Make easier to use a poorly designed API
- It can hide away the details from the client.
- Reduces dependencies on the outside code.

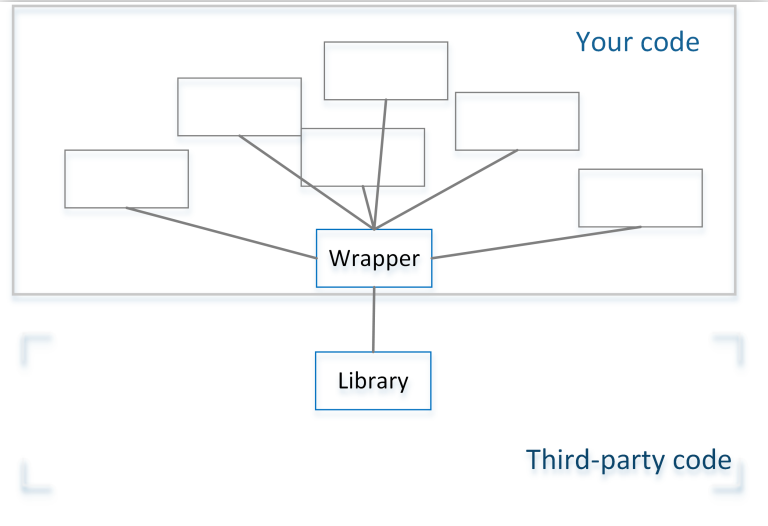


Secure Software Starts from Design !

Secure Software Starts from Design !

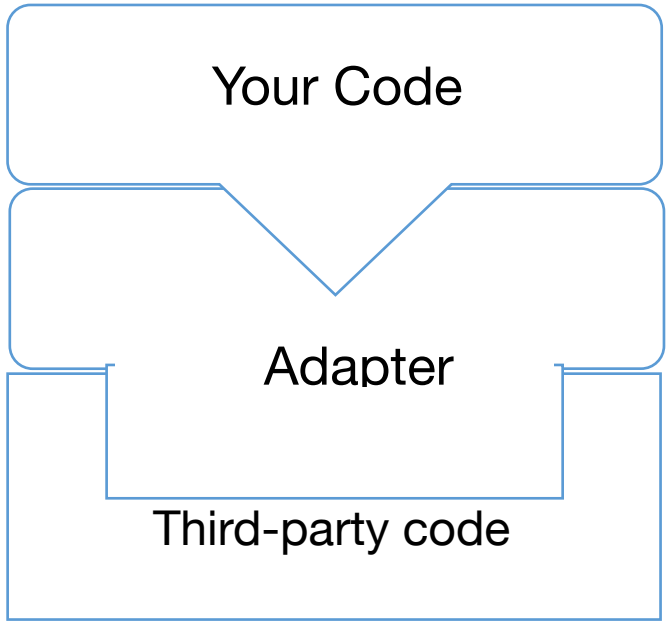
Wrapper

To expose only required functionality and hide unwanted behaviour.



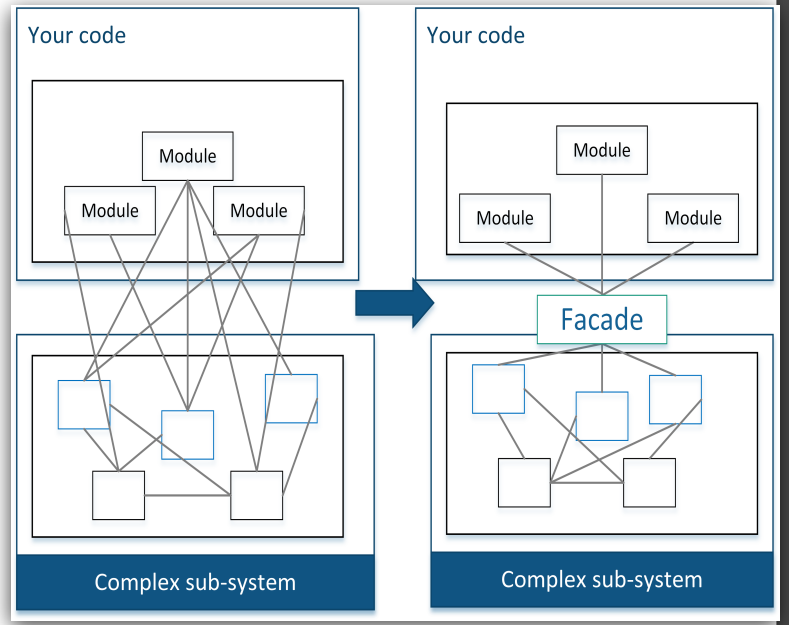
Adapter Pattern

To convert from the required interface to provided interface



Façade Pattern

To simplify the interaction with a complex sub-system.



How often ?

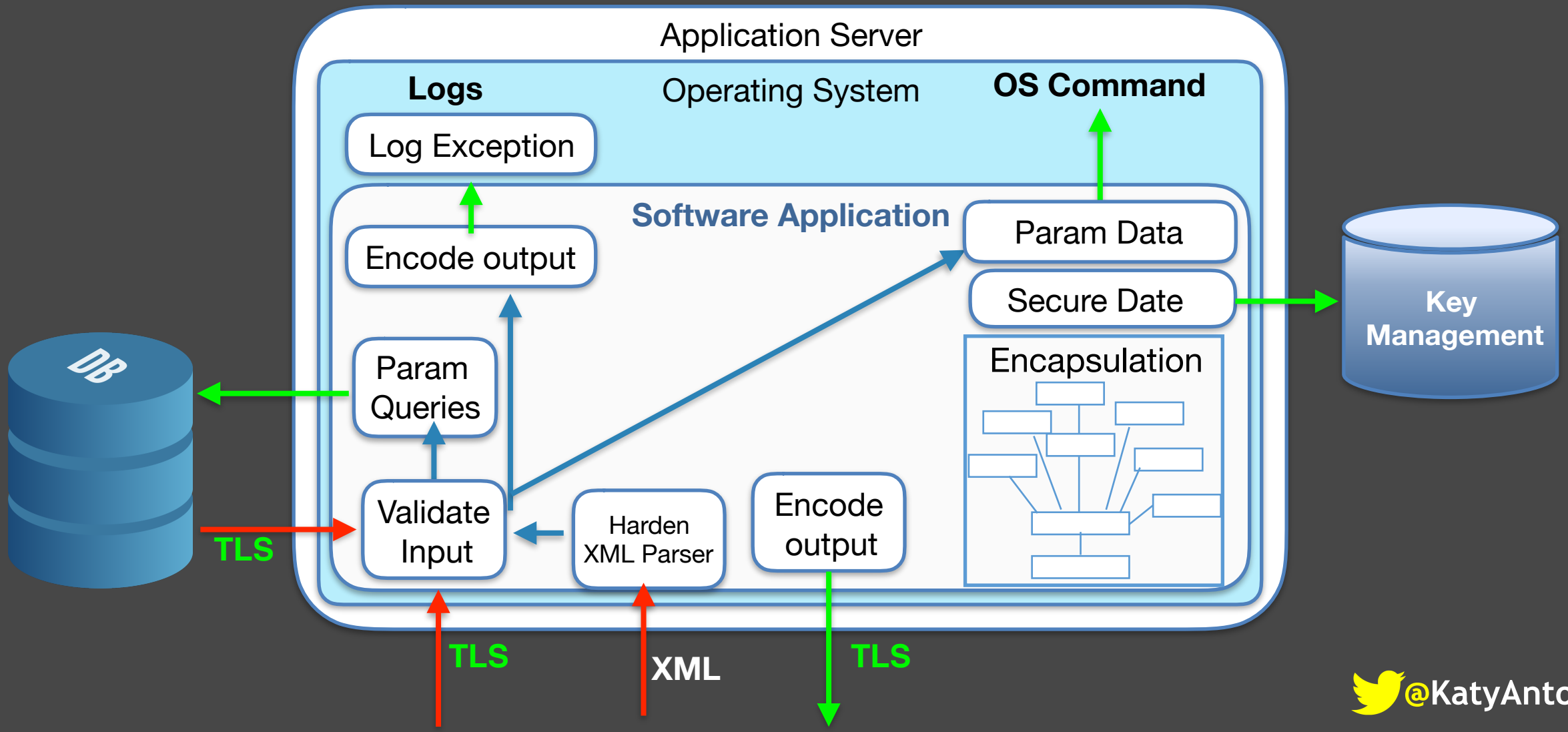
Rick Rescorla

- United States Army office of British origin
- Born in Hayle, Cornwall, UK
- Director of Security for Morgan Stanley at WTC

@KatyAnton

Security Controls Recap

Security Controls In Development Cycle



@KatyAnton

Final Takeaways

Focus on Security Controls

CWEs

Final Takeaways

Focus on Security Controls

CWEs

Thank you very much

Katy Anton
Principal Application Security Consultant