OWASP
AppSec Europe
London 2nd-6th July 2018

Injecting Security Controls into Software Applications

Katy Anton

About me

@KatyAnton

- Software development background
- Principal Security Consultant - CA Technologies I Veracode
- OWASP Bristol Chapter Leader
- Project co-leader for OWASP Top 10 Proactive Controls (@OWASPControls)

Injection

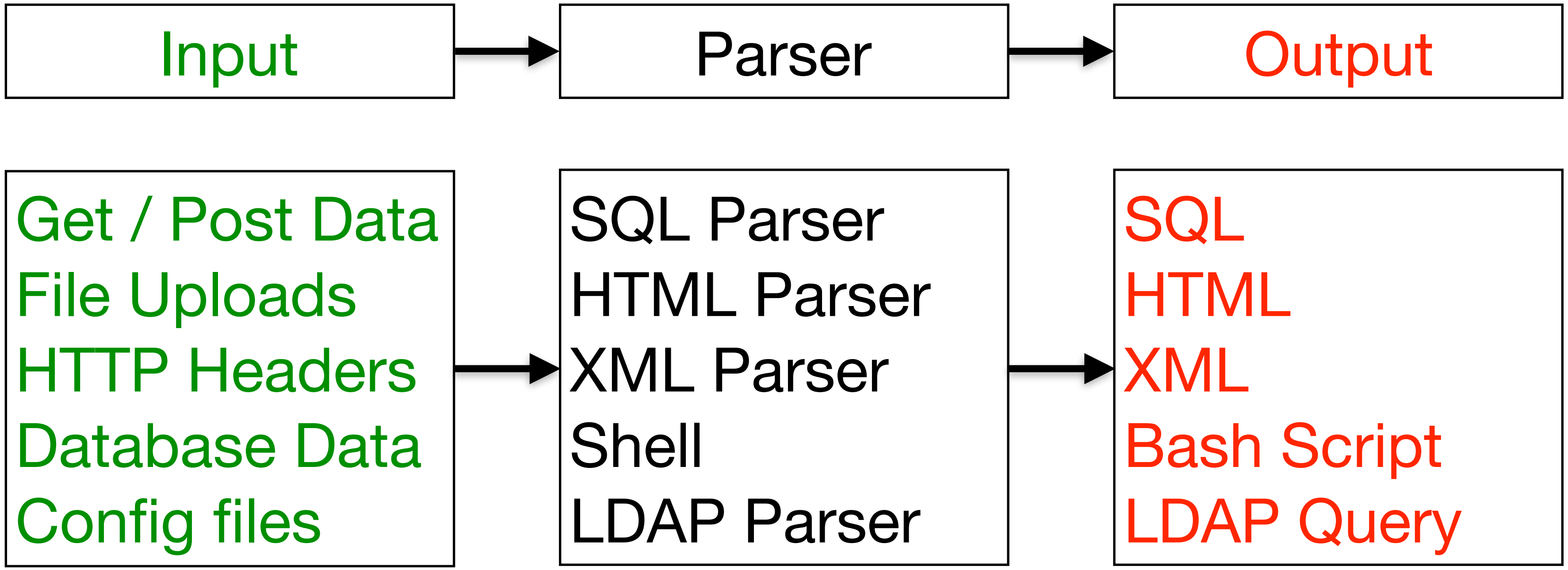
Injection

First mentioned in Phrack magazine in 1998
20 years anniversary

	2004	*2009*	*2010*	*2013*	*2017*
Injection	***A6***	***A2***	***A1***	***A1***	***A1***

Decompose the Injection

Data interpreted as Code



Input → Parser → Output

Get / Post Data
File Uploads
HTTP Headers
Database Data
Config files

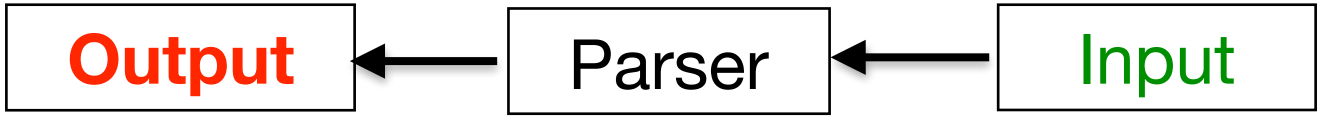
→

SQL Parser
HTML Parser
XML Parser
Shell
LDAP Parser

→

SQL
HTML
XML
Bash Script
LDAP Query

Extract Security Controls



Vulnerability	Encode Output	Parameterize	Validate Input
SQL Injection		☑	☑
XSS	☑		☑
XML Injection (XPATH Injection)	☑		☑
OS Cmd Injection	☑	☑	☑
LDAP Injection	☑		☑

Primary Controls

Defence in depth

Sensitive Date Exposure

Data at Rest and in Transit

Vulnerabilities

Data Types	Encryption	Hashing
Data at Rest Require initial value E.q: credit card	☑	
Data at rest Don’t require initial value E.q: user passwords		☑
Data in transit	☑	

Data at Rest: Vulnerabilities

How Not to Do it !

In the same folder - 2 file:

```
encrypted-password.txt
password-entities.txt
```

The content of password.txt:

```
cryptography.seed=abcd
cryptography.salt=12345
cryptography.iterations=1000
```

encryption_key = PBKF2(password, salt, iterations, key_length);

Security Controls

Cryptographic Storage

Strong Encryption Algorithm

- AES

Key Management

- Store unencrypted keys away from the encrypted data.
- Protect keys in a Key Vault (Hashicorp Vault / Amazon KMS)
- Keep away from home grown key management solutions.
- Define a key lifecycle.
- Build support for changing algorithms and keys when needed
- Document procedures for managing keys through the lifecycle

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

Security Controls

Password Storage - Use a Strong Algorithm

- PBKDF2
- bcrypt
- scrypt
- Argon2i
  - Java
  - PHP - password_hash() supports Argon2i from version 7.2

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

Security Controls

Data in Transit

- Client —> Application server
- Server—> Non-browser components

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. “

Security Controls

Security Logging

- Security logging: The security control that developers can use to log security information during the runtime operation of an application.

Logging implementation

- Logging framework : SLF4J with Logback or Apache Log4j2.
- Use a standard logging approach to facilitate correlation and analysis.

The 6 Best Detection Point Types

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

Intrusion Detection Points Examples

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)

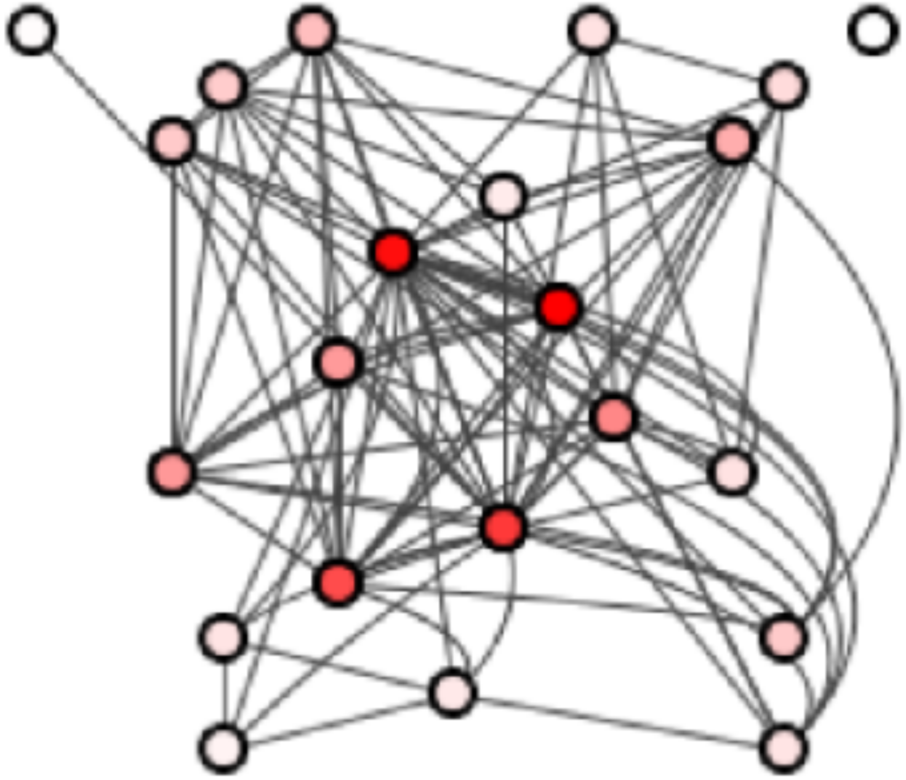
Source: https://www.owasp.org/index.php/AppSensor_DetectionPoints

Vulnerable Components

Using Software Components with Known Vulnerabilities

Root Cause

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



Third Party Components Age

“45% of the third-party components are over 4 years old”

Source: Synopsys - State of Software Composition 2017

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

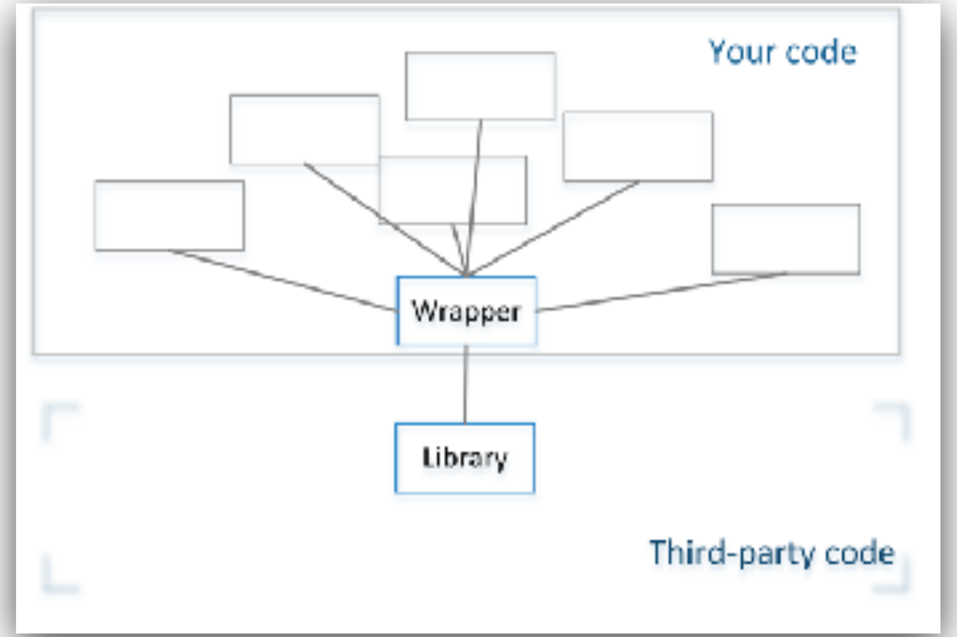
- Third-party - provides logging levels:
- FATAL, ERROR, WARN, INFO, DEBUG.

- 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.



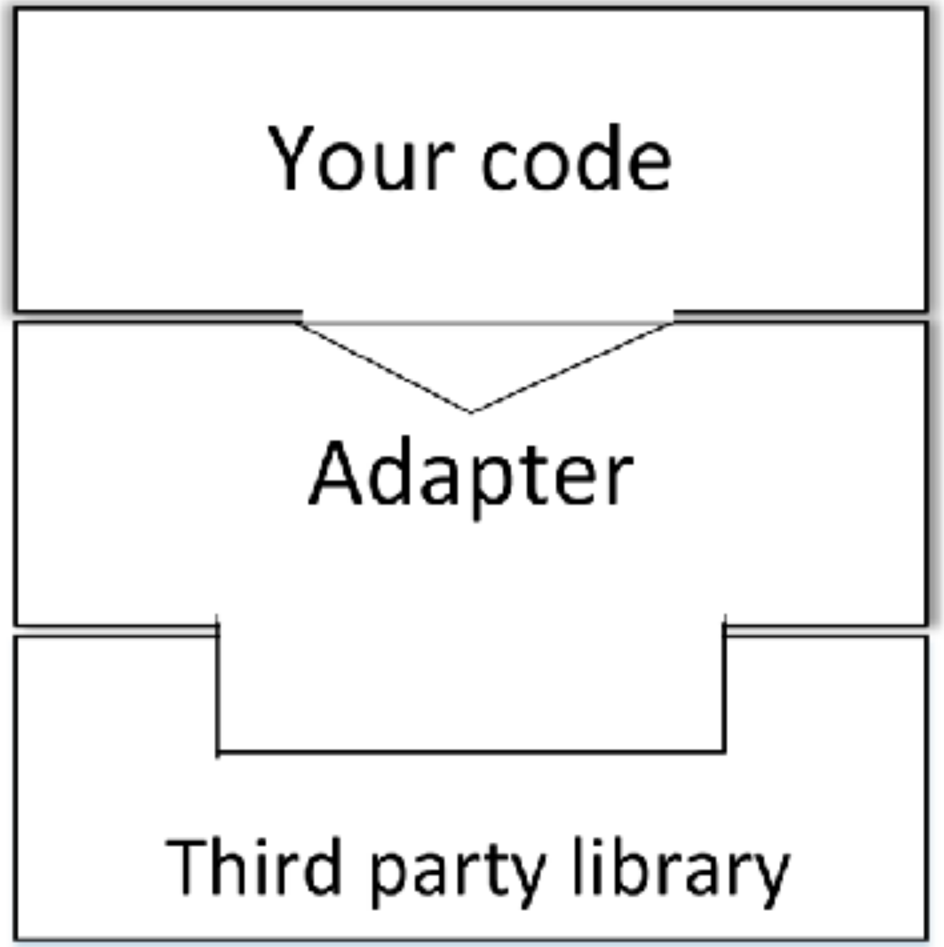
Example 2: Implement a payment gateway

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.

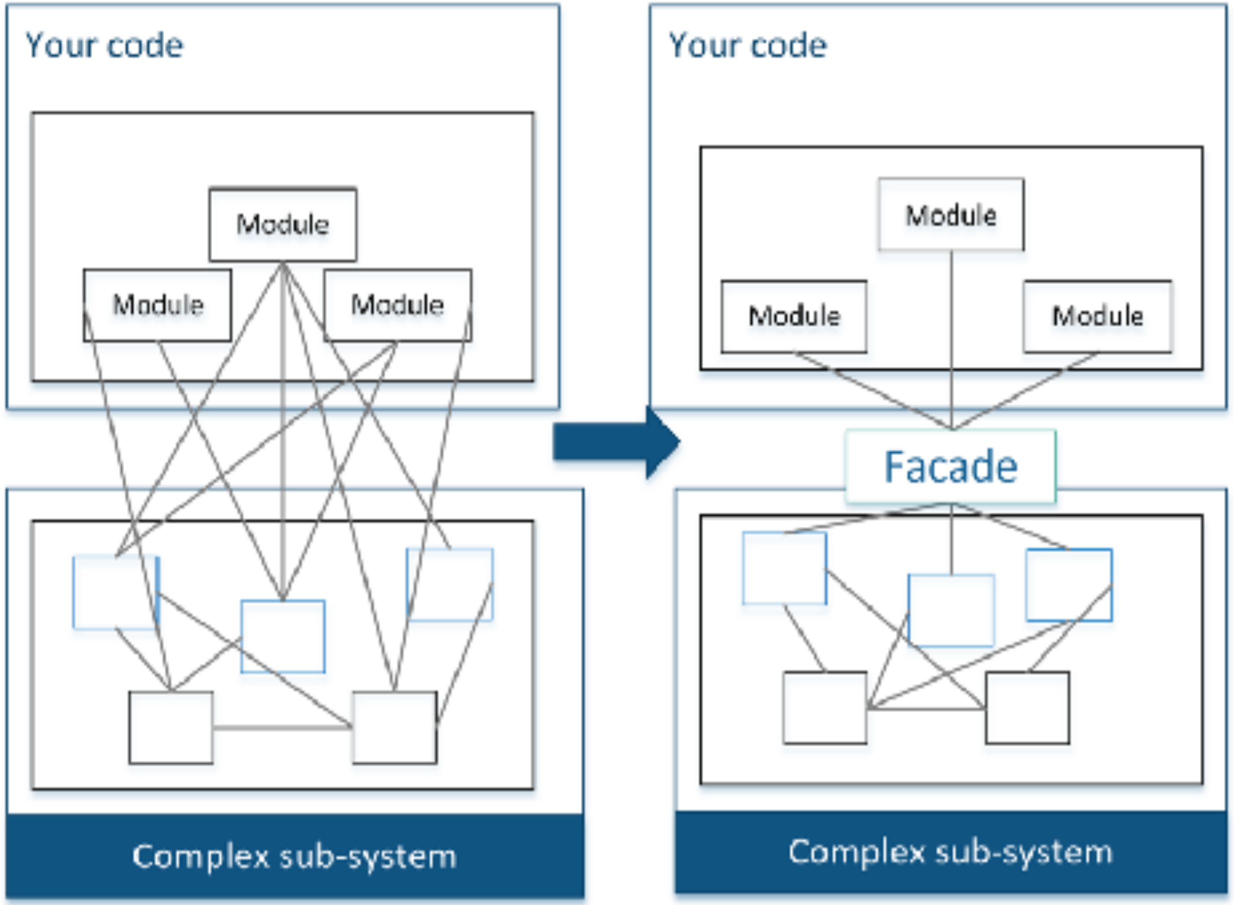


Example 3: Implement a Single Sign-On

- 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.

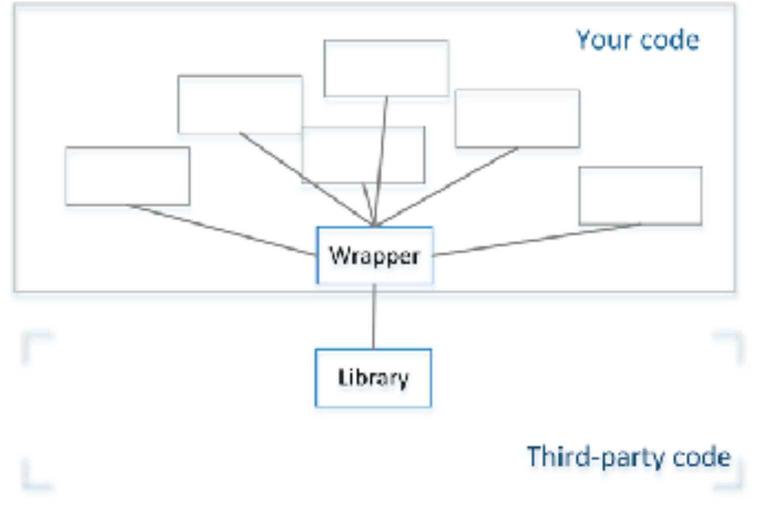


Secure Software Starts from Design !

@KatyAnton

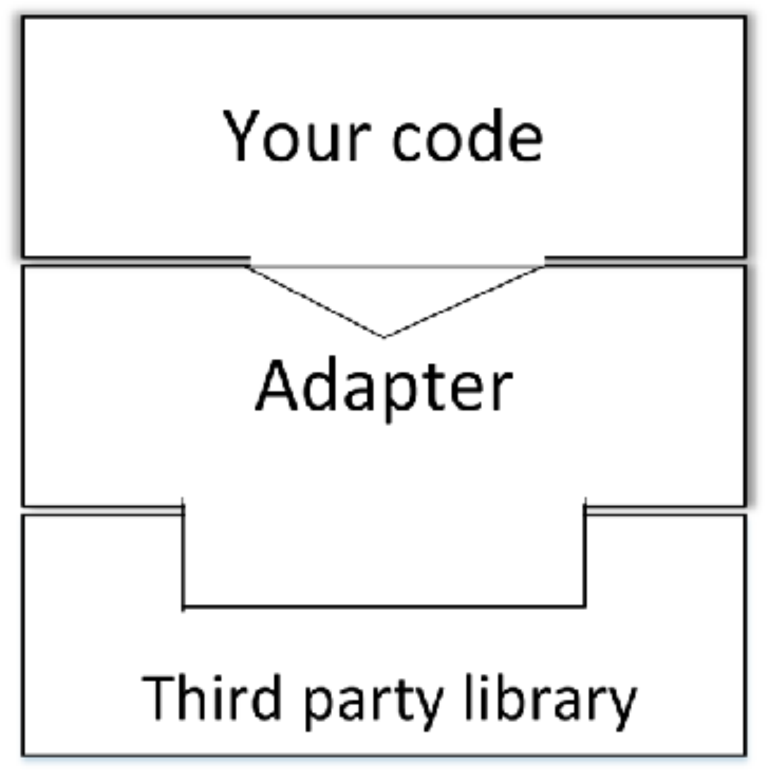
Wrapper

To expose only required functionality and hide unwanted behaviour.



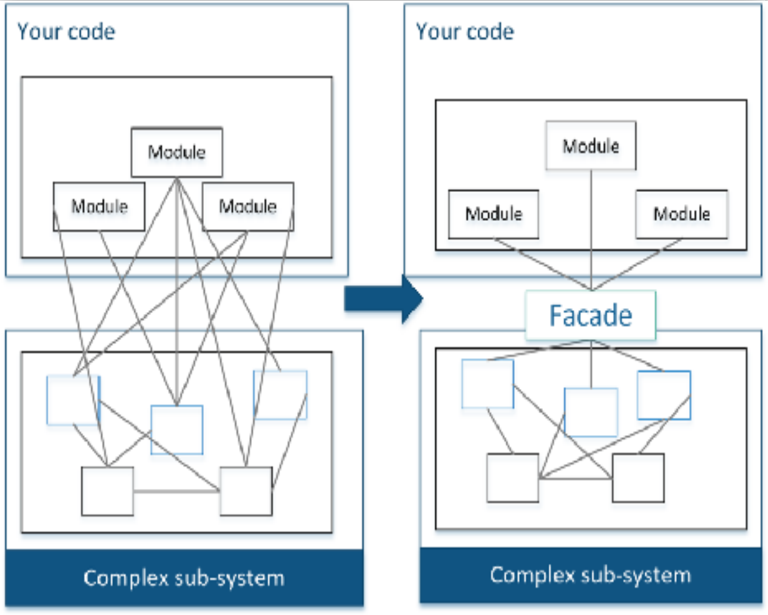
Adapter Pattern

To convert from the required interface to provided interface



Façade Pattern

To simplify the interaction with a complex sub-system.



How often?

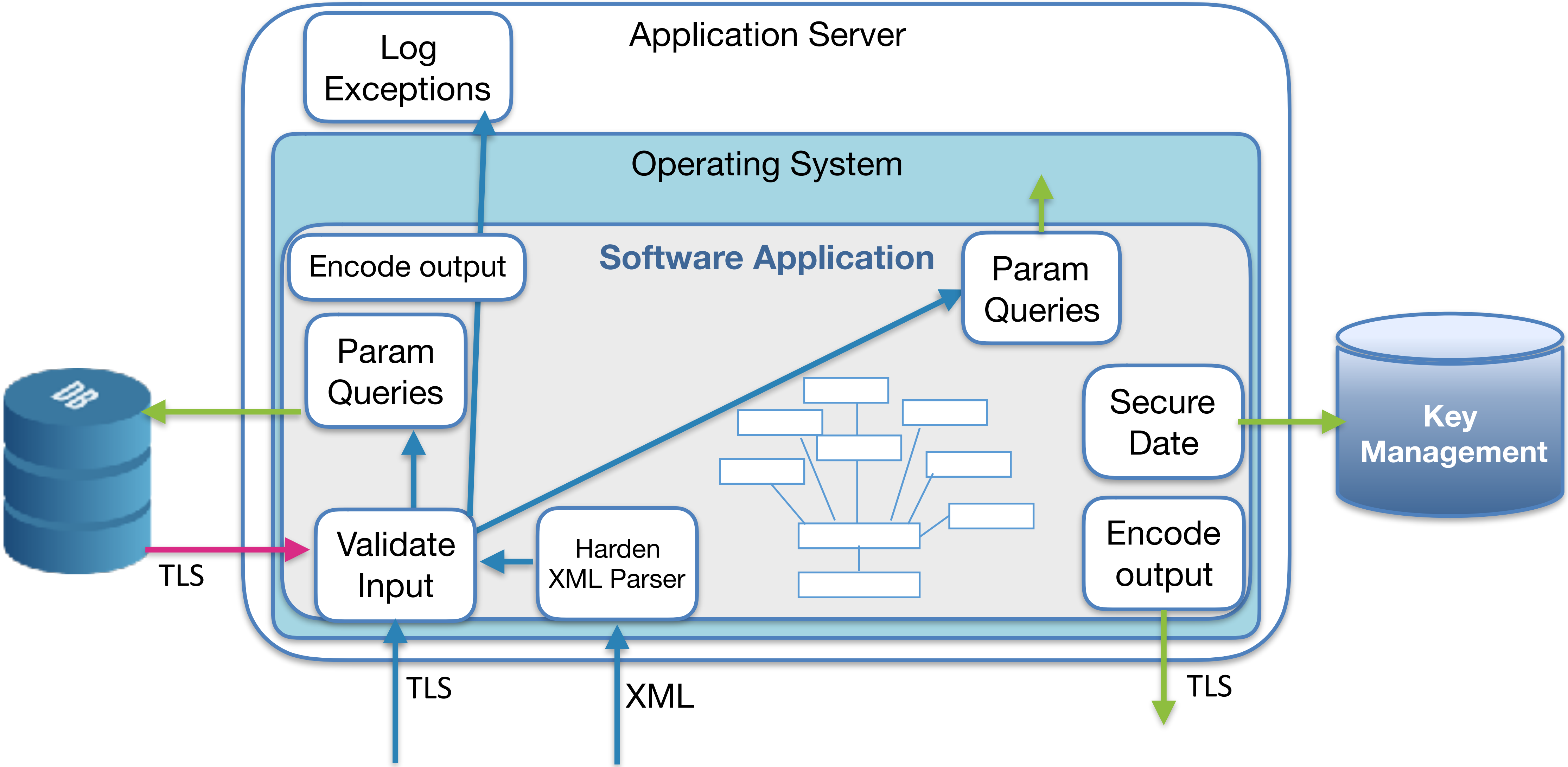
Rick Rescorla

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

Security Controls Recap

Security Controls Recap

@KatyAnton



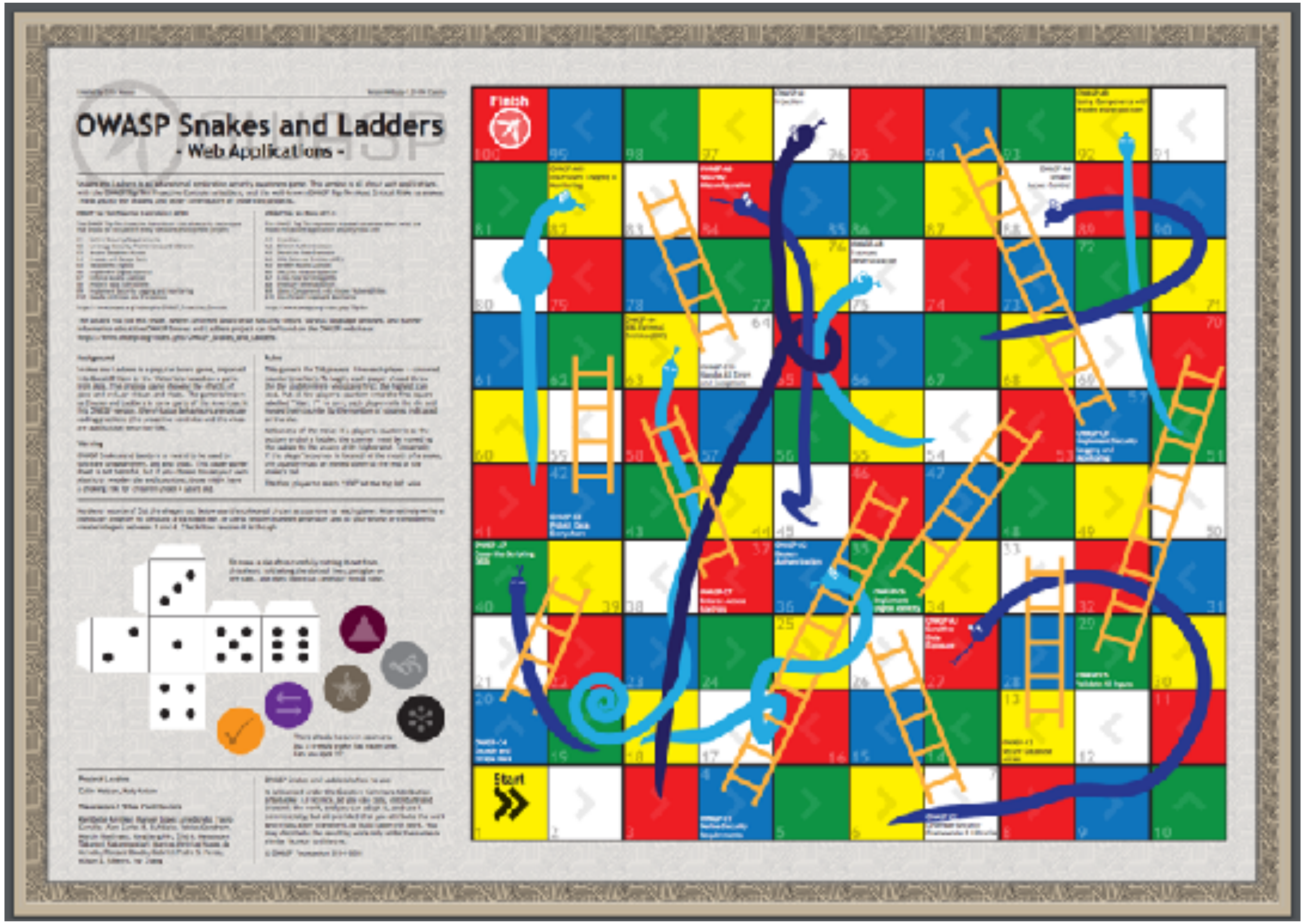
Get the Basics Right

"Most cyber threats are not that sophisticate … actors will use simple tools and techniques if they work.

Implementing basic cyber security practices remains the best way to tackle the majority of cyber threats."

Source: Director of GCHQ
CyberUK18

OWASP Snakes and Ladders



OWASP Snakes and Ladders
- Web Applications -

https://www.owasp.org/images/0/08/OWASP-SnakesAndLadders-WebApplications-EN.pdf

Thank you very much

@KatyAnton