

# SECURITY

**One API at a Time**

# WHOAMI

## ↳ Current

- ↳ Penetration Tester
- ↳ Security Consultant
- ↳ Lecturer

## ↳ Experience

- ↳ 2 years Software Developer
- ↳ >8 years Linux System Engineer
- ↳ 1½ years Information Security Management

# WHAT THIS TALK IS NOT

- ↳ Browser-focused
- ↳ Complete
- ↳ Applicable for every app without modification

WHY BOTHER AT ALL?

Amazon Suffers Security Breach; 80,000 Login Credentials Leaked (Updated)

**Office 365, Azure users are locked out after a global multi-factor authentication outage**

**A leaky database of SMS text messages exposed password resets and two-factor codes**

**Gmail Bugs Allow Changing From: Field and Spoofing Recipient's Address**

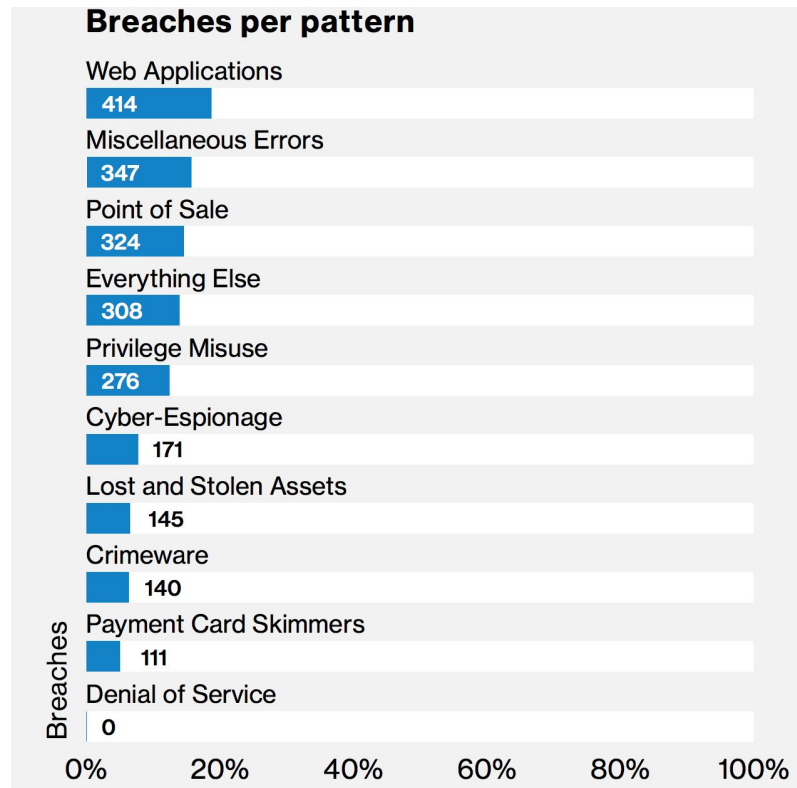
DJI Drone Vulnerability

Research by: Oded Vanun, Dikla Barda and Roman Zaikin

**Security researchers have busted the encryption in several popular Crucial and Samsung SSDs**

- WebApplications account for ~18% (n=2,216) of breaches in 2017[6]
- 23.244 WebApplications compromised as a mean to attack something else[1]

[1][https://www.verizonenterprise.com/resources/reports/rp\\_DBIR\\_2018\\_Report\\_en\\_xg.pdf](https://www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_en_xg.pdf)

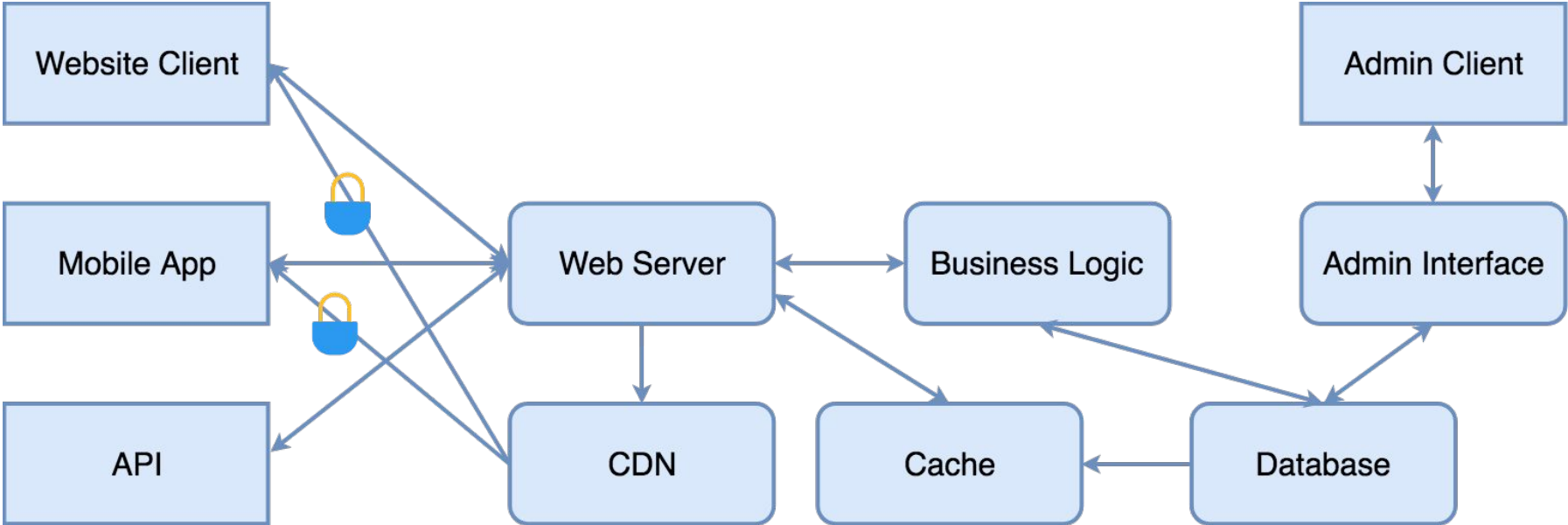


# FACTS

- ↳ Security is a team effort
- ↳ It can't be bought as an add-on
- ↳ It can't be patched on as an afterthought

# OUR DEMO APP





# THE BASICS


# TLS?


Secure HTTPS

 <https://www.google.com>

 <https://www.google.com>


HTTP

 [www.example.com](http://www.example.com)

 [www.example.com](http://www.example.com)

HTTPS with  
minor errors

 <https://mixed.badssl.com>

 <https://mixed.badssl.com>

Broken HTTPS

 <https://expired.badssl.com>

 <https://expired.badssl.com>

# HOW TO DEPLOY TLS CORRECTLY?

- ↳ <https://mozilla.github.io/server-side-tls/ssl-config-generator/>
- ↳ <https://bettercrypto.org>
- ↳ Don't have TLS at all?
  - ↳ <https://letsencrypt.org>

# AUTHENTICATION

- ↳ Basic Auth (you already use TLS. amirite?)
  - Use sessions!?
  - Without session-use:
    - Password-hashing with sensible workload slows your API down
- ↳ Token Based
  - oauth2
  - Certificates

↻ Scott Bollinger Retweeted



**Elliot Alderson** @fs0c131y · 6m

The first iOS app I reversed yesterday:

- Password is send in clear text
- Twitter api keys are stored in a plist file
- ...

And this is not even a crackme app



2

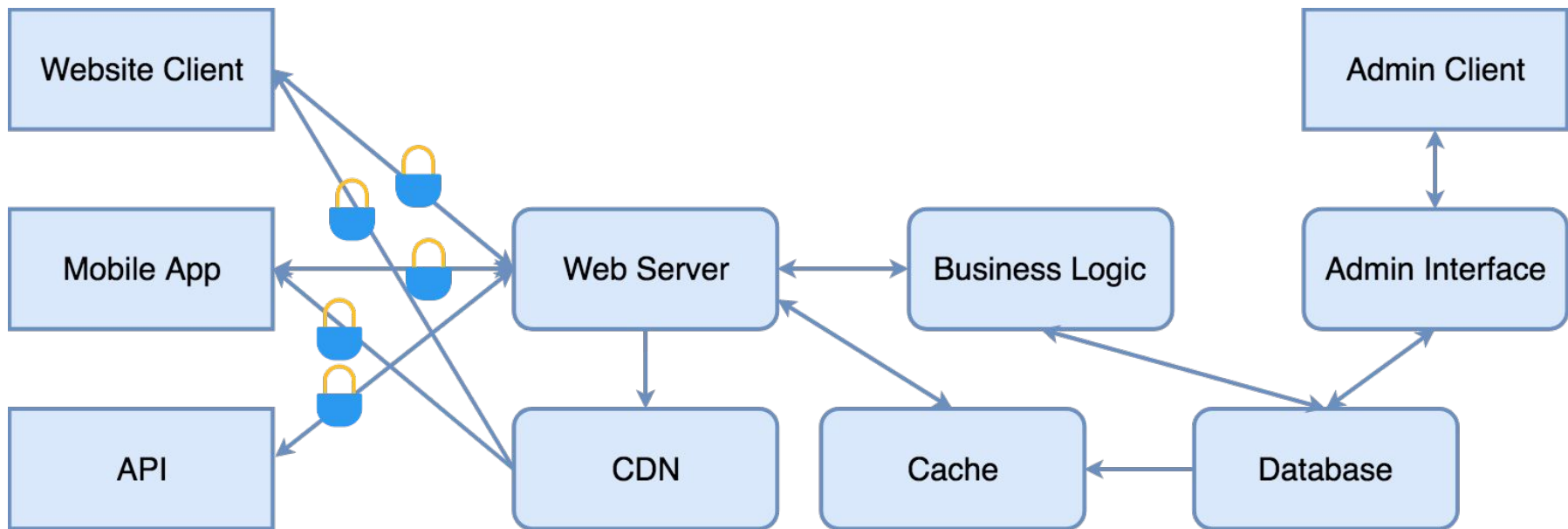


5



21





# HOW TO VALIDATE?

- nmap
- TLS scanners
  - <https://www.ssllabs.com>
  - BURP Suite
  - ...
- Integration Tests
  - You know, your auth methods have to work?



INTERMEDIATE

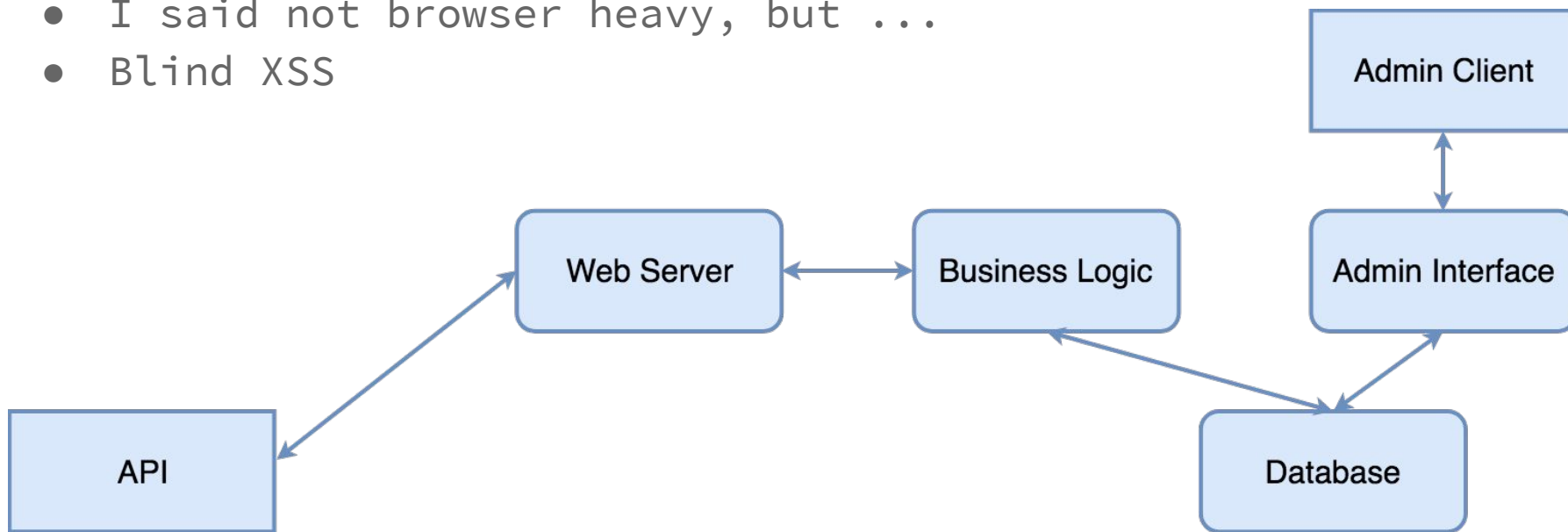
# INPUT VALIDATION

- Do not sanitise
  - Can still be exploited in a second step
- Know your input data
  - poor ol' Miles O'Brien might still want to login



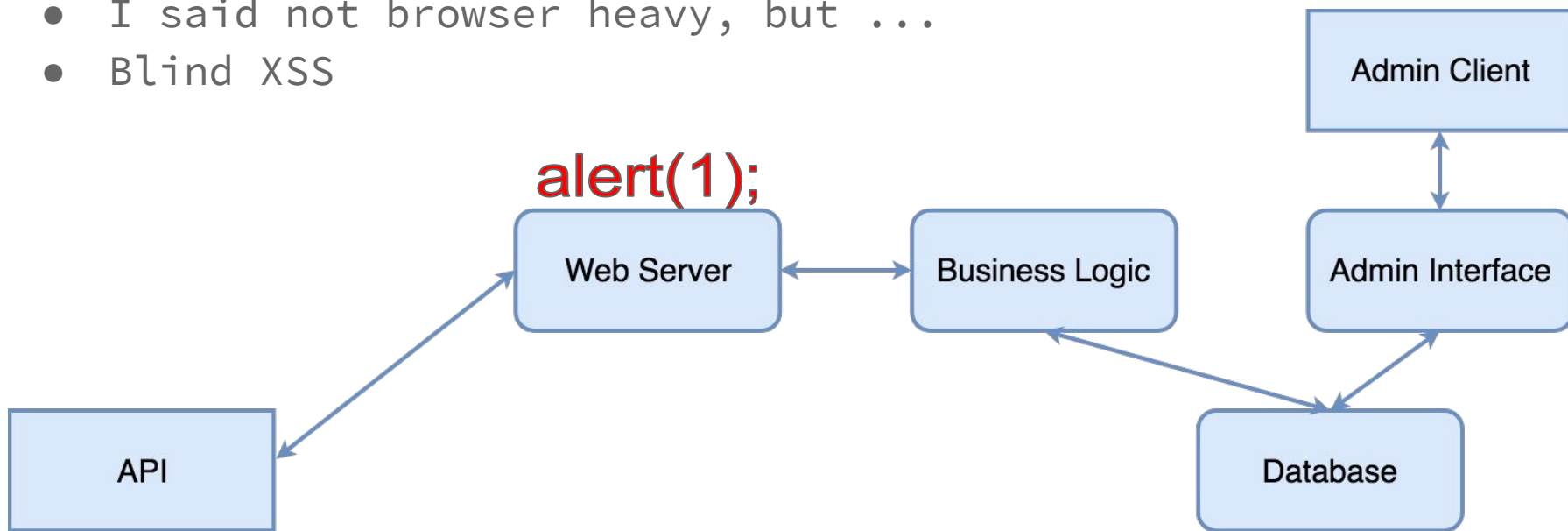
# XSS

- I said not browser heavy, but ...
- Blind XSS



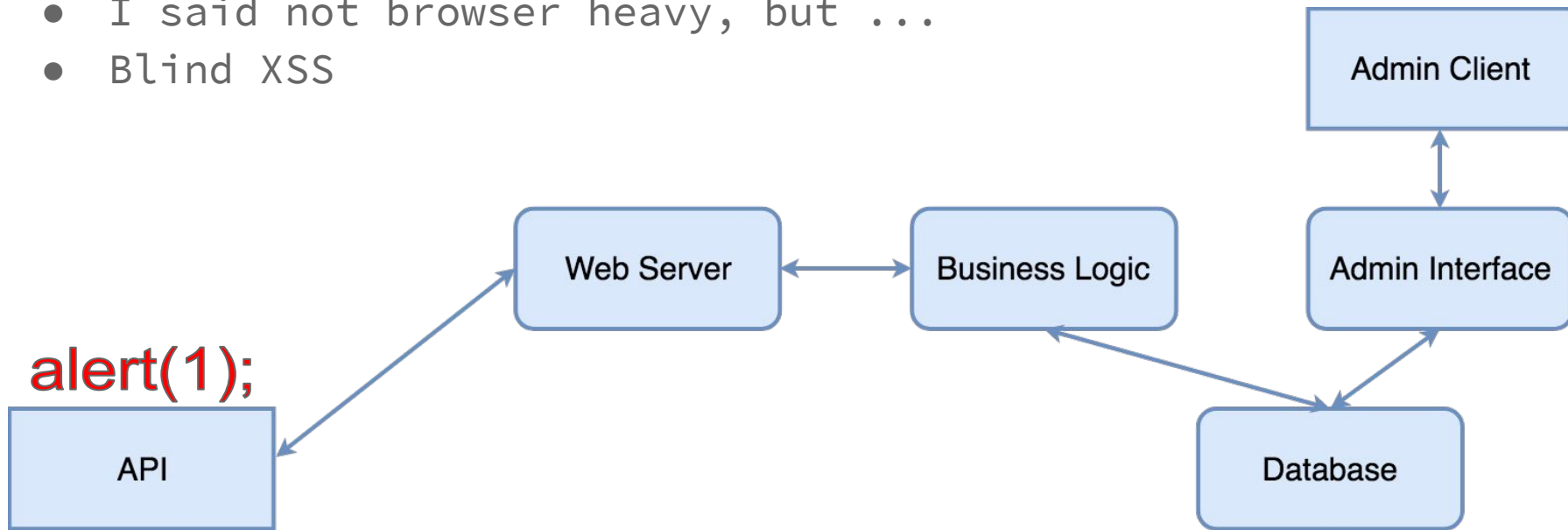
# XSS

- I said not browser heavy, but ...
- Blind XSS



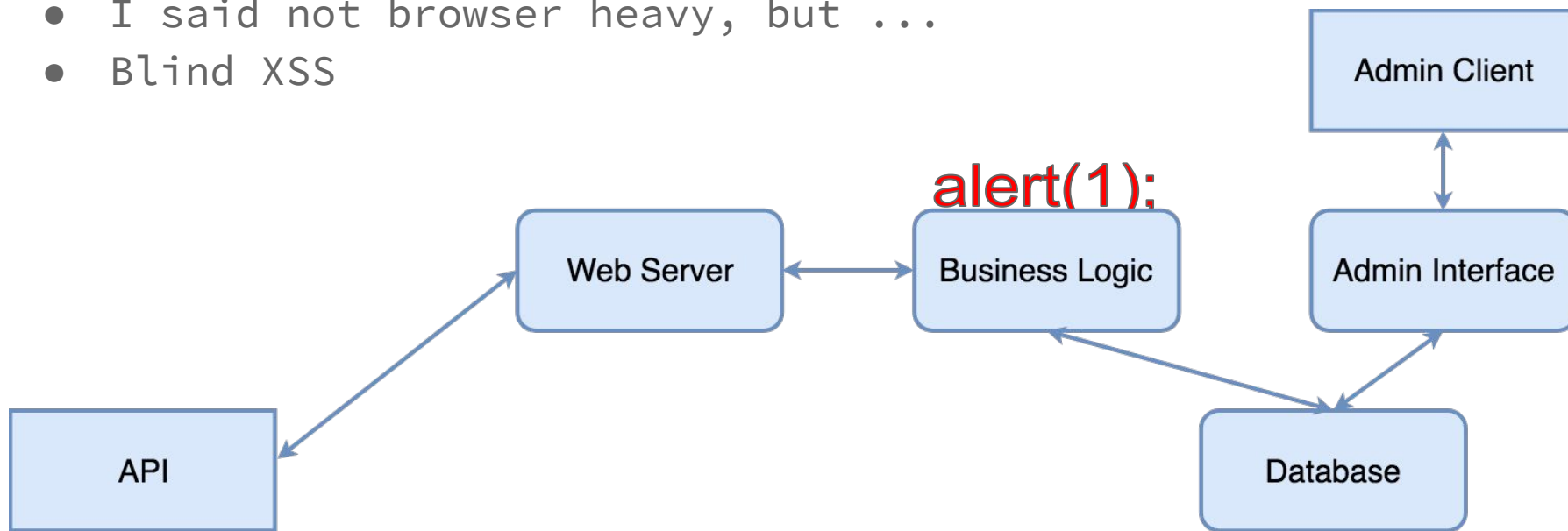
# XSS

- I said not browser heavy, but ...
- Blind XSS



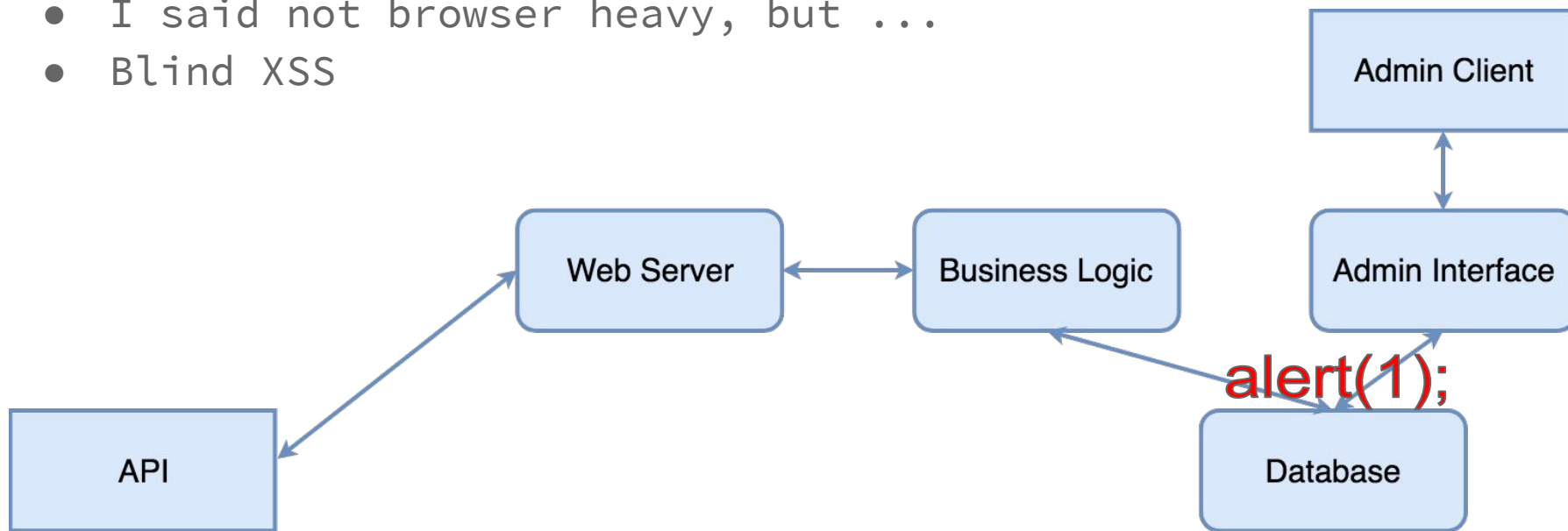
# XSS

- I said not browser heavy, but ...
- Blind XSS



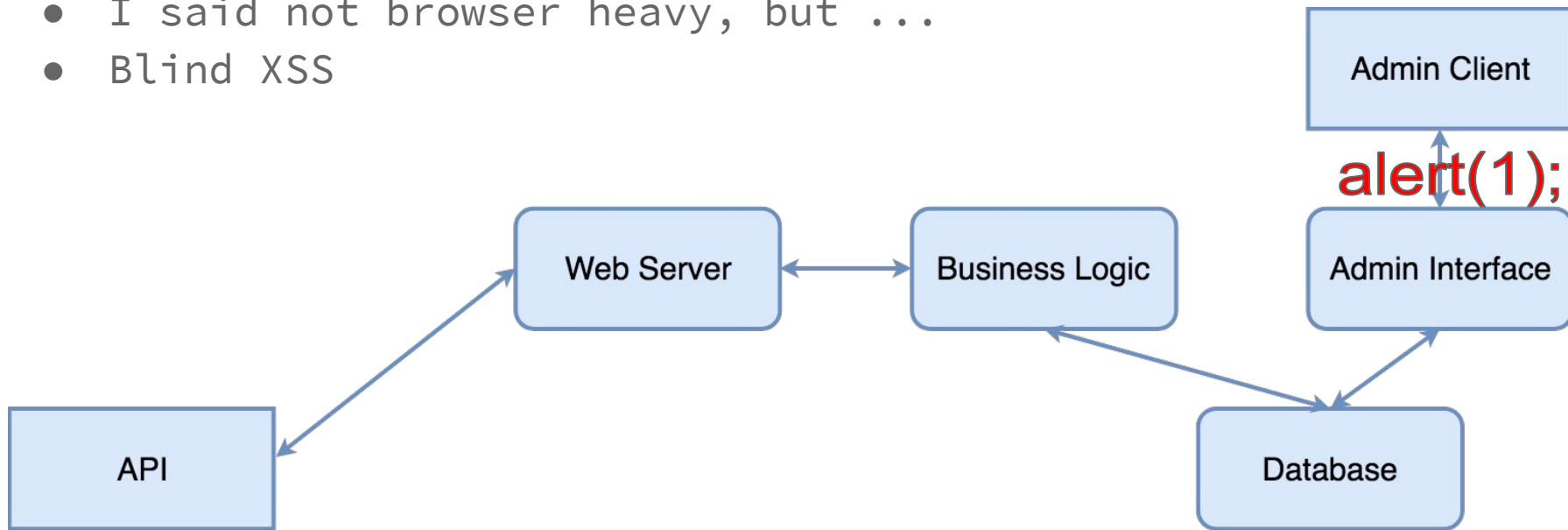
# XSS

- I said not browser heavy, but ...
- Blind XSS



# XSS

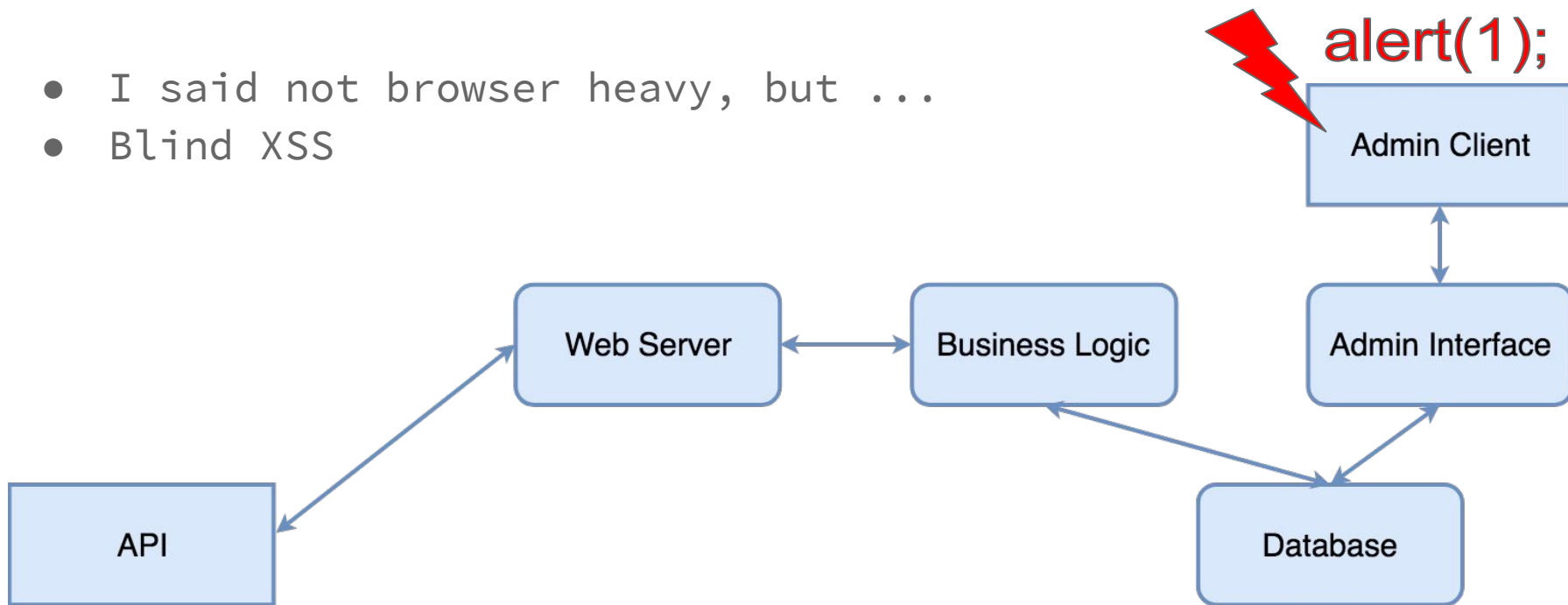
- I said not browser heavy, but ...
- Blind XSS





# XSS

- I said not browser heavy, but ...
- Blind XSS



# BUT ALSO

# LOGGING & MONITORING

- You already have
- Add security logging to the mix
  - Failed login attempts
  - Succeeded login attempts
  - Calls to privileged functions
  - ...
- Logging without monitoring is only forensic data
  - Helps only to reconstruct what happened

# HOW TO VALIDATE

- Vulnerability Scanning
  - OWASP ZAP
- Penetration Tests
  - Extern
  - Intern
- Code Reviews and Audits

# RECAP

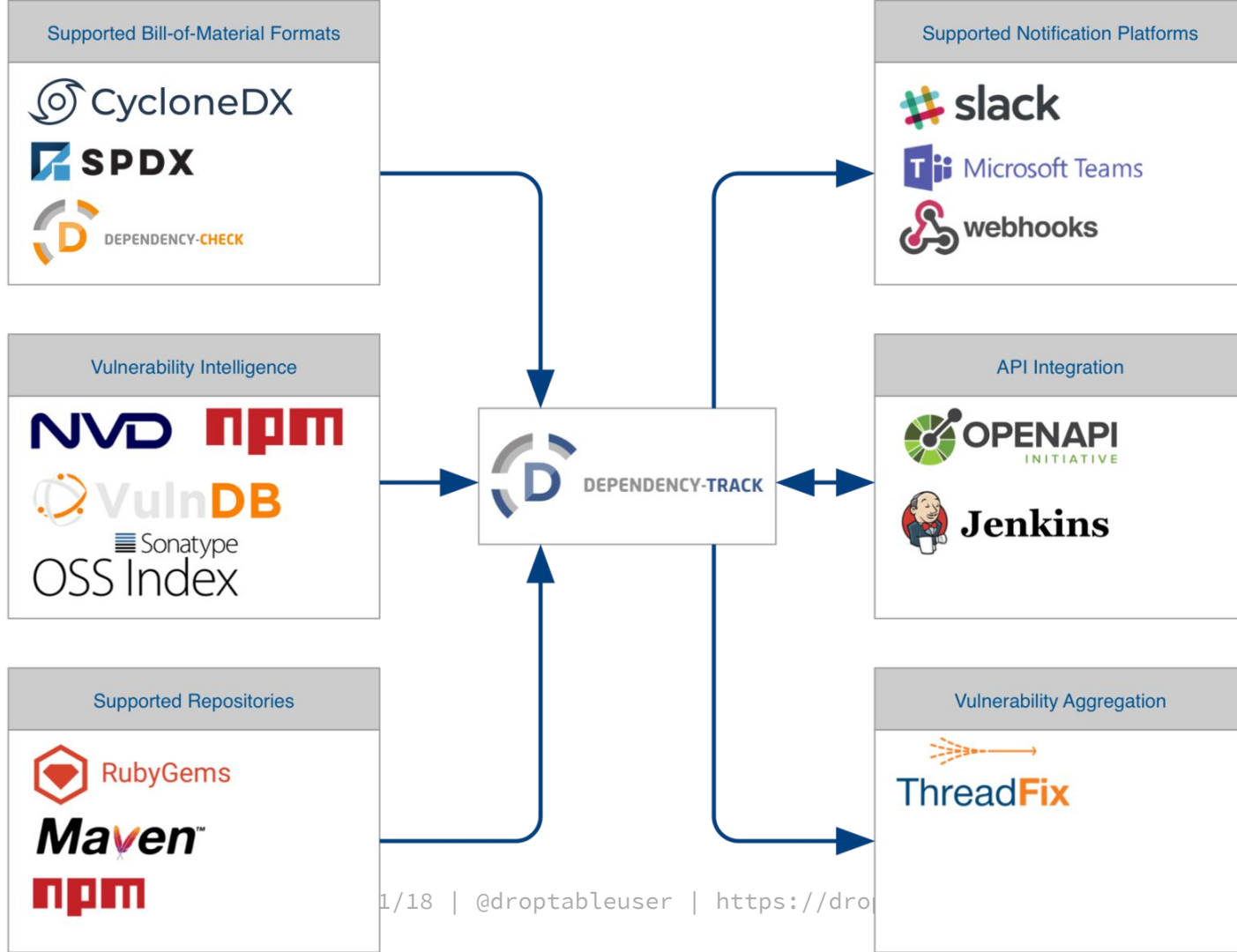
What did we already cover?

- ↳ HTTPS on our API
- ↳ Authentication
- ↳ Authorization
- ↳ Input Validation
- ↳ Logging for Security
- ↳ Monitoring

ADVANCED

# CI/CD FOR SECURITY

- Previously mentioned tests
  - Good tools are either integratable or deliver replicable calls
- Dependency Management



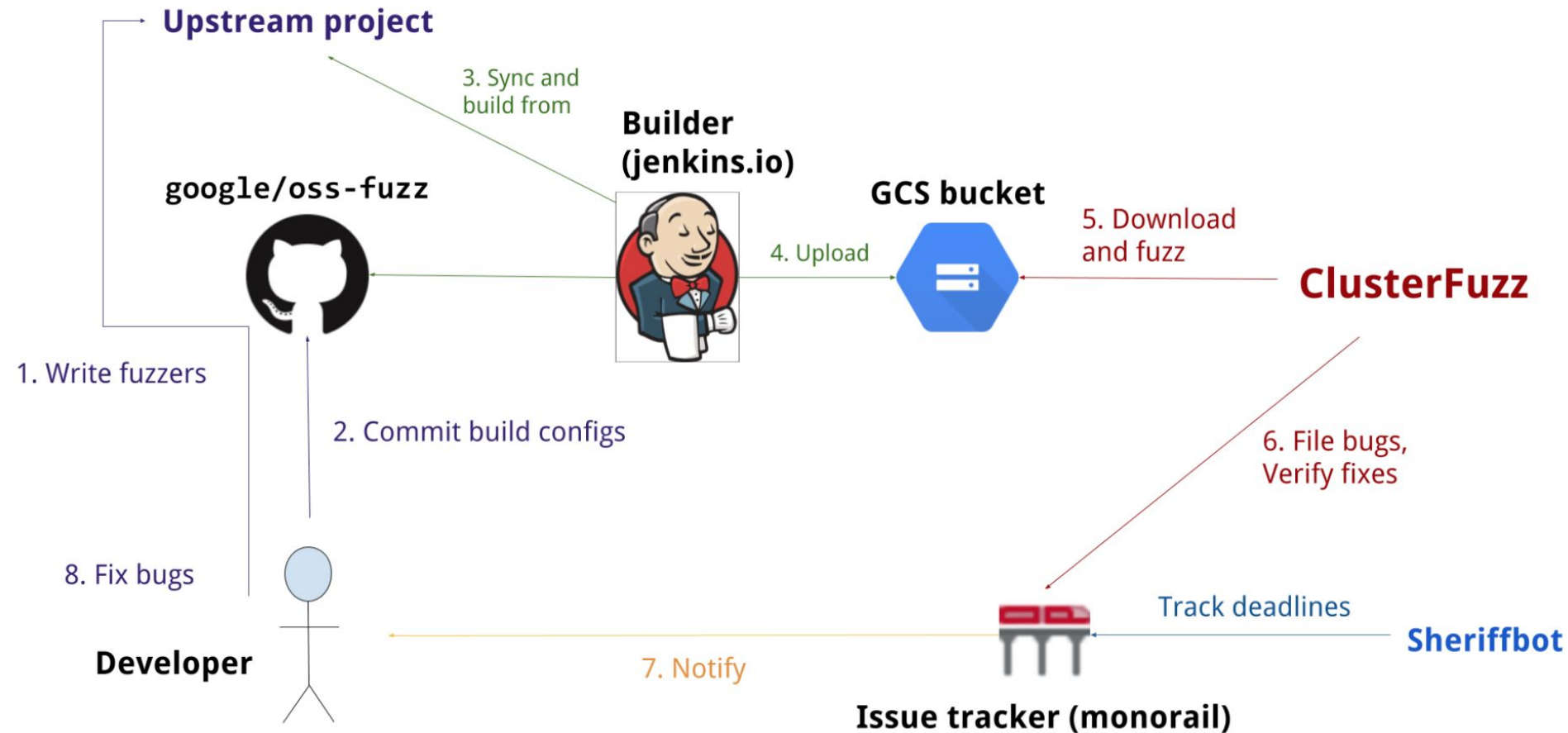


# CI/CD FOR SECURITY

- Previously mentioned tests
  - Good tools are either integratable or deliver replicable calls
  -
- Dependency management
- Security tests need to fail the build
- Load tests include security tests

# FUZZING

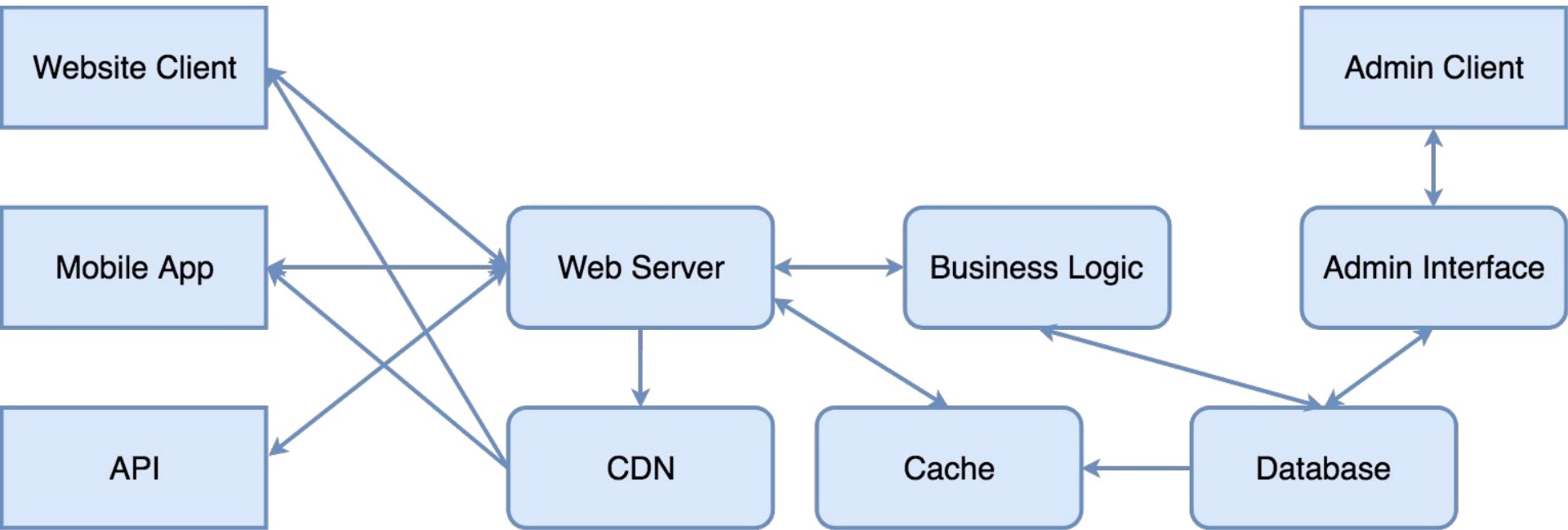
- Generating random input for test cases
- Complete code coverage
- Usually runs very long
- Fuzzing success means crash/error condition
  - This test case moves into regular testing setup



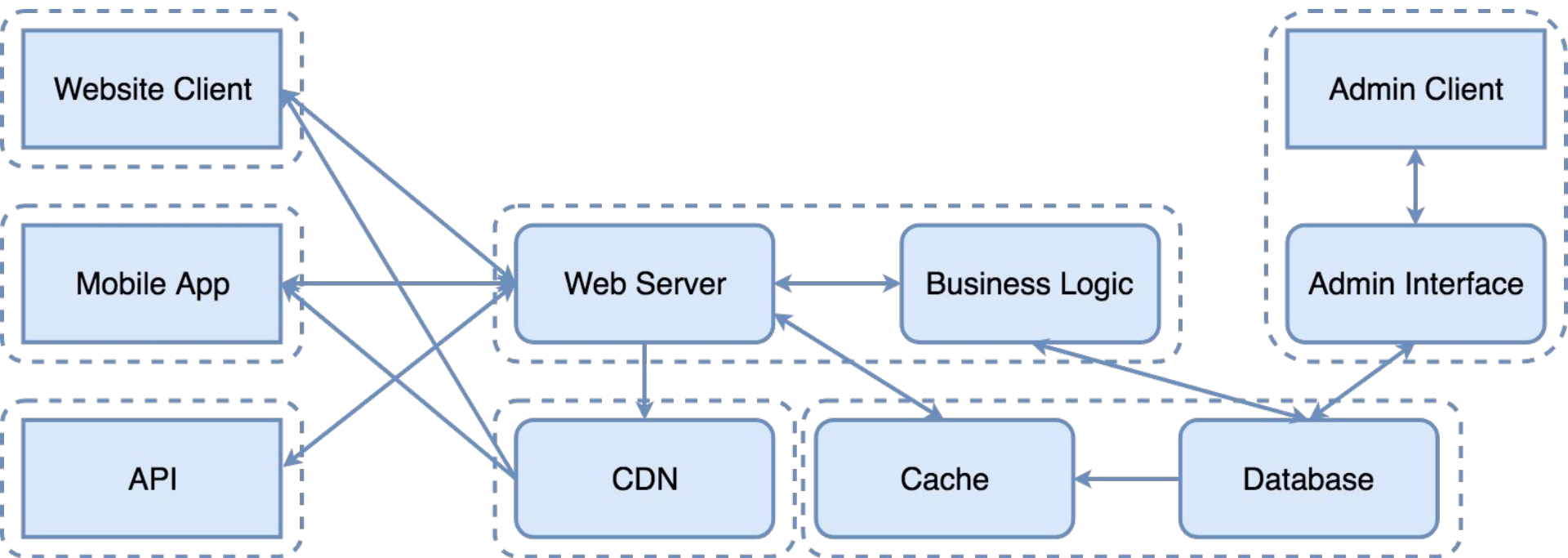
# THREAT MODELING

- Structured approach
- Trying to get the complete coverage for whole application
- How to get started?

# APPLICATION DESIGN REVIEW

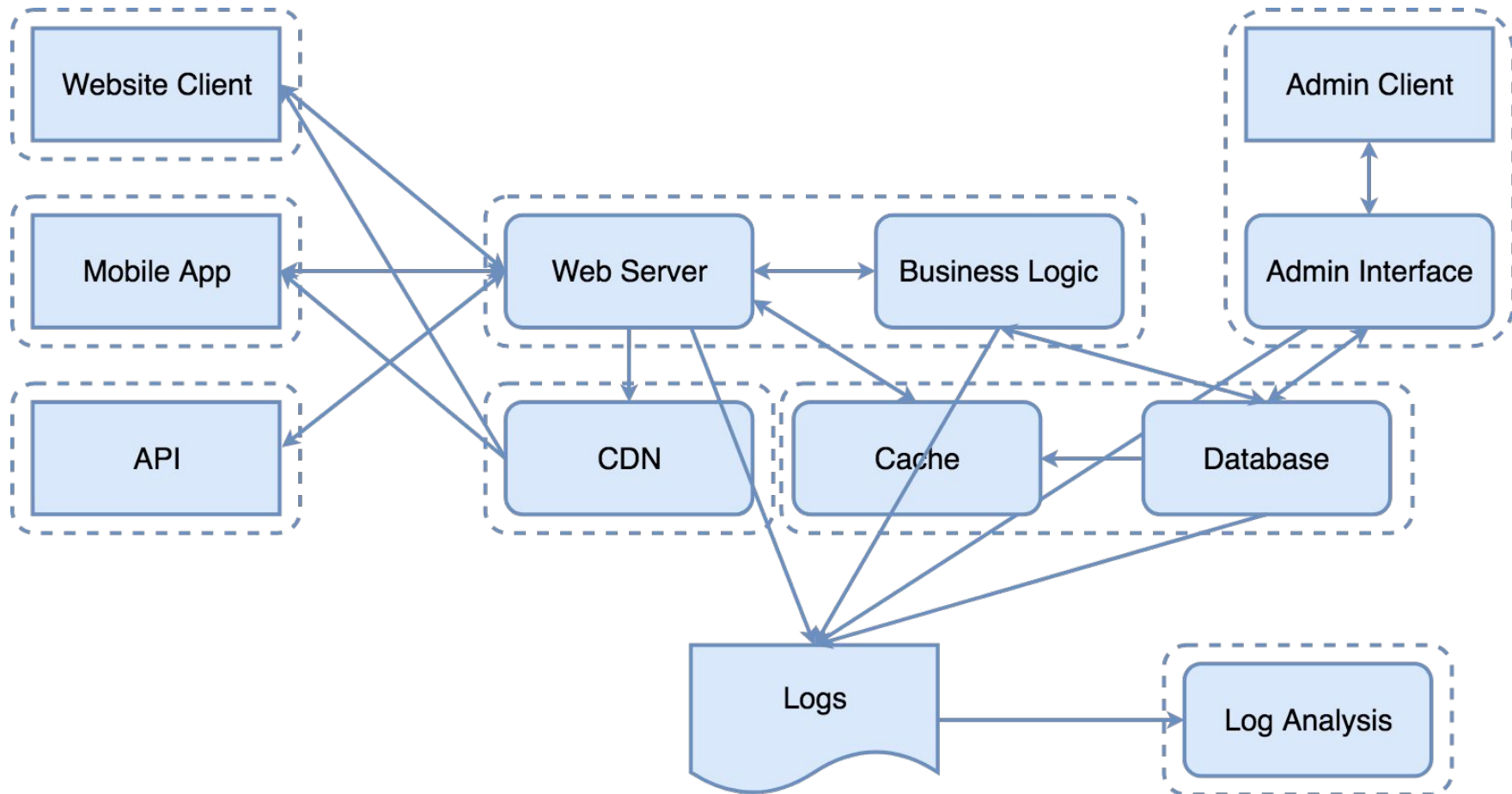


# ADD TRUST BOUNDARIES



# CHECK FOR COMPLETENESS

- Did we add things to the app?

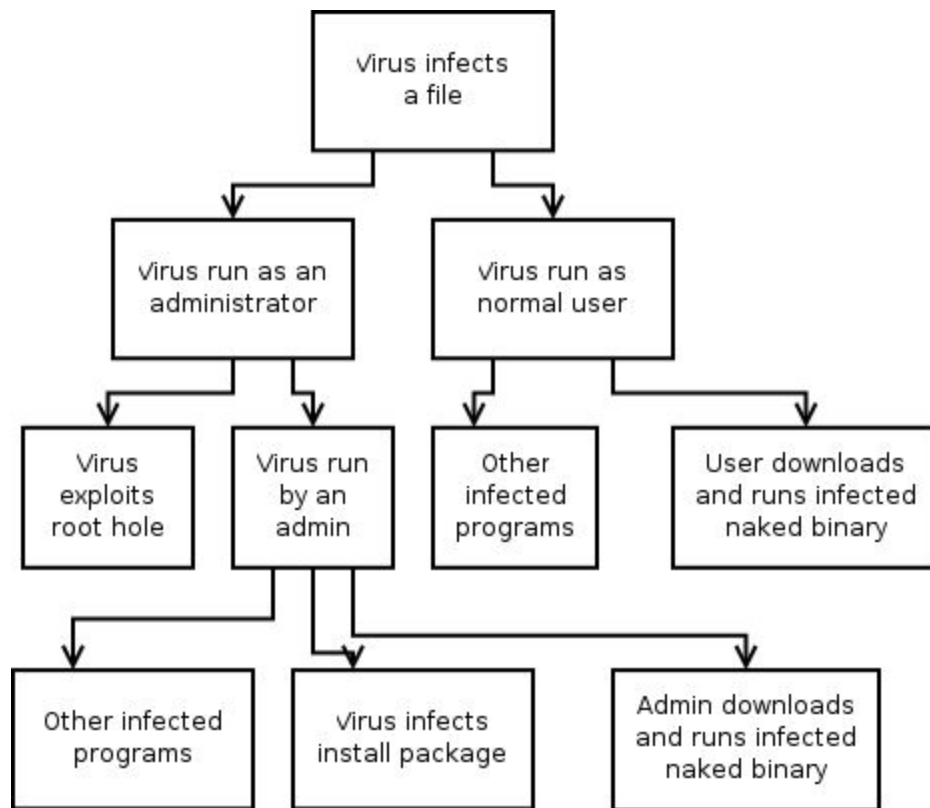




# THREAT MODELING

- STRIDE
  - Spoofing
  - Tampering
  - Repudiation
  - Information Disclosure
  - Denial of Service
  - Elevation of Privilege

# ATTACK TREES



# MITRE CAPEC

- Common Attack Pattern Enumeration and Classification
  - <https://capec.mitre.org/data/definitions/1000.html>

WHO IS GOING TO DO  
ALL OF THIS?

# A SECURITY ENGINEER

# YOU NEED:

- A dedicated person
  - Software Development Exp.
  - Security Exp.
- Time and Budget
- Included in your processes
- The sooner in your development process the better

# THANKS

## Any Questions?