Security

One API at a Time

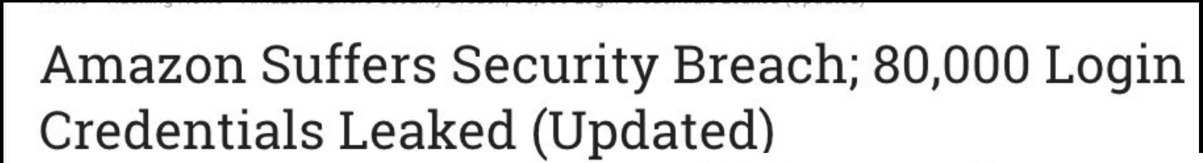
whoami

- ↳ Current
  - ↳ Penetration Tester
  - ↳ Security Consultant
  - ↳ Lecturer
- ↳ Experience
  - ↳ 2 years Software Developer
  - ↳ >8 years Linux System Engineer
  - ↳ 1½ years Information Security Management

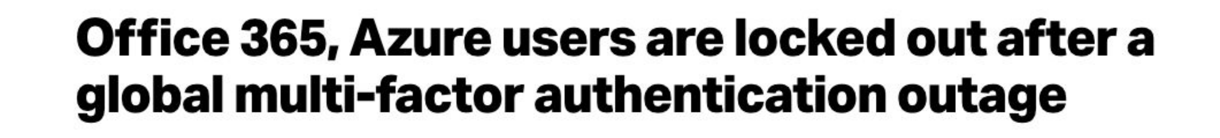
What this talk is not

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

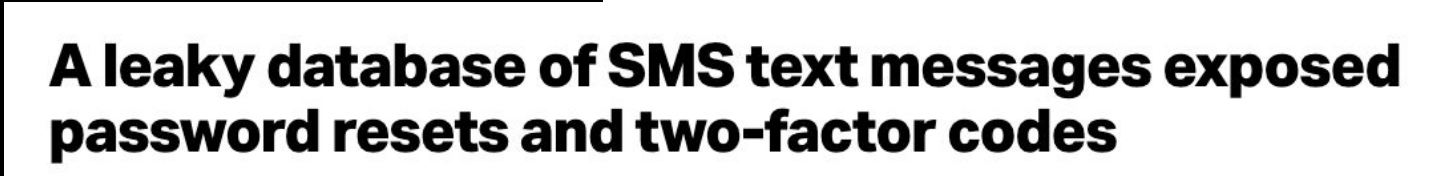
Why bother at all?



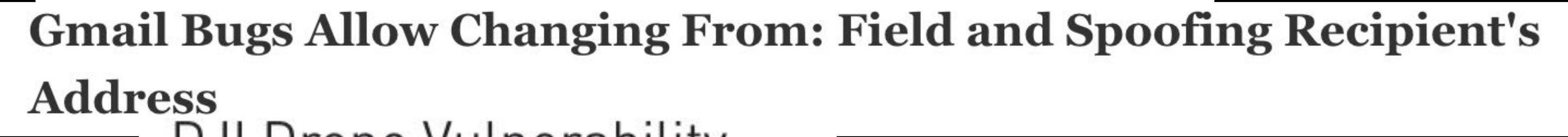
Amazon Suffers Security Breach; 80,000 Login Credentials Leaked (Updated)



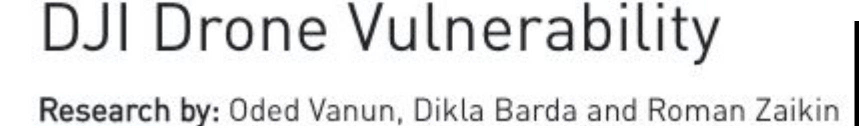
Office 365, Azure users are locked out after a global multi-factor authentication outage



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

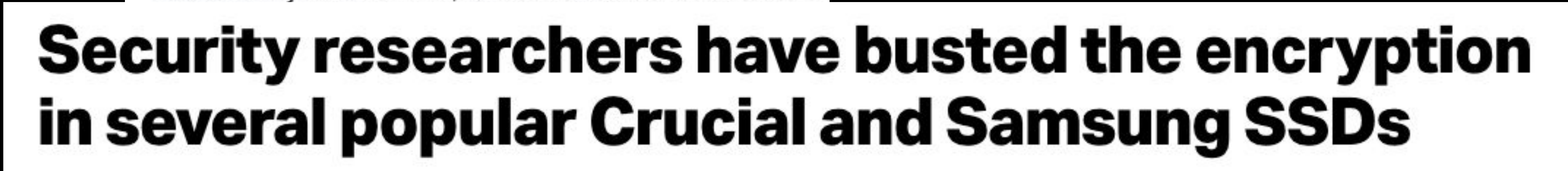


Gmail Bugs Allow Changing From: Field and Spoofing Recipient's Address



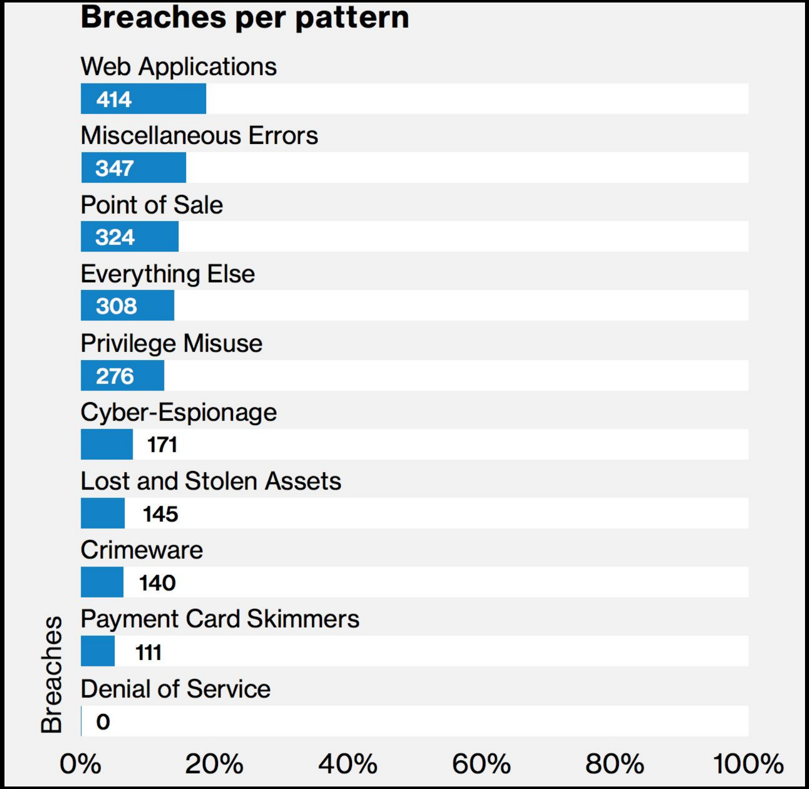
DJI Drone Vulnerability

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]

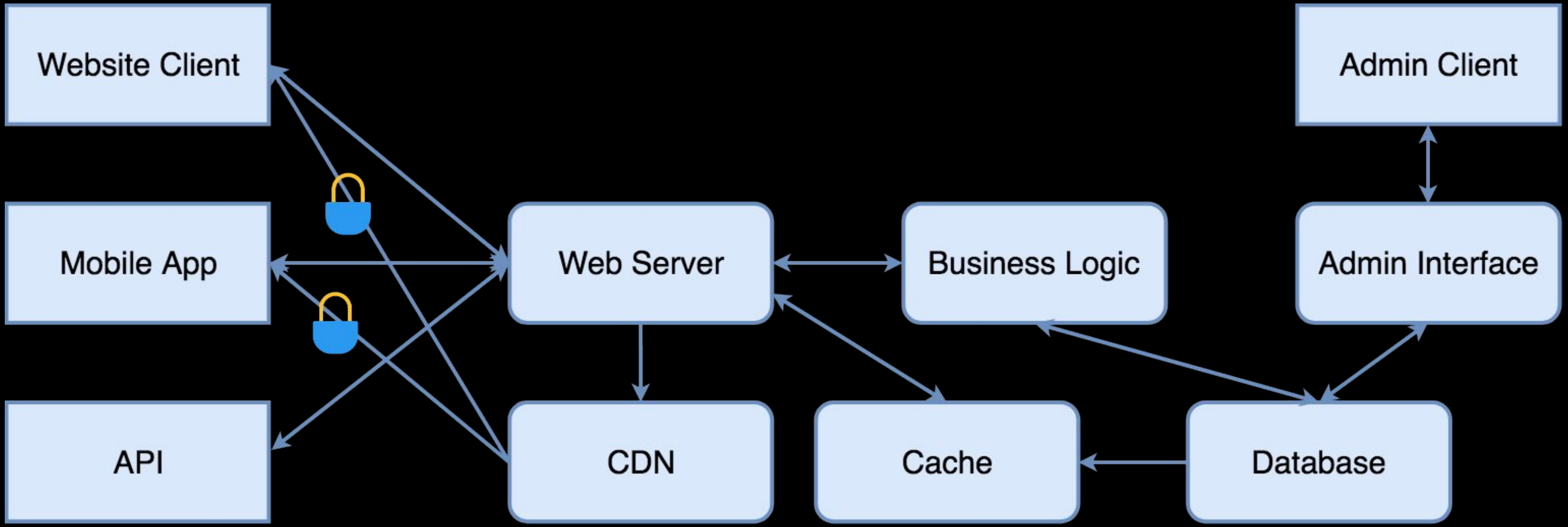


[1]https://www.verizonenterprise.com/resources/reports/rp_DBIR_2018_Report_en_xg.pdf

Facts

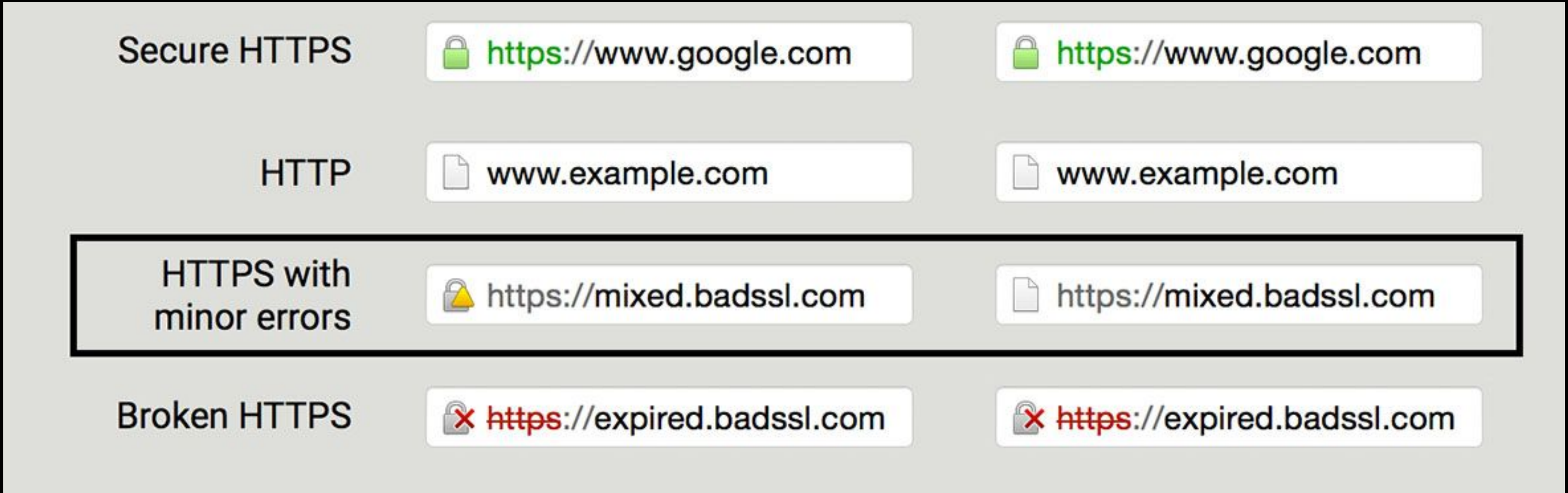
- ↳ Security is a team effort
- ↳ It can’t be bought as an add-on
- ↳ It can’t be patched on as an afterthought

Our Demo App


Website Client
Admin Client
Mobile App
Web Server
Business Logic
Admin Interface
API
CDN
Cache
Database

The Basics

TLS?



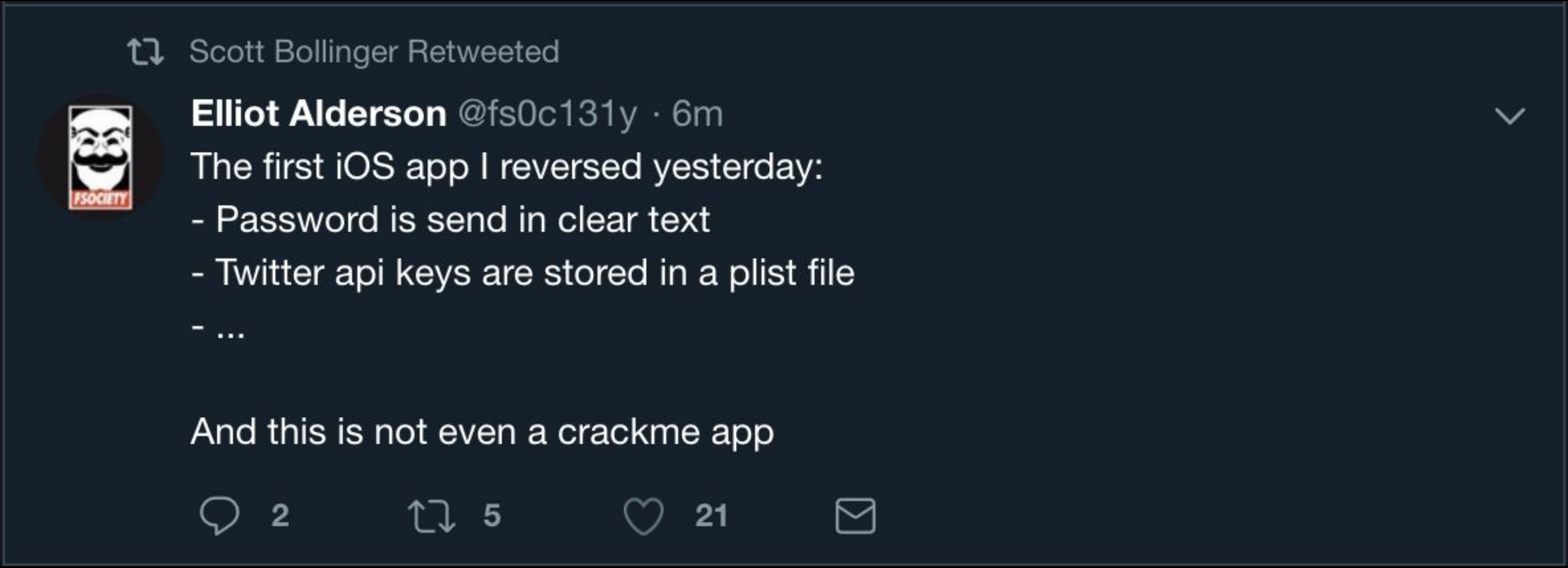
Secure HTTPS	https://www.google.com	https://www.google.com
HTTP	www.example.com	www.example.com
HTTPS with minor errors	https://mixed.badssl.com	https://mixed.badssl.com
Broken HTTPS	~~https~~://expired.badssl.com	~~https~~://expired.badssl.com

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



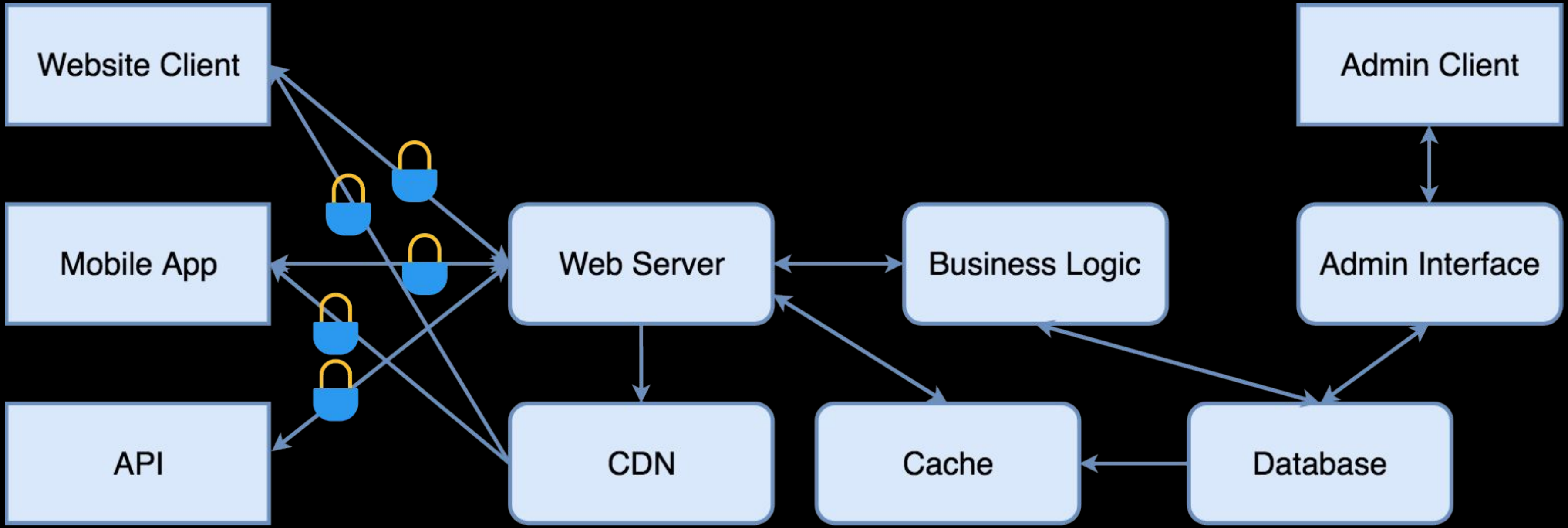
Scott Bollinger Retweeted

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


Website Client
Mobile App
API
Web Server
CDN
Business Logic
Cache
Admin Client
Admin Interface
Database

How to validate?

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

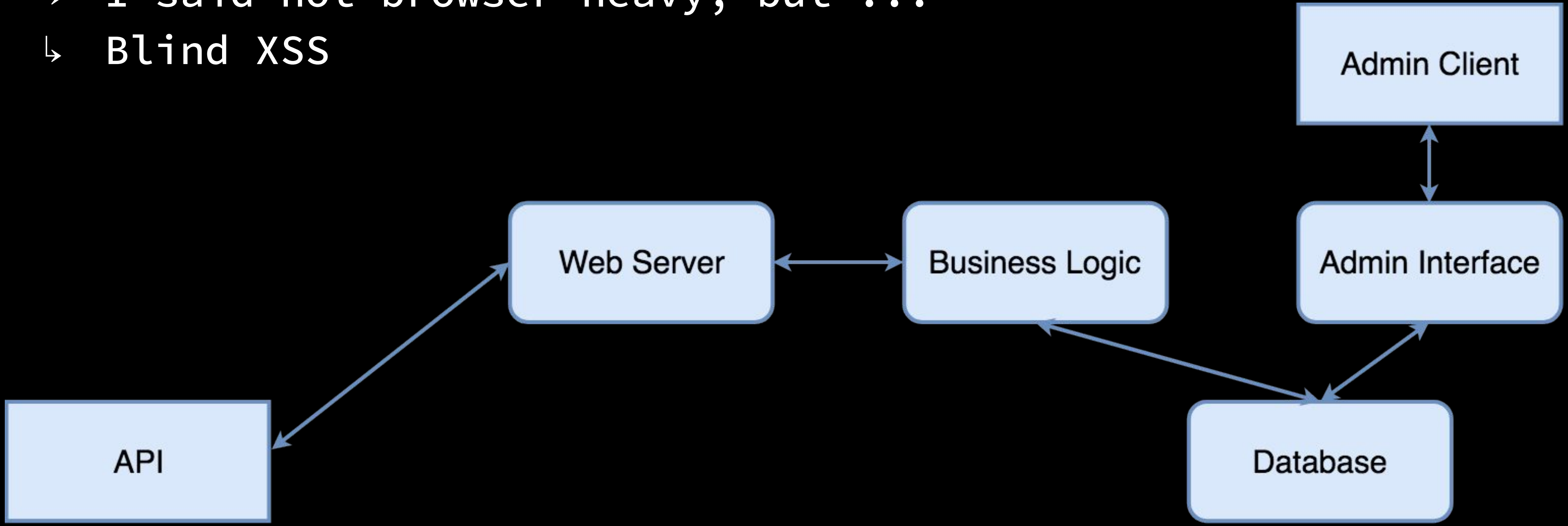
Intermediate

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

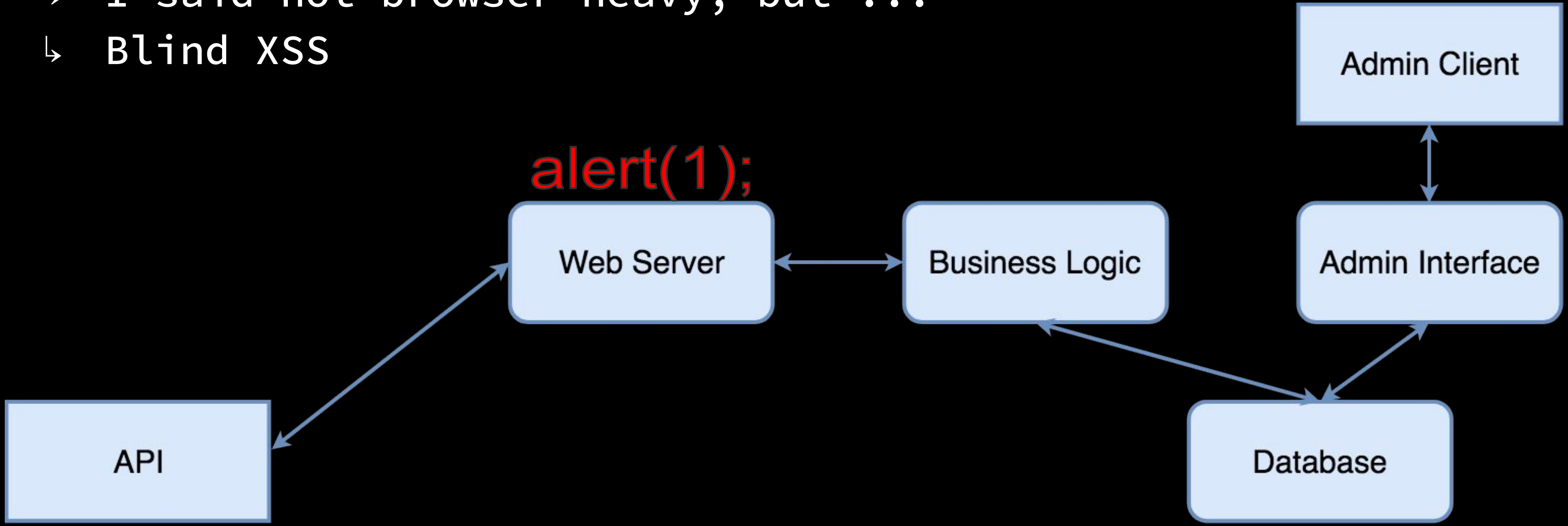
XSS

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



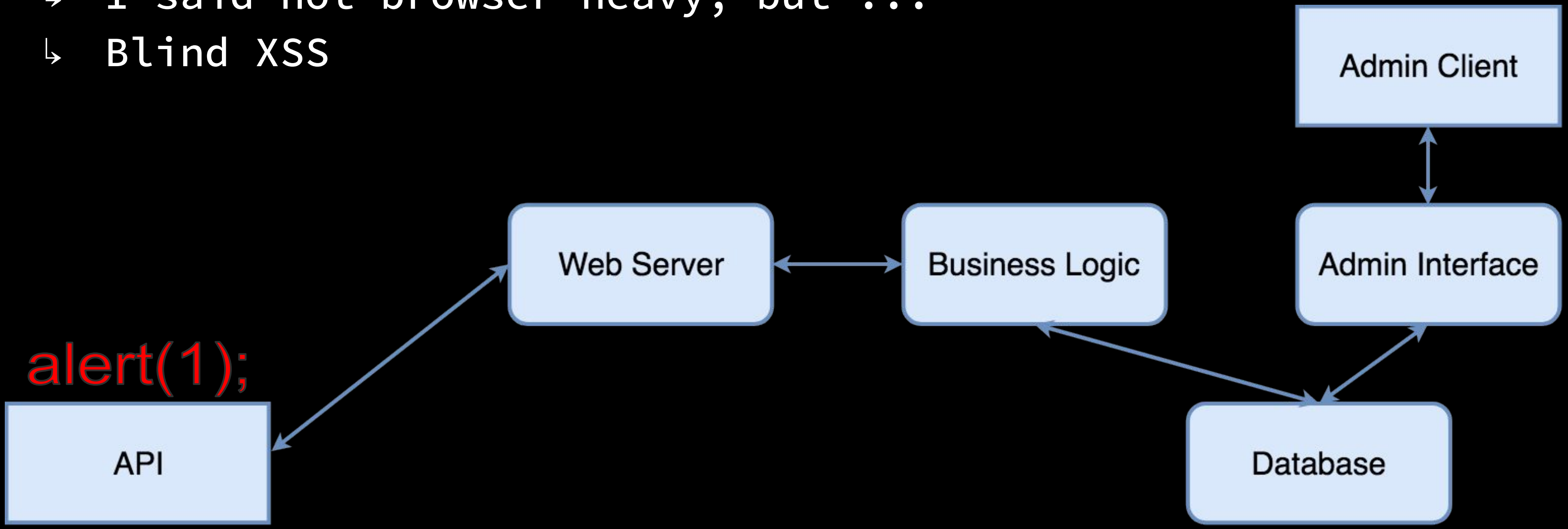
XSS

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



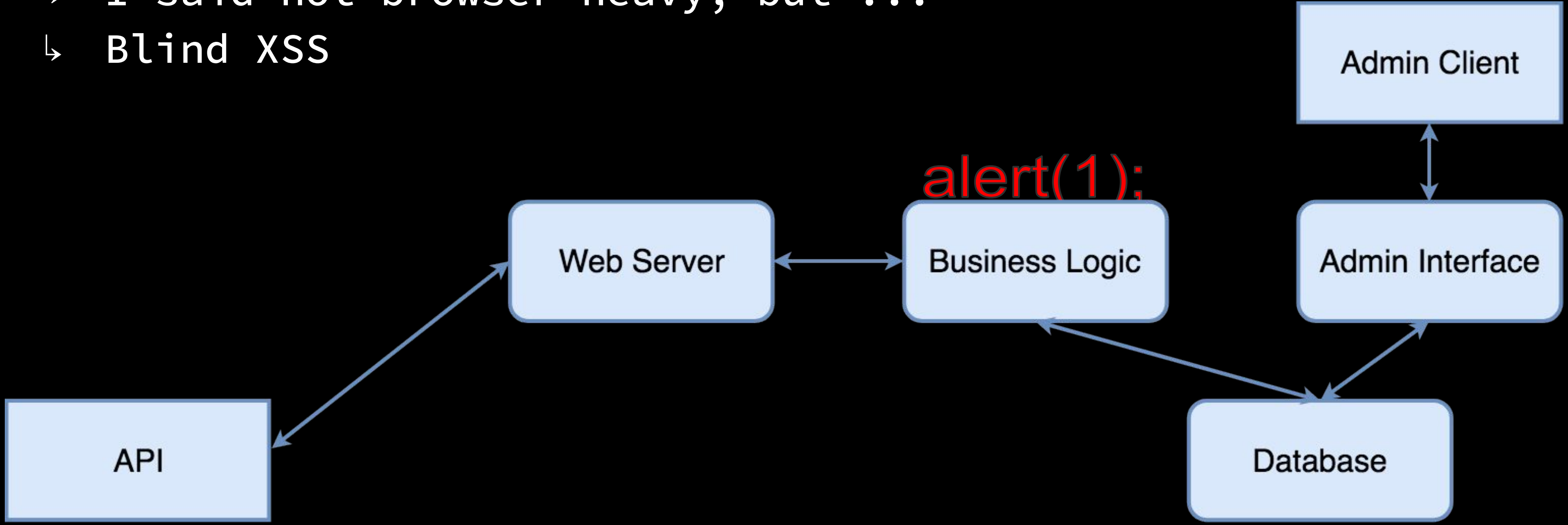
XSS

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



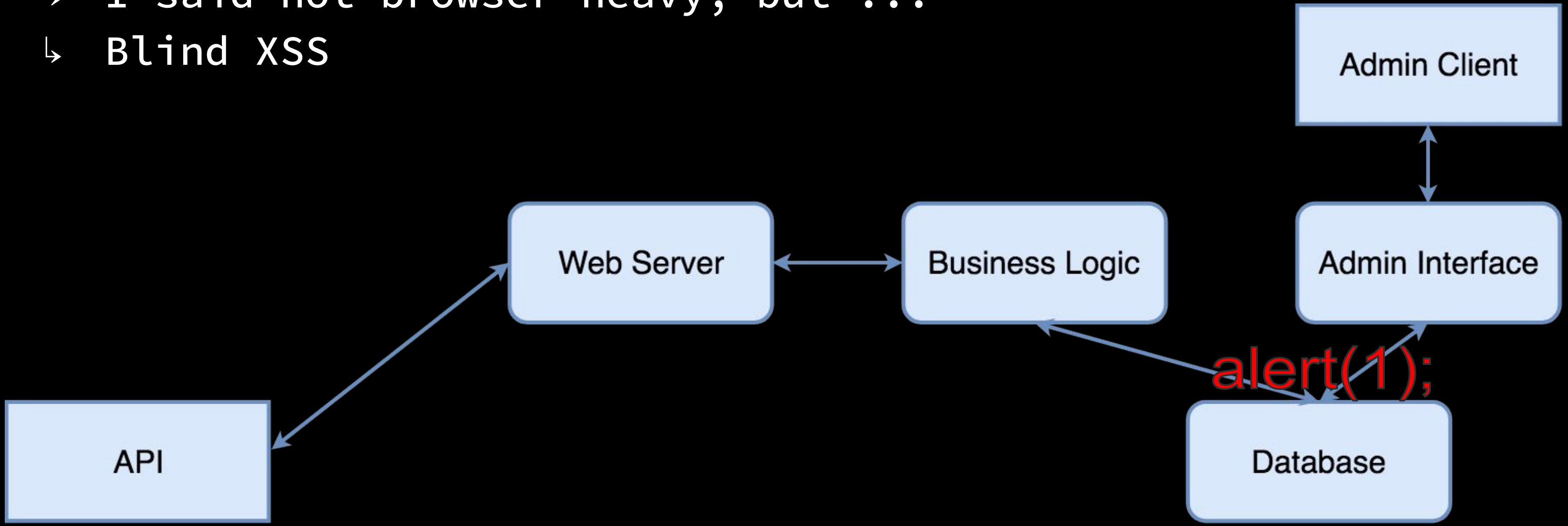
XSS

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



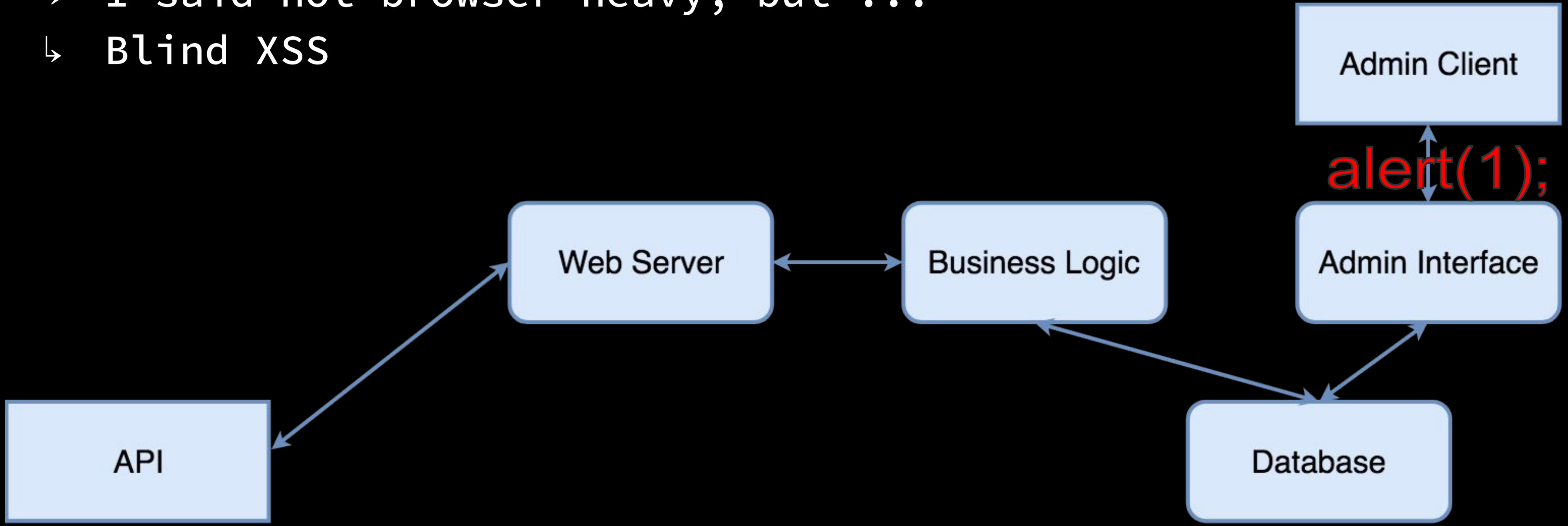
XSS

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



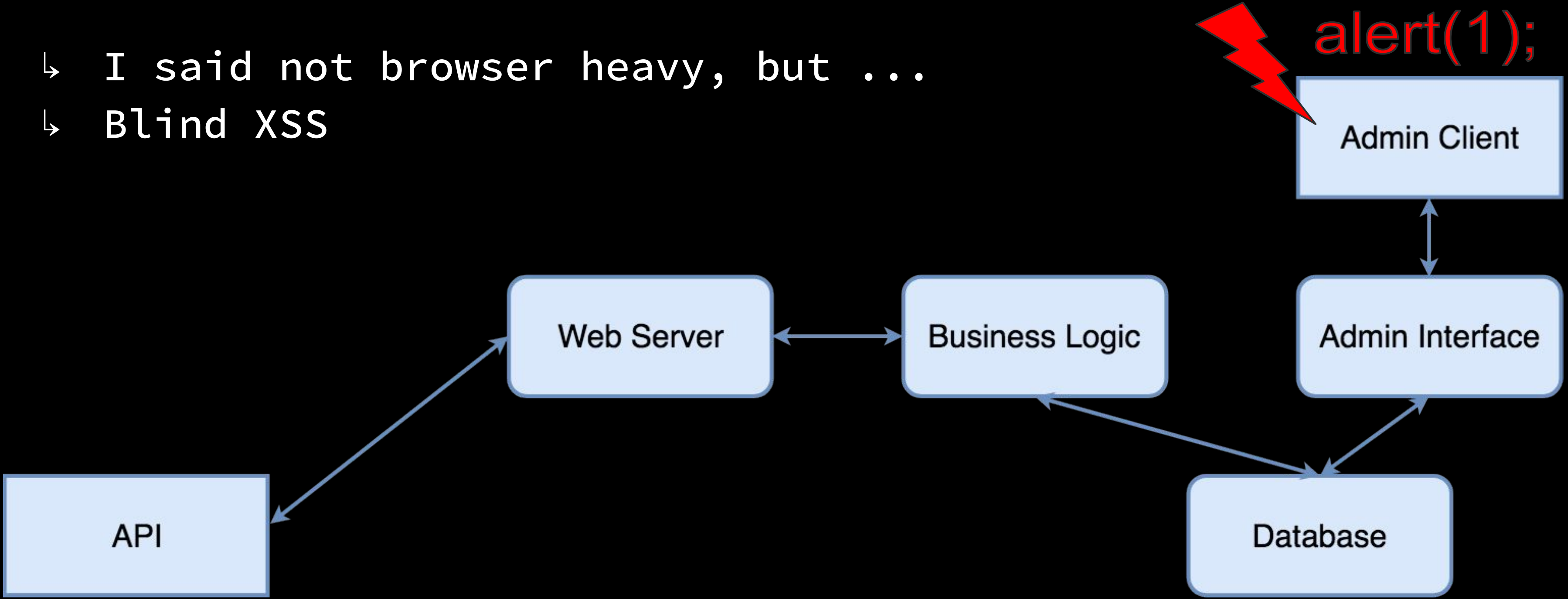
XSS

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



XSS

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



But also

- XXE (XML External Entity Processing)
- Injections
    - Code
    - SQL
    - ...

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

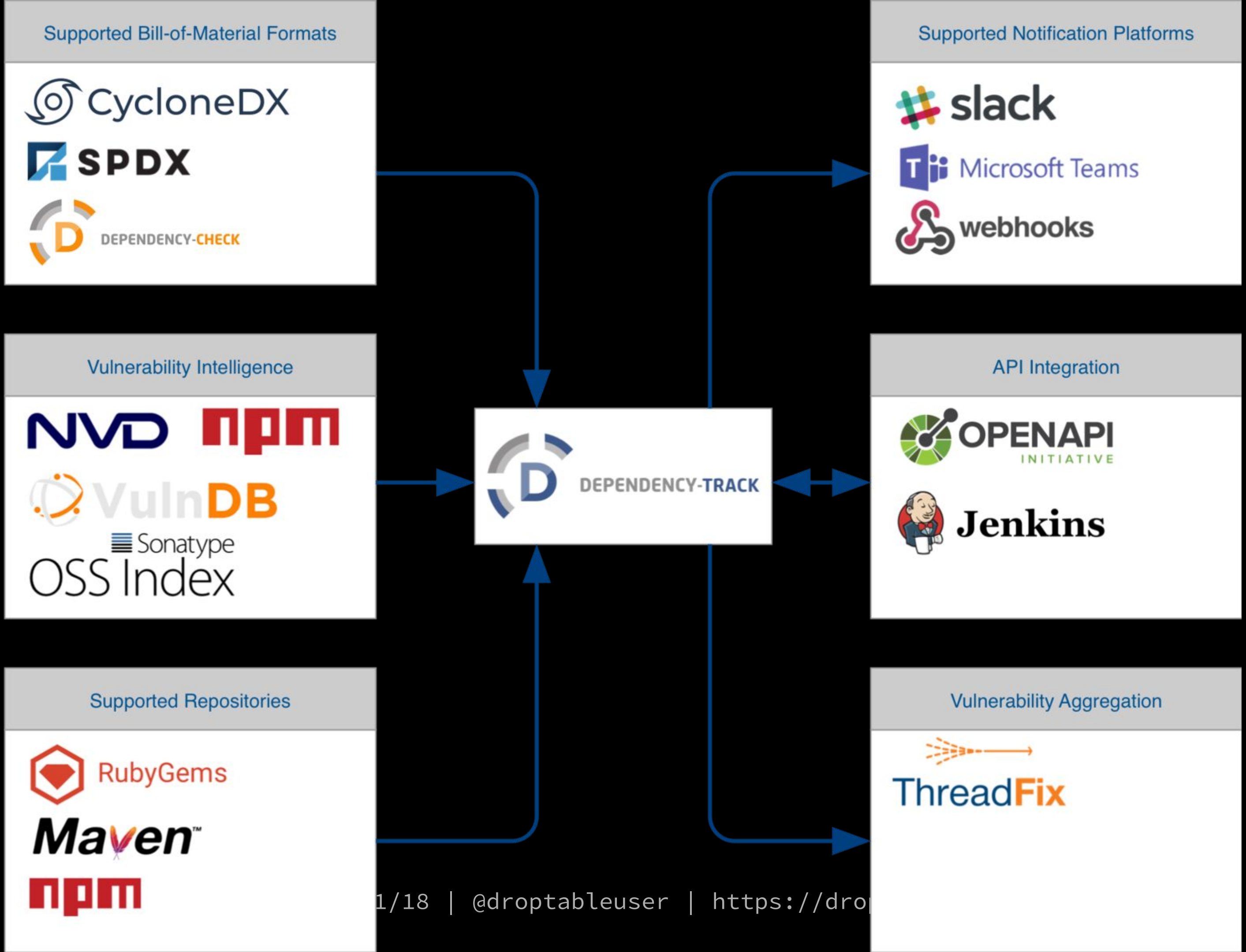
What did we already cover?

- ↳ 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



Supported Bill-of-Material Formats

Vulnerability Intelligence

Supported Repositories

Supported Notification Platforms

API Integration

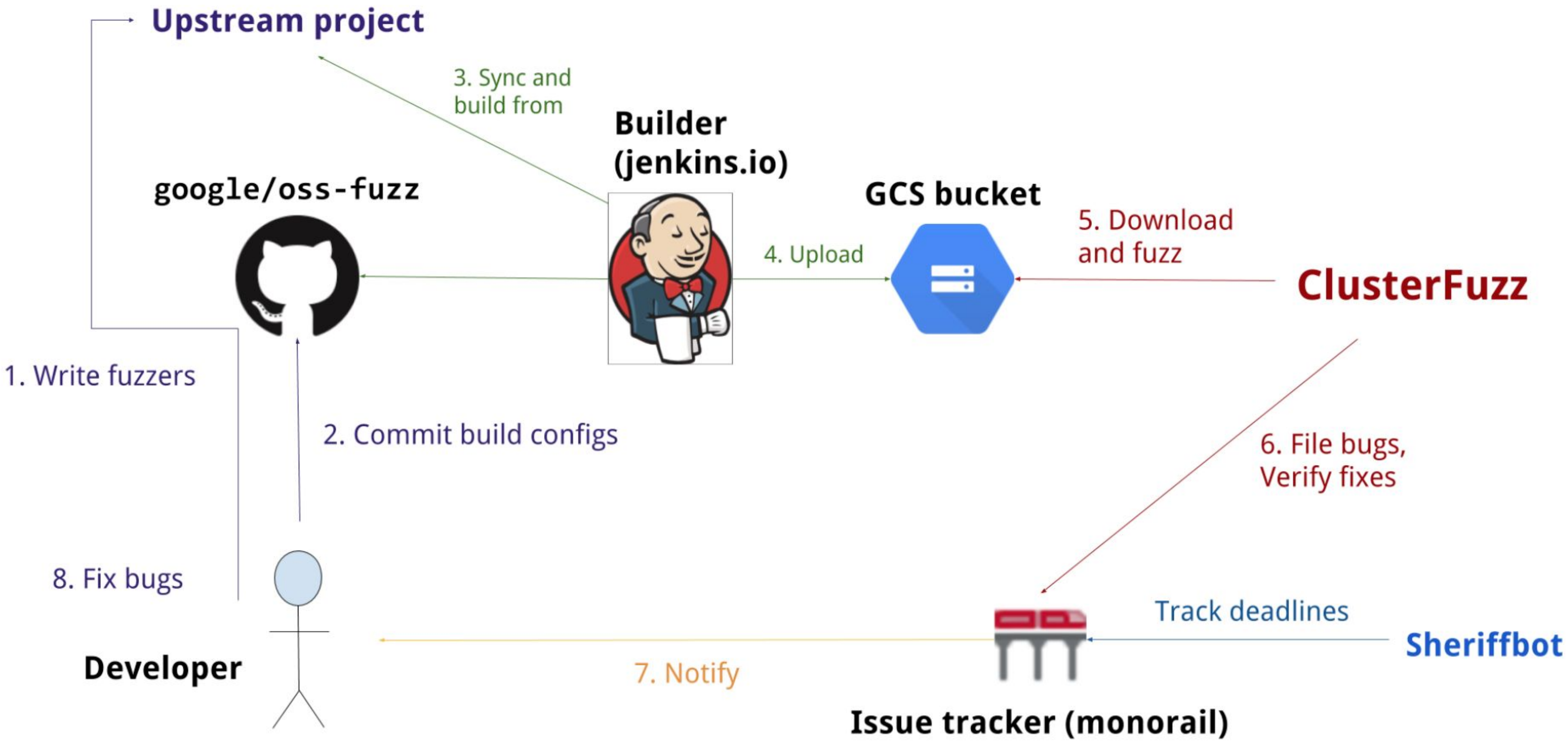
Vulnerability Aggregation

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

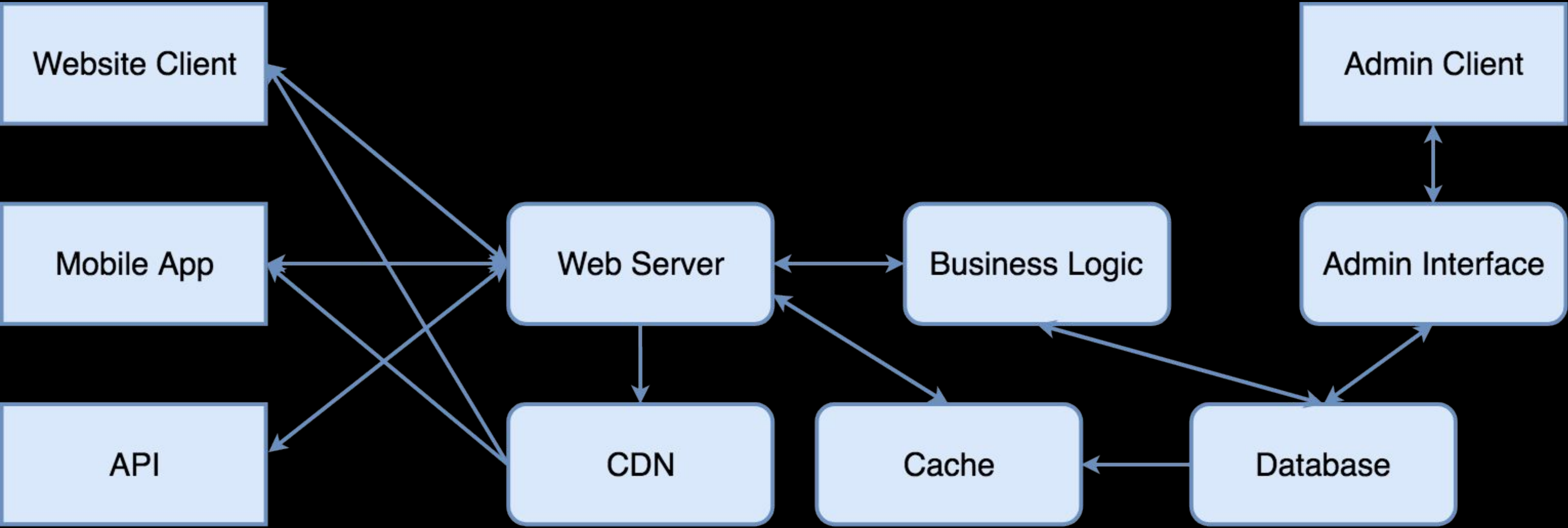
- ↳ Generating random input for test cases
- ↳ Complete code coverage
- ↳ Usually runs very long
- ↳ Fuzzing success means crash/error condition
  - ↳ This test case moves into regular testing setup


Upstream project
3. Sync and build from
Builder (jenkins.io)
google/oss-fuzz
GCS bucket
5. Download and fuzz
4. Upload
ClusterFuzz
1. Write fuzzers
2. Commit build configs
6. File bugs, Verify fixes
8. Fix bugs
Track deadlines
Sheriffbot
Developer
7. Notify
Issue tracker (monorail)

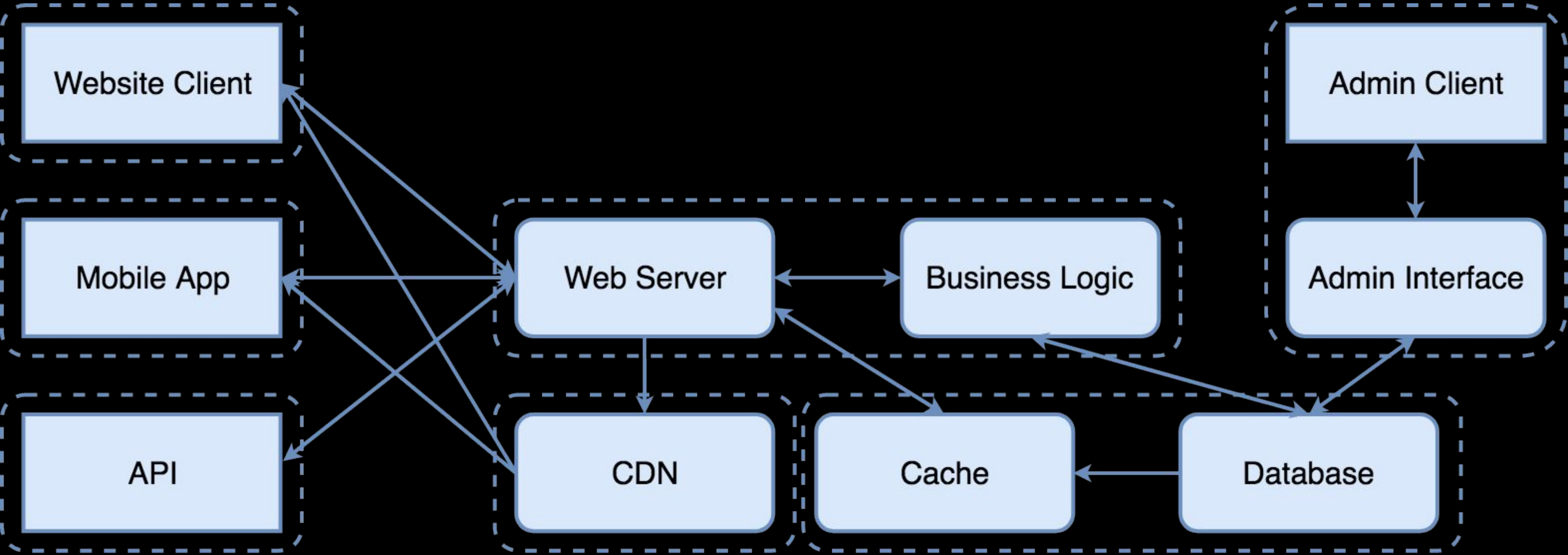
Threat Modeling

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

Application Design Review

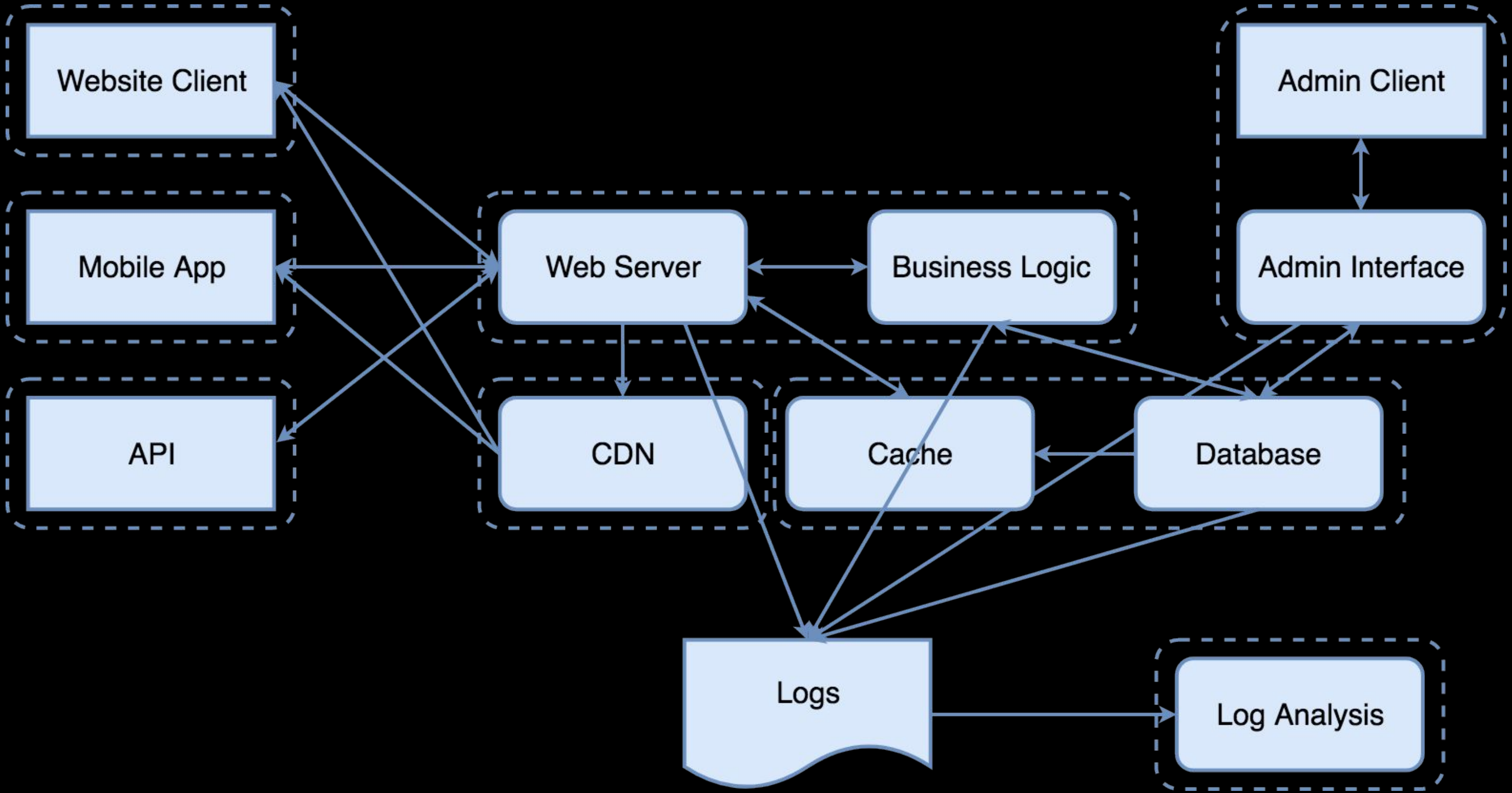


Add Trust Boundaries



Check for completeness

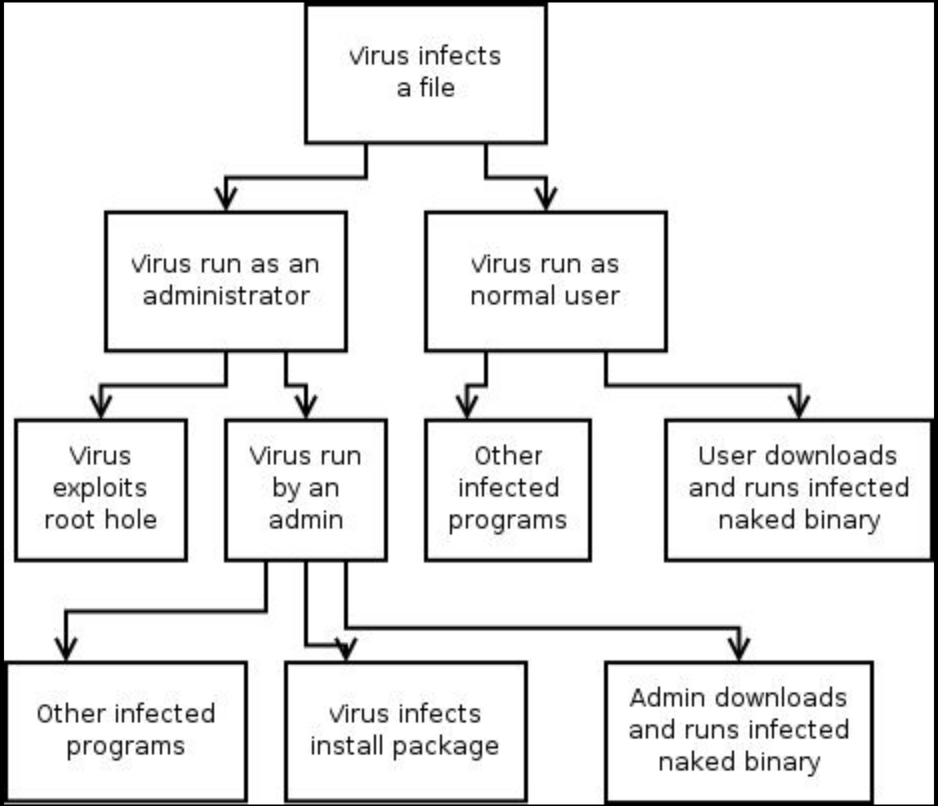
↳ Did we add things to the app?


Website Client
Mobile App
API
Web Server
Business Logic
Admin Client
Admin Interface
CDN
Cache
Database
Logs
Log Analysis

Threat Modeling

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

Attack Trees


Virus infects a file
Virus run as an administrator
Virus run as normal user
Virus exploits root hole
Virus run by an admin
Other infected programs
User downloads and runs infected naked binary
Other infected programs
Virus infects install package
Admin downloads and runs infected naked binary

Source: https://upload.wikimedia.org/wikipedia/commons/c/c6/Attack_tree_virus.png

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?