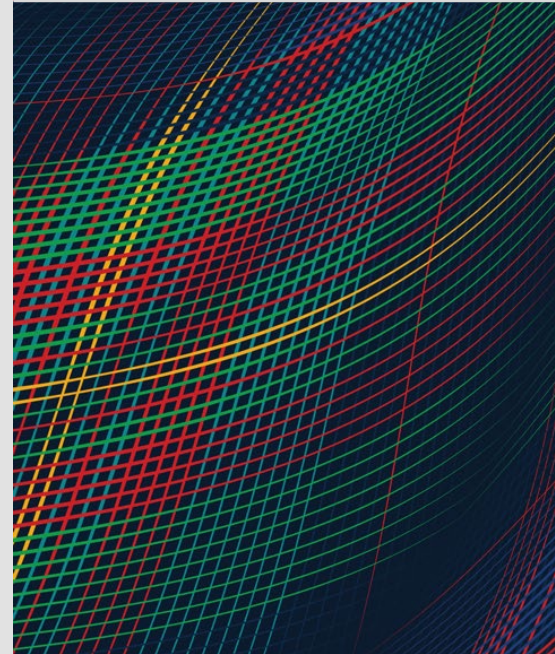


Modern API Security

Engineering security into the API lifecycle

AUGUST 6TH, 2024

Alejandro Gomez



Software Engineering Institute

About

Our Work

Publications

News and Events

Education and Outreach

Careers

[Home](#) > [Publications](#) > [Digital Library](#) > On the Design, Development, and Testing of Mode...

On the Design, Development, and Testing of Modern APIs

JULY 30, 2024 • WHITE PAPER

By [Alejandro Gomez](#) and [Alex Vesey](#)

This white paper discusses the design, desired qualities, development, testing, support, and security of modern application programming interfaces (APIs).

Document Markings

Carnegie Mellon University 2024

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

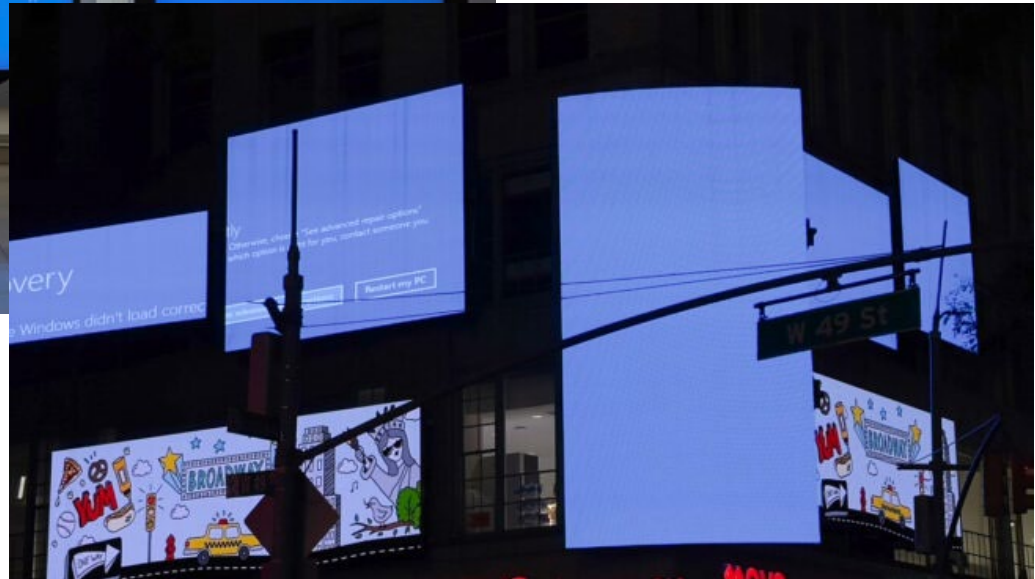
References herein to any specific entity, product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute nor of Carnegie Mellon University - Software Engineering Institute by any such named or represented entity.

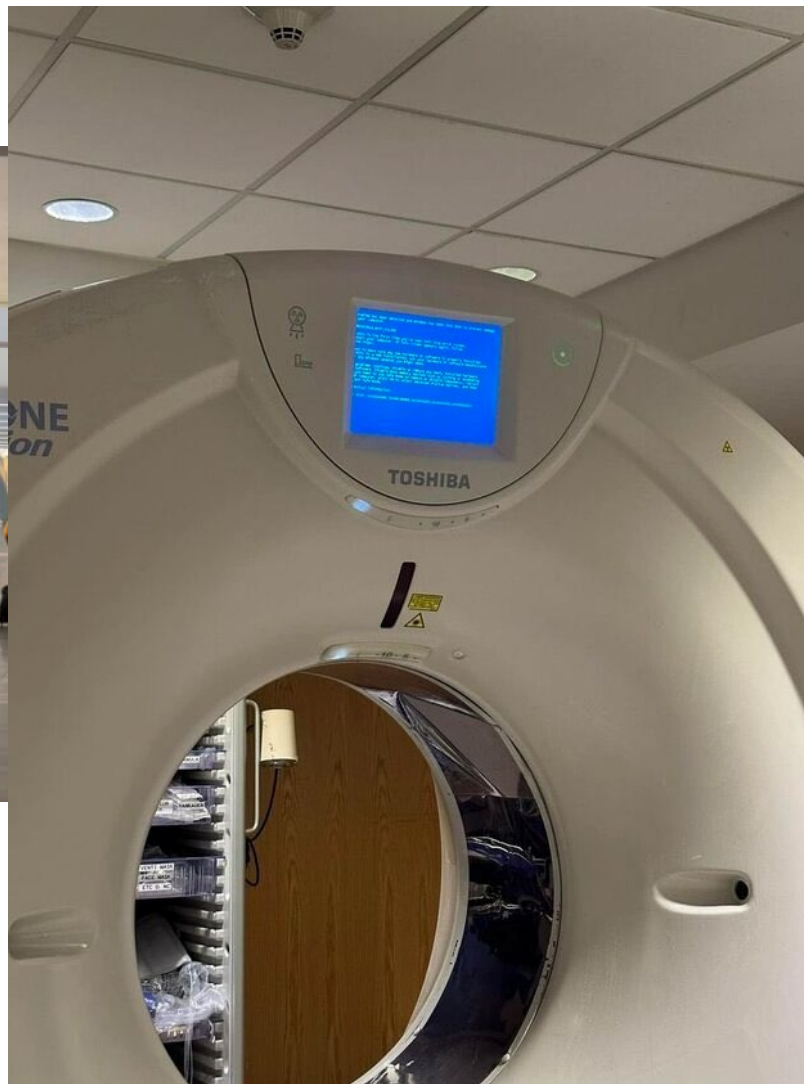
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.







Agenda


1. Definitions
2. The State of APIs
3. Engineering Security in APIs
4. Q&A

Modern API Security

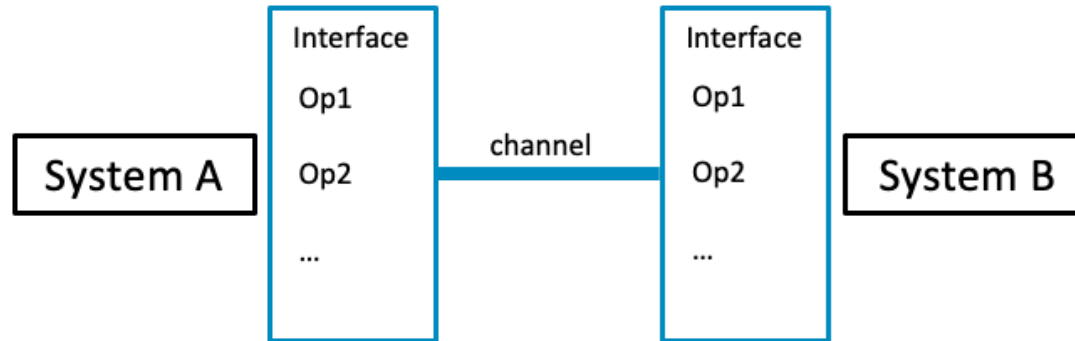
Definitions

Application Programming Interfaces

A dark-colored F-35 fighter jet is shown in flight, viewed from a low angle. The aircraft is dark grey or black, with a prominent orange-colored intake fan visible on the left side. The background is a cloudy sky.



“A set of functionalities independent of their implementation, allowing the implementation to vary without compromising the users of the component.” -Joshua Bloch



Security / Cybersecurity

Engineering Cybersecurity *must be:*

1. Empirical (i.e., data-driven)
2. Cost-effective
3. Maintain CIA qualities

Modern API Security

The State of Modern APIs

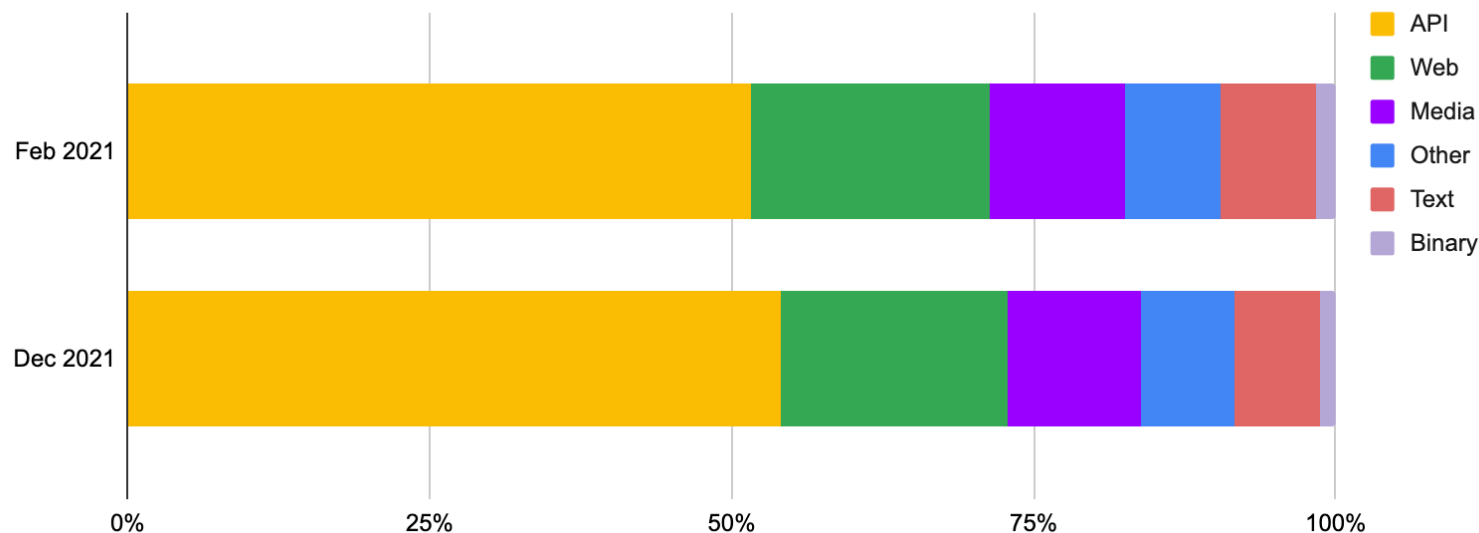
1. Ubiquitous
2. Expose system functionality to clients
3. Require evolvable interfaces

All these are avenues for attackers to exploit!

Internet traffic is overwhelmingly API-based

Its share is also increasing relative to other types of requests

Traffic composition by content type

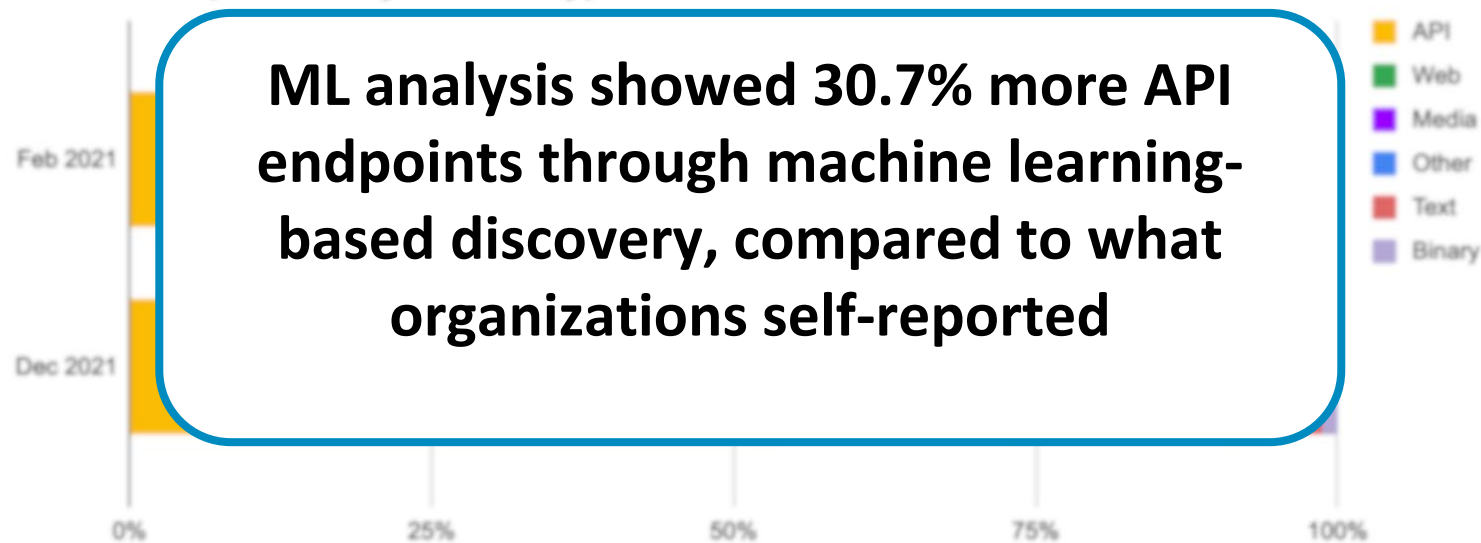


Source: [Cloudflare 2022](#)

Internet traffic is overwhelmingly API-based

Its share is also increasing relative to other types of requests

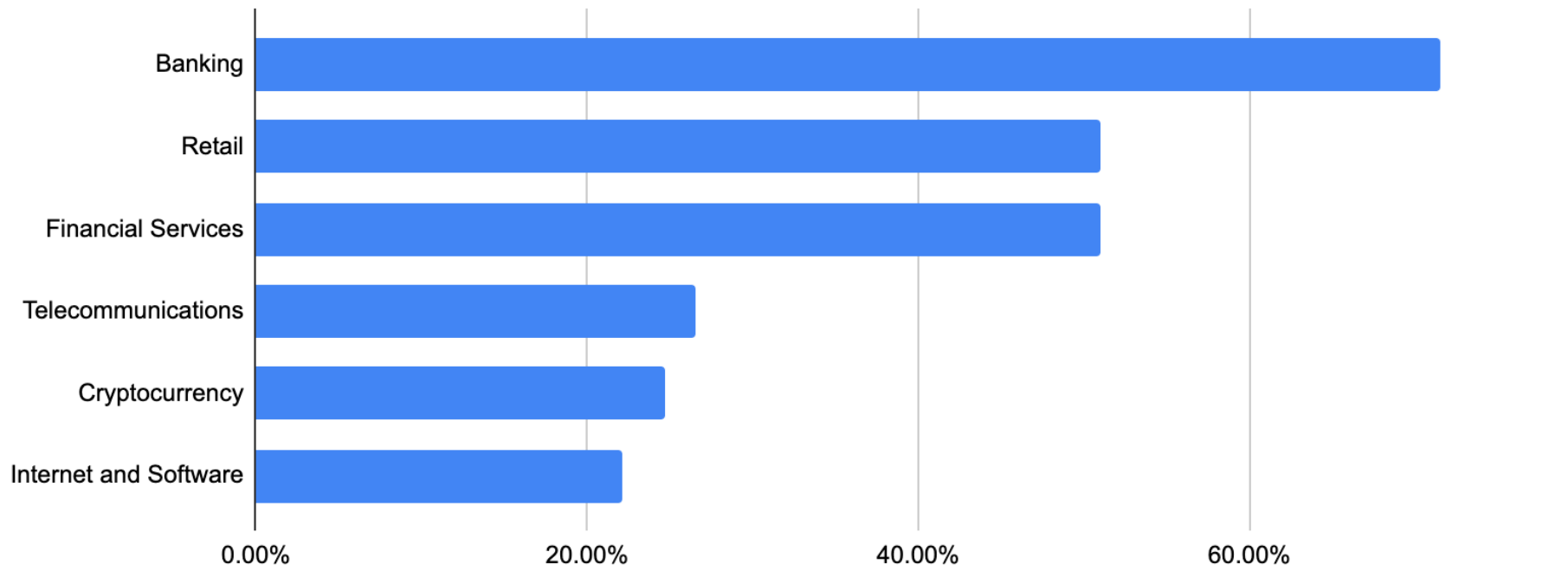
Traffic composition by content type



Source: Cloudflare 2022

API usage is growing in heavily regulated industries

Banking, Financial Services and Telecommunications are seeing a rise in API usage



Source: [Cloudflare 2022](#)

The U.S. Gov't's API use is increasing

Metrics collected from data.gov shows a linear increase in API use in recent years.



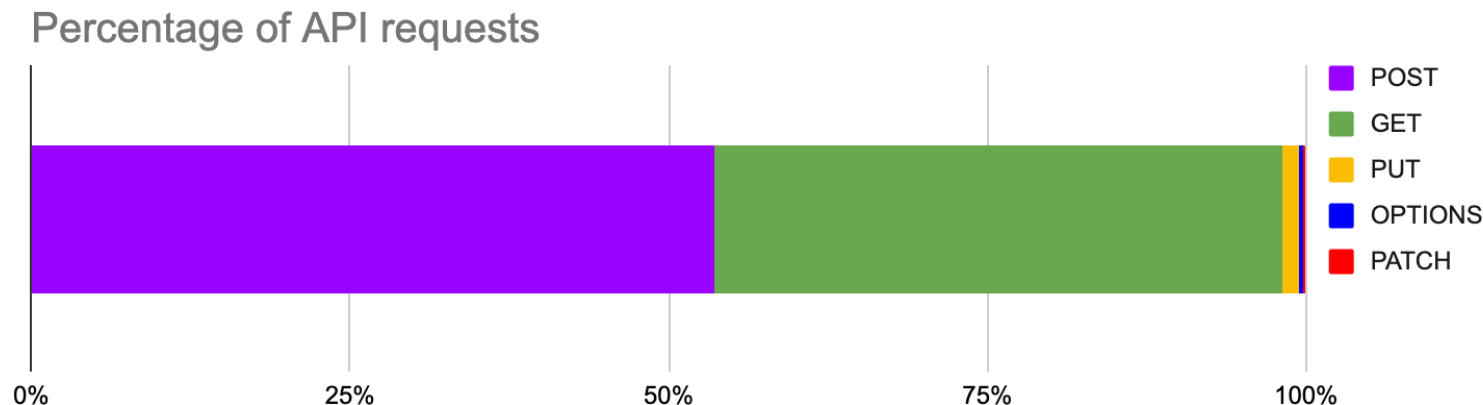
12,725,818,236
Hits

318,510
Unique API Keys

Source: data.gov 2024

Over half of API requests are write requests

Most internet APIs are making their system operations available to end users, increasing system attack surface



Source: [Cloudflare 2022](#)



Source: [CNN 2024](#)

SOS: SUPPORT OUR SUB - POSTMASTERS

**“Each time the user pressed ‘enter’
on the frozen screen, it would
silently update the record”**

Source: [CNN 2024](#)

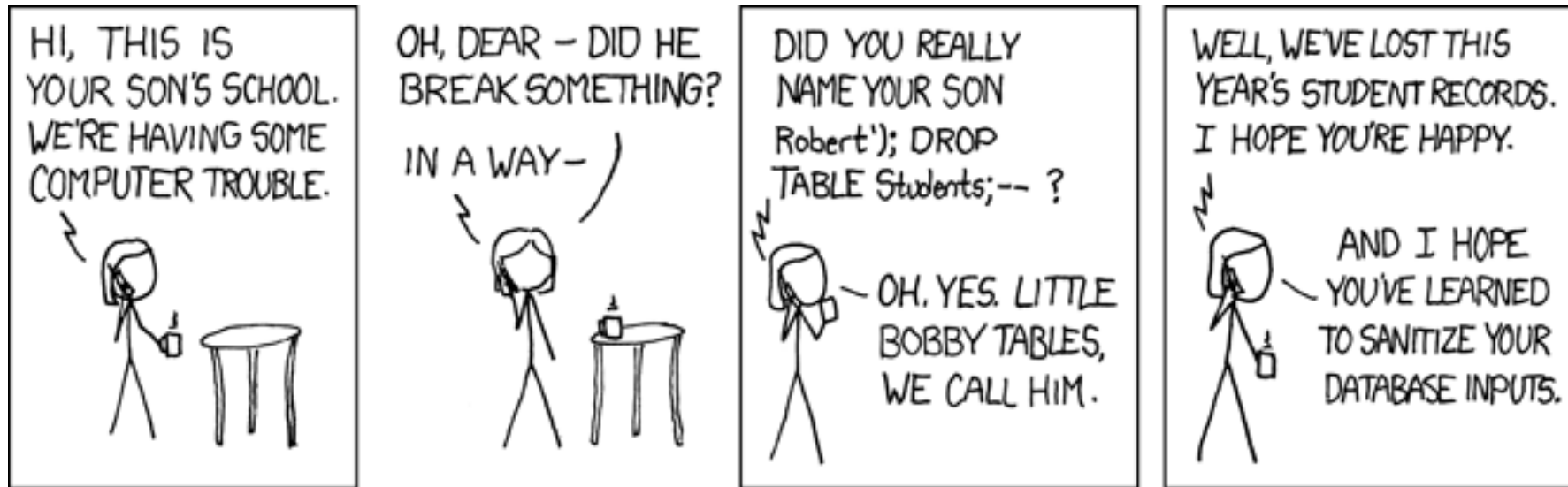
APIs create risk

By exposing fragile system functionality to unknown clients

- **92%** of organizations experienced an API-related security incident in 2022
- **57%** of these experienced multiple API-related security incidents
- **63%** of incidents involved a data breach or data loss

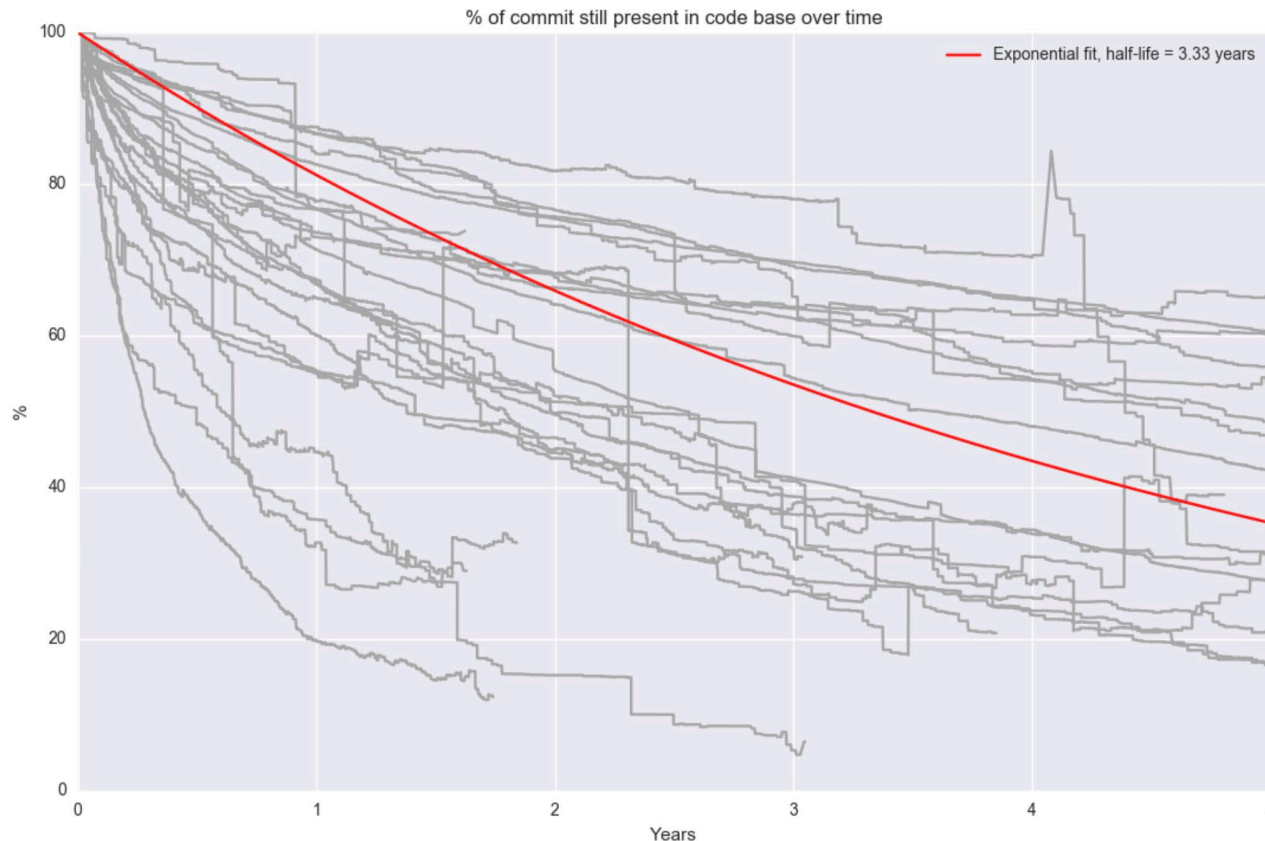
Source: Palo Alto Networks 2023

Akamai 2023



The rate of change in software is faster than ever

APIs evolve at the rate of change they're expected to accommodate



Source: Bernhardtsson 2016

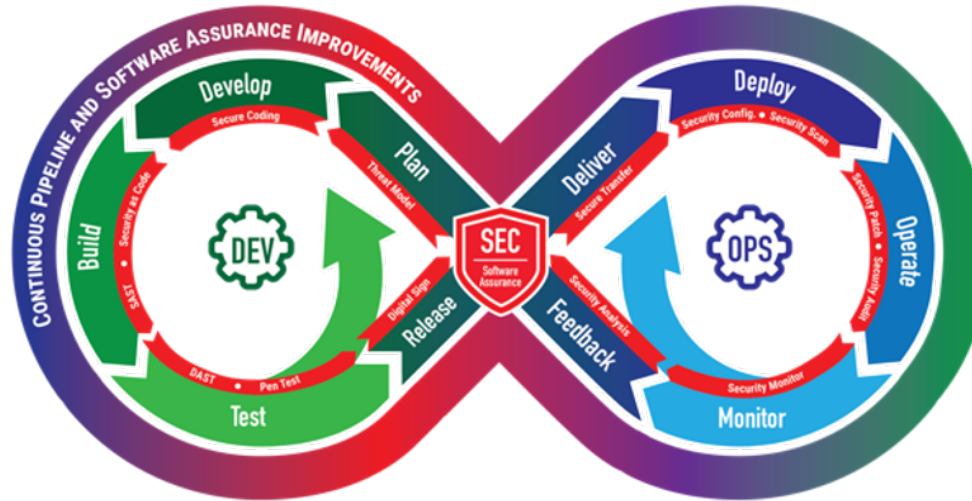
We as software designers need to provide assurance that:

*APIs work as expected for its users.
Are trustworthy instead of vulnerable.
Are an asset instead of a liability.
Create value instead of harm.*

Modern API Security

Engineering Security in APIs

DevSecOps



Development



Scanning for vulnerabilities on each change

Scanners enable automation of vulnerability detection

Static code analyzer

Runtime scanner

Configuration scanner

IaC scanner

Secrets scanner

Database scanner

Port scanner

Cloud vulnerability scanner

Network scanner

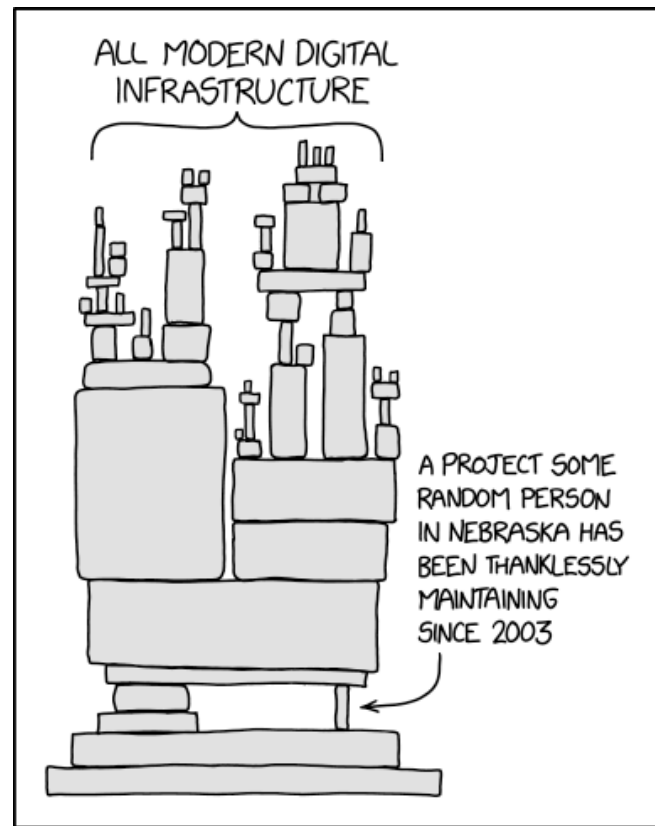
DAST scanner

Open-Source scanner

Container scanner

License scanner

Code Quality scanner



Versioning

...at source code level

- Code changes
- Manifests
- Configurations
- Images
- Documentation
- OS
- Libraries (lockfiles)

...at version control level

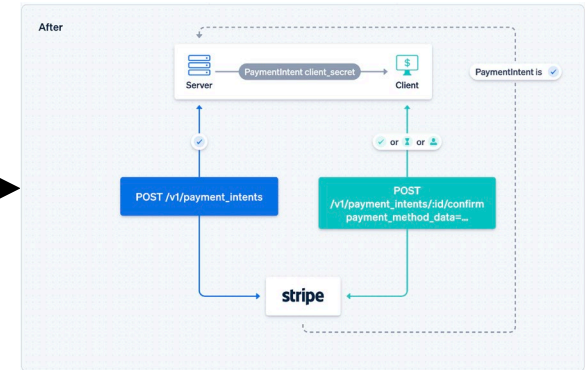
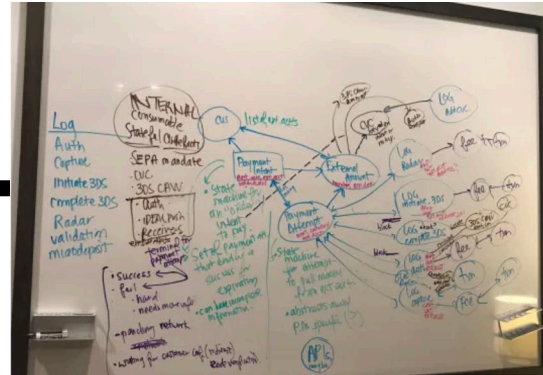
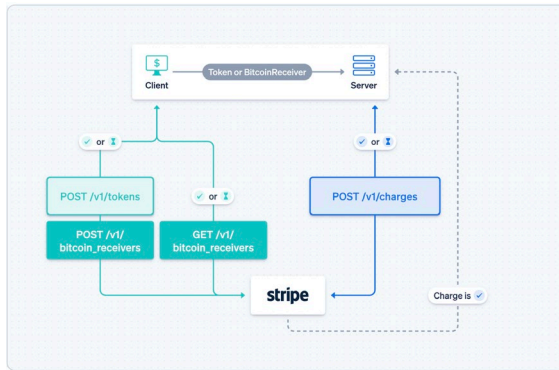
- Version interface separately from the system it supports.
- Various strategies to allow for clients to use different API versions including:
 - Endpoint
 - Feature flags
 - Using data structures that allow for different compatibility

Case study: **stripe**

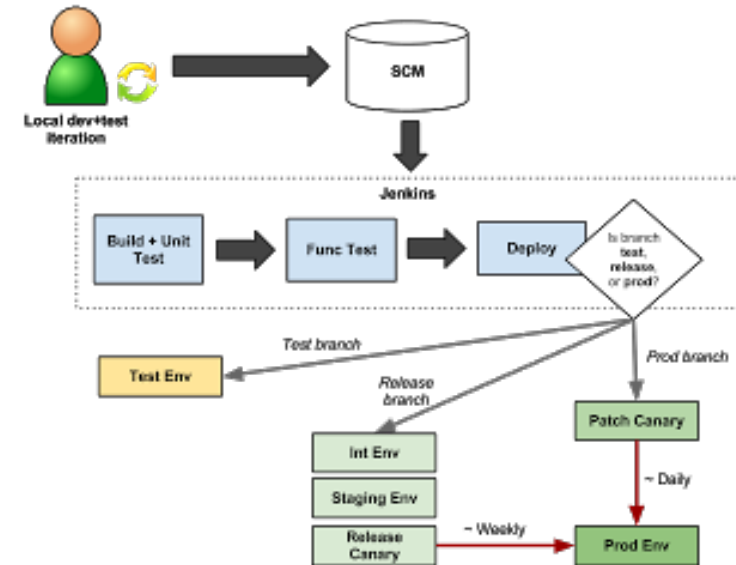
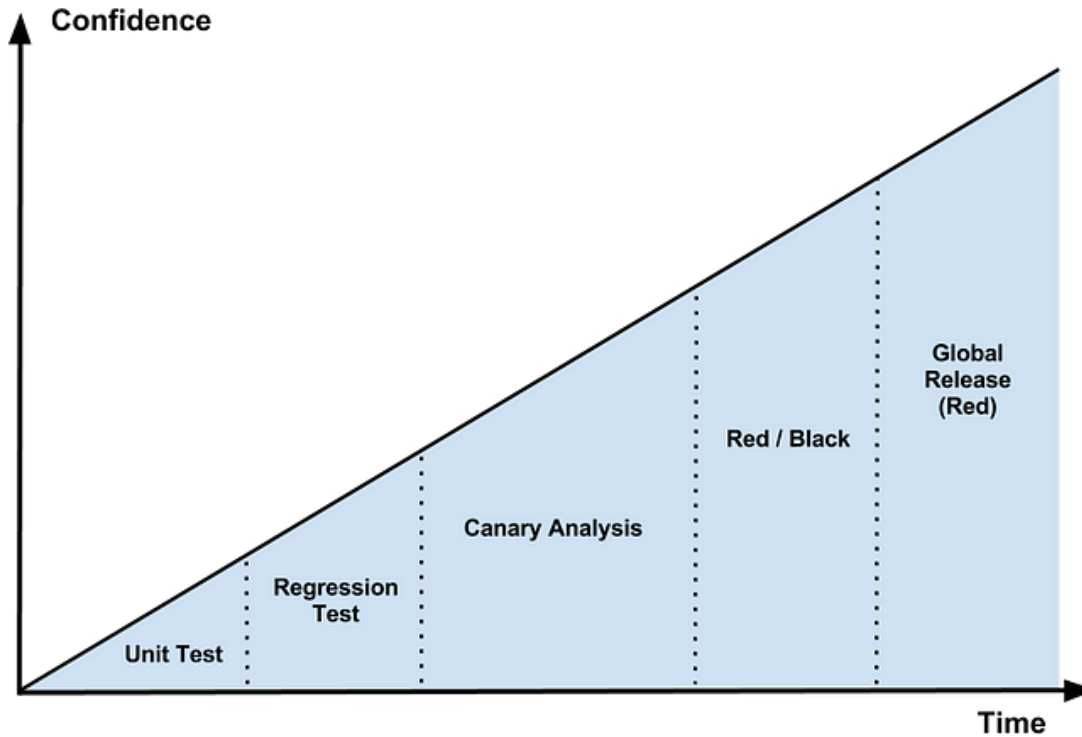
How do you design an API for all the world's payment methods?

Q: How do you design an API for all the world's payments?

- You start with a small, focused approach, then iterate on it.
- Sometimes add extensions.
- Occasionally, do a complete rewrite (with backward compatibility)



Testing



Testing Strategies for API Inputs

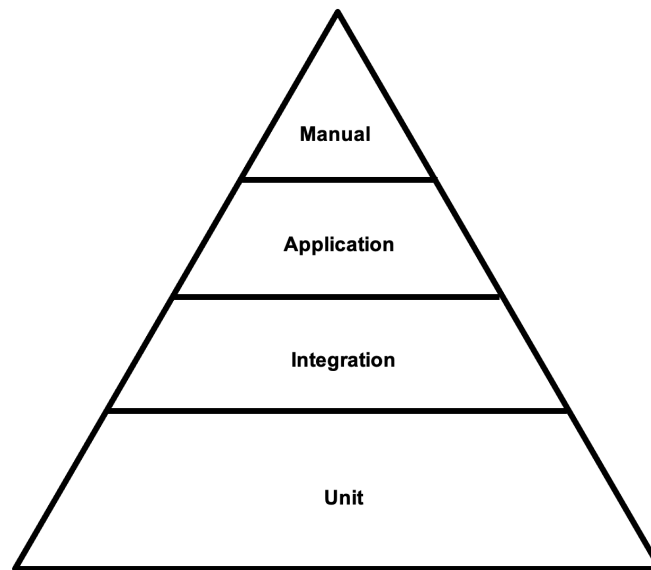
Testing for resilience of API inputs can be broken down as...

Deterministic

- Unit
- Integration
- E2E
- Accessibility

Non-Deterministic

- Fuzzers
- Chaos testing
- QA team
- Manual
- Red-team testing



Deployment/Operations

How do you engineer secure API deployments?

Canary deployments

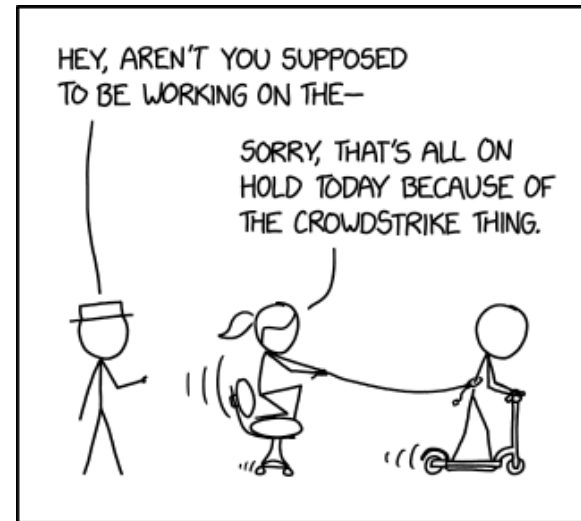
- Allows for feature testing
- Limits blast radius

Rollbacks

- Enables rapid failure recovery
- Incentivizes small, atomic commits in development.

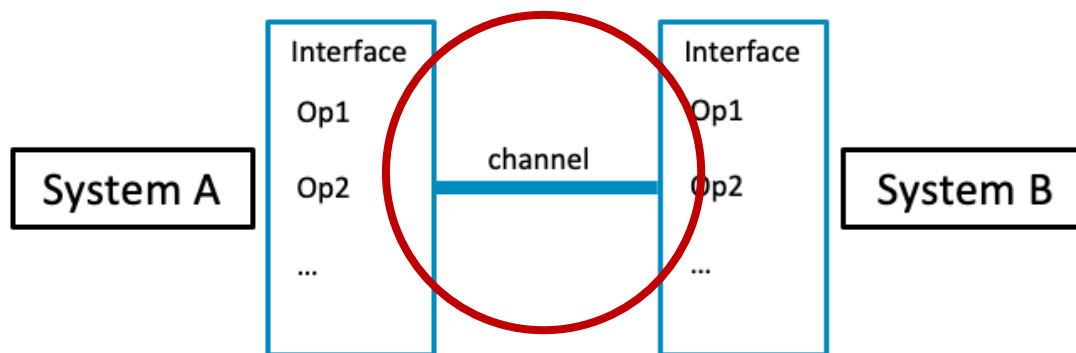
Monitoring & Logging

- Find problems *before* they appear



PROTIP: AS LONG AS YOU'RE NOT ACTUALLY IN CHARGE OF FIXING THE CROWDSTRIKE THING, YOU CAN USE THIS EXCUSE FOR PRETTY MUCH ANYTHING YOU WANT TO DO TODAY.

Security - Channel



a network, wire, pipe, etc.

API Channel Security

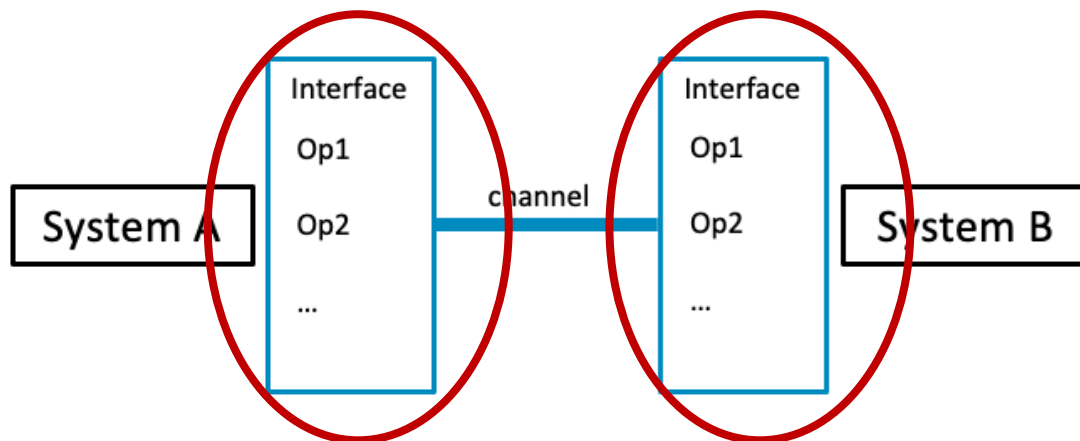
Defense-in-Depth through Zero-Trust Architecture

- In Zero Trust (ZT), there's no distinction between internal vs external network security
- Perform Schema Validation of requests
- Use TLS, IPSec or similar protocols that provide confidentiality, integrity, authentication –even inside a “safe” network!



BeyondCorp

Security - Interface



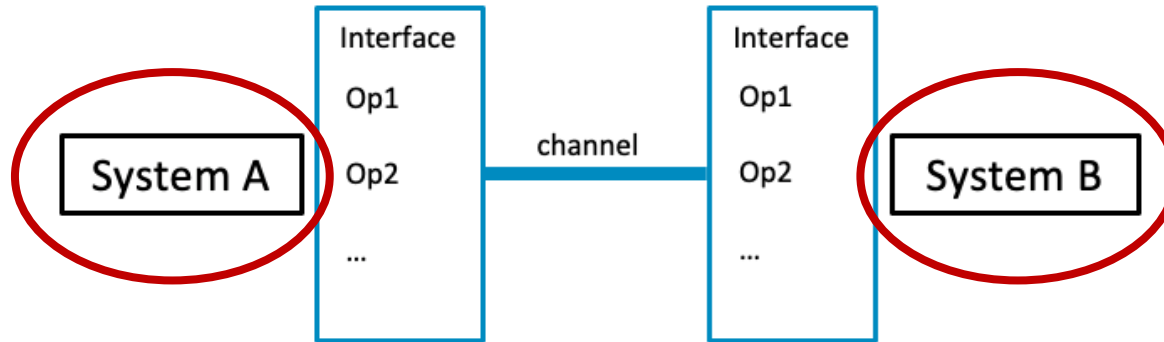
the set of functionalities exposed to clients

API Interface Security

“Be liberal with your inputs, restrictive with your outputs.”

- Runtime:
 - API Gateways are frequently used to aggregate multiple functionality
 - Use Stress Testing to test assure rate-limiting mechanism
- Config Management:
 - Find vulnerable configurations in authentication & middleware
 - Configuration of policy engine that allows access to resources

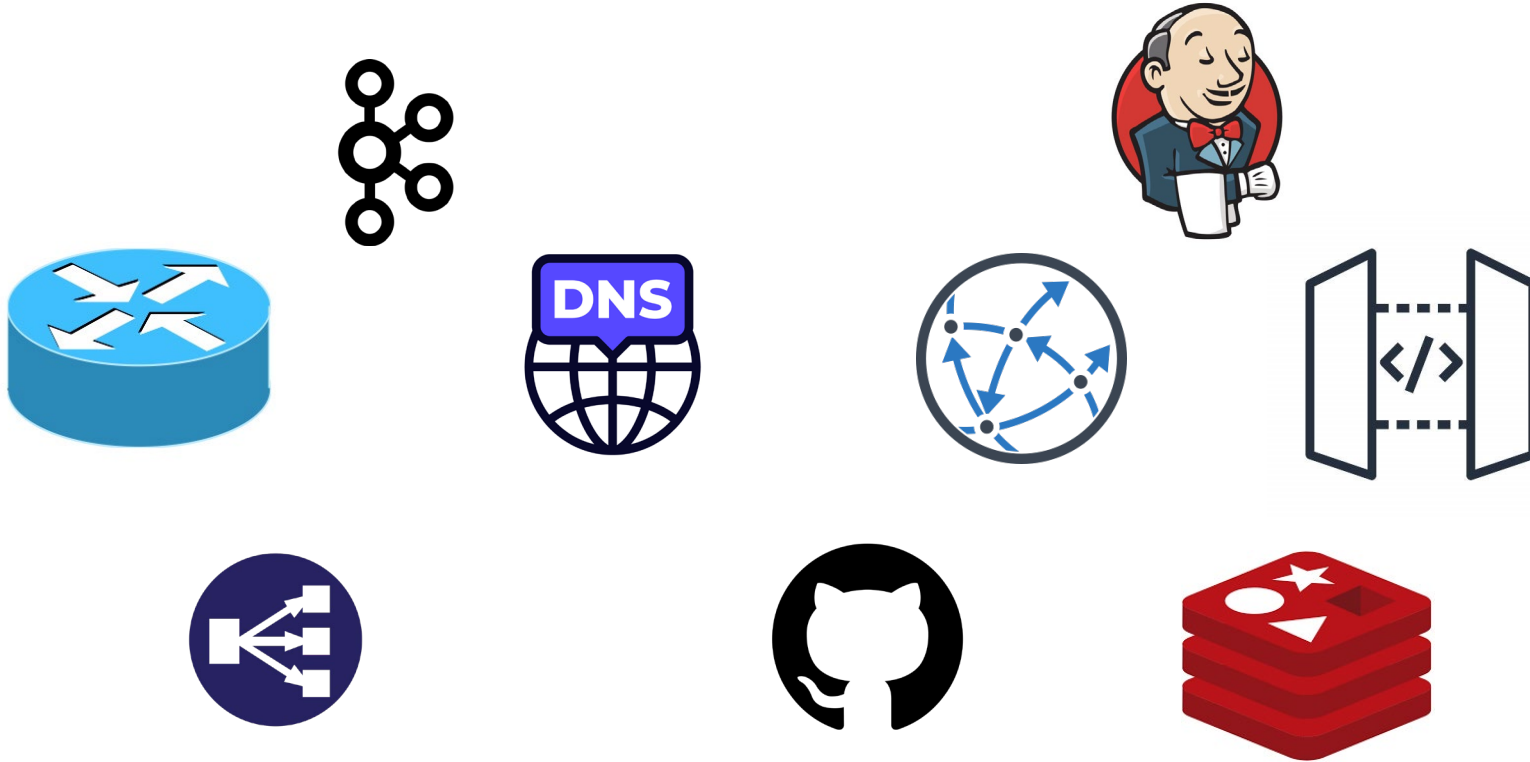
Security - Systems



a system that performs a task and delegates the output to an interface

API System Security

Securing the systems that enable APIs





Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

```
EXCEPTION_RECORD: fffffb0d18d3ec28 -- (.cxr 0xfffffb0d18d3ec28)
ExceptionAddress: fffff8021df335a1 (csagent+0x000000000000e35a1)
ExceptionCode: c0000005 (Access violation)
ExceptionFlags: 00000000
NumberParameters: 2
  Parameter[0]: 0000000000000000
  Parameter[1]: 0000000000000009c
Attempt to read from address 0000000000000009c

CONTEXT: fffffb0d18d3e460 -- (.cxr 0xfffffb0d18d3e460)
rax=fffffb0d18d3f2b0 rbx=0000000000000000 rcx=0000000000000003
rdx=fffffb0d18d3f280 rsi=ffff9a81b596f9a4 rdi=ffff9a81b596605c
rip=fffff8021df335a1 rsp=fffffb0d18d3ee60 rbp=fffffb0d18d3ef60
  r8=0000000000000009c  r9=0000000000000000  r10=0000000000000000
  r11=00000000000000014 r12=fffffb0d18d3ef28 r13=fffffb0d18d3f0d0
  r14=0000000000000001a r15=00000000000000004
iopl=0         nv up ei pl nz na po nc
cs=0010  ss=0018  ds=002b  es=002b  fs=0053  gs=002b             efl=00050206
csagent+0xe35a1:
fffff802`1df335a1 458b08          mov     r9d,dword ptr [r8] ds:002b:00000000`00000009c=????????
Resetting default scope

BLACKBOXBSD: 1 (!blackboxbsd)

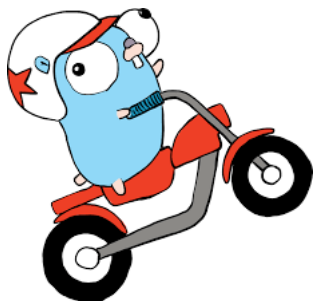
BLACKBOXNTFS: 1 (!blackboxntfs)

BLACKBOXPNP: 1 (!blackboxpnf)

BLACKBOXWINLOGON: 1

PROCESS_NAME: System
READ_ADDRESS: 0000000000000009c
ERROR_CODE: (NTSTATUS) 0xc0000005 - The instruction at 0x%p referenced memory at 0x%p. The memory could not
EXCEPTION_CODE_STR: c0000005
EXCEPTION_PARAMETER1: 0000000000000000
EXCEPTION_PARAMETER2: 0000000000000009c
EXCEPTION_STR: 0xc0000005

STACK_TEXT:
fffffb0d`18d3ee60 fffff802`1df09152 : 00000000`00000000 00000000`e01f008d fffffb0d`18d3f202 fffff802`1e
fffffb0d`18d3f000 fffff802`1df0a3e9 : 00000000`00000000 00000000`00000010 00000000`00000000 ffff9a81`b5
fffffb0d`18d3f130 fffff802`1e14954f : 00000000`00000000 00000000`00000000 00000000`00000000 00000000`00
fffffb0d`18d3f260 fffff802`1e145d9b : ffff9a81`93735280 fffffb0d`18d3f5d0 00000000`00000000 00000000`00
fffffb0d`18d3f4d0 fffff802`1deb8fd0 : 00000000`000030f1 fffffb0d`18d3f790 ffff9a81`992cbb30 fffe409`b7
```



Use memory safe languages



Modern API Security

Q&A

**Carnegie
Mellon
University**
Software
Engineering
Institute