Bullish

How to thwart malicious automation and kick bot butt for $0

Randy Gingeleski
OWASP Global AppSec US 2021
11.11.2021

Agenda

whoami

Now

- ProdSec @ Bullish.com

Earlier

- ProdSec @ HBO Max
- Consulting and pen testing
- Aspect Security ❤️ OWASP

General

- Spends too much time online
- @gingeleski

Other = 🤺 + 🏹 + SF* + NYC

* South Florida

Disclaimer

Research credit goes out to many others.

I've just "**operationalized**."

Thank you to so many who open-source and share.

Disclaimer

Views and opinions are **strictly my own**

and do not represent those of my clients or employers, past or present, ad infinitum. 😇

The bots are coming 🤖

The bots are coming 🤖

```
185.88.103.161 - [24/Oct/2021:19:12:06 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
154.201.43.251 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
168.81.130.57 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
170.84.230.163 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
2605:6440:3003:1::2:82b4 - [24/Oct/2021:19:12:10 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
54.201.43.251 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
185.88.103.161 - [24/Oct/2021:19:12:13 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
185.88.103.161 - [24/Oct/2021:19:12:06 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
154.201.43.251 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
168.81.130.57 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
170.84.230.163 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
2605:6440:3003:1::2:82b4 - [24/Oct/2021:19:12:10 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
54.201.43.251 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
185.88.103.161 - [24/Oct/2021:19:12:13 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux:
185.88.103.161 - [24/Oct/2021:19:12:06 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
154.201.43.251 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
168.81.130.57 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
170.84.230.163 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
2605:6440:3003:1::2:82b4 - [24/Oct/2021:19:12:10 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
185.88.103.161 - [24/Oct/2021:19:12:06 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
154.201.43.251 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
168.81.130.57 - [24/Oct/2021:19:12:07 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
170.84.230.163 - [24/Oct/2021:19:12:09 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
2605:6440:3003:1::2:82b4 - [24/Oct/2021:19:12:10 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
160.116.245.18 - [24/Oct/2021:19:12:11 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
94.231.218.87 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
54.201.43.251 - [24/Oct/2021:19:12:12 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
185.88.103.161 - [24/Oct/2021:19:12:13 +0000] "POST /api/login HTTP/1.1" 401 17 "-" "Mozilla/5.0 (Linux;
```

Spin up any web app that (even looks like it) can support login

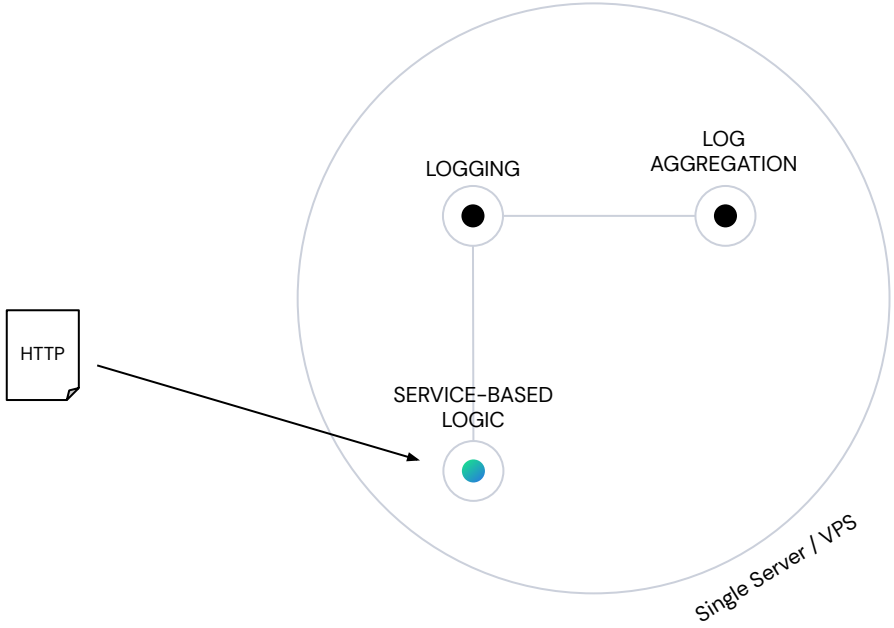
Watch your HTTP logs

You have sufficient logging, right?

https://github.com/yunginnanet/HellPot 👺

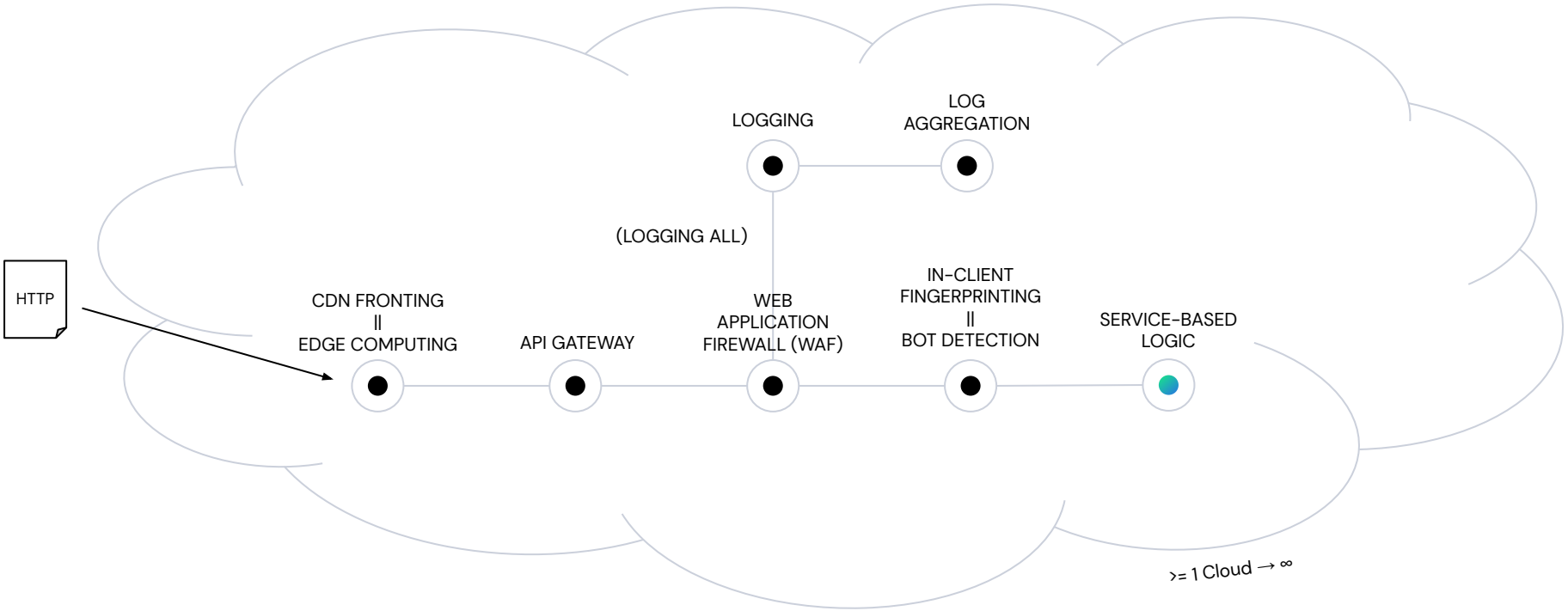
Layout of a modern web application

Layout of a modern web application



Classic and valid pattern 😉

Layout of a modern web application



Big enterprise doings 💰

Bot detection approaches

Bot detection approaches

1.
Good*

In-client fingerprinting

Open : <abrahamjuliot/creepjs>, <niespodd/browser-fingerprinting>, supercookie.me , <Cleafy/refingerprint>, <antoinevastel/fp-collect>, <fingerprintjs/fingerprintjs> + <fingerprintjs/BotD> …

"Free" → **$$** : hCaptcha, Google reCAPTCHA, Geetest …

2.
Better*

Network-based fingerprinting

Open : <NikolaiT/zardaxt>, <salesforce/ja3>, <FlUxIuS/pOf3plus>, <Ivan-Markovic/proxyCheck>, antoinevastel.com/bots/ip , getipintel.net , <z0ccc/locatejs>, <Umkus/ip-index> …

"Free" → **$$** : Cloudflare Bot Management, Google reCAPTCHA supplements …

3.
Best*

Behavioral analysis

Open : <prescience/dark-knowledge>, <IPL/fraud-detection-papers>, <safe-graph/DGFraud>, DIY risk scores …

"Free" → **$$** : Callsign, Kount, TypingDNA, Google reCAPTCHA supplements …

** Relative difficulty to adulterate (subjective!)*

In-client fingerprinting

What is it?

- Ask the browser or mobile application to execute (generally obfuscated) code, which will probe the frontend environment for data we later expect to receive on the backend.
- Look at hardware, device, and/or browser attributes as accessible in whatever sandbox environment the code might run in.

Notable open source works to help include …

- Abraham Juliot's **creepjs**, a library that facilitates "creepy device and browser fingerprinting."
  - https://abrahamjuliot.github.io/creepjs/
  - https://github.com/abrahamjuliot/creepjs
- Dariusz Niespodziany's **browser-fingerprinting**, an appropriately named web app and testing code for live browser fingerprinting.
  - https://niespodd.github.io/browser-fingerprinting/
  - https://github.com/niespodd/browser-fingerprinting
- Antoine Vastel's **bot-zoo**, a repository of centralized "bot" examples for testing purposes.
  - https://github.com/antoinevastel/bots-zoo
- **FingerprintJS**, a "browser fingerprinting library with the highest accuracy" + their bot detector. ***Some freemium features so borders on paid***
  - https://github.com/fingerprintjs/fingerprintjs/
  - https://github.com/fingerprintjs/BotD
- More → supercookie.me , <Cleafy/refingerprint>, <antoinevastel/fp-collect>

"Free" to $$ vendor tech includes **hCaptcha**, **Google reCAPTCHA**, **Geetest** …

In-client fingerprinting

appropriate key based on your use cases.

Comparison category	Score-based site key (Recommended)	Checkbox site key
Description	Score-based site keys let you verify whether an interaction is legitimate without any user interaction.	Checkbox site keys use a checkbox challenge that requires user interaction to verify that the user is not a robot. Also, you can use checkbox site keys to protect specific actions with CAPTCHA challenges. ★ We do not recommend using checkbox site keys because they increase user friction and don't significantly improve accuracy. For more information, see the FAQs.
How it works	With score-based site keys, the reCAPTCHA Enterprise API returns a score, which you can use to take action in the context of your site. Examples of actions you might take include requiring additional factors of authentication, sending a post to	A checkbox site key renders an **I'm not a robot** checkbox that a user must click to verify that they're not a robot. This checkbox site key might or might not challenge them with CAPTCHA challenges. In both cases, the reCAPTCHA Enterprise API returns a score.

https://cloud.google.com/recaptcha-enterprise/docs/choose-key-type

In-client fingerprinting

hCaptcha

Plans | Enterprise | Docs | Labeling Services | Login | Signup

Great User Experience

For Enterprise users, options are available to completely avoid challenges in most scenarios.

Reduce friction by avoiding a challenge entirely for 99.9% of your legitimate traffic while still providing excellent security, thanks to advanced bot models.

https://hcaptcha.com/enterprise https://abrahamjuliot.github.io/creepjs/

In-client fingerprinting

https://abrahamjuliot.github.io/creepjs/

Your ID: 9f18386e

fingerprints renewed 8/27/2021

Browser
trust score: 0% F-
visits: 2
first: 11/1/2021, 9:34:01 AM
last: 11/7/2021, 12:24:23 PM
persistence: 147.8 hours

has trash: false
has lied: true -279%
has errors: false
loose fingerprint: 004448b8
loose switched: 1x +20% reward
bot: 0%

add a signature to your fingerprint | Sign

Session ID d3620e0b
Session Loads 1
Session Switched none

Trash
gathered (0):

Lies 6d6e5e46
unmasked (9): details

Errors
captured (0):

WebRTC cf0edf66
165.225.222.90
165.225.222.90
165.225.222.90
type: srflx
foundation: 842163049
protocol: udp
codecs: 7c2c2130
codecs sdp: e90c5928

Timezone d7dab170
Eastern Standard Time
America, New York
-27028667038000
300

Intl 4f83ab37
en-US
July, Eastern Daylight Time
American English
21 million
next year
0 or 1
one

crowd-blending score: 100% A

Prediction
*Intel Mac OS X 10

pending review: 14

Headless a857b084
chromium: true
29% like headless: 43dca615
0% headless: 29be19c4
0% stealth: 6c149d26

Resistance aa9bca22
privacy: unknown
security: unknown
mode: unknown
extension: d488a24a

ServiceWorkerGlobalScope

Worker 938ce827
device:

https://abrahamjuliot.github.io/creepjs/

Network fingerprinting

What is it?

- Scrutinize network data like IP address, finer TCP/IP characteristics, HTTP request velocity, etcetera.
  - i.e. **compare this connection's MTU : MSS ratio a standard table of values, flag mismatches** ☠️
- Requires execution *at* or *very close to* the edge of our infrastructure.
  - Further away from edge = **lower data integrity**

Notable open source works to help include …

- Salesforce's **ja3**, "a method for creating SSL/TLS client fingerprints that should be easy to produce on any platform and can be easily shared for threat intelligence."
  - https://github.com/salesforce/ja3
- Nikolai Tschacher's **zardaxt**, a "passive TCP/IP fingerprinting tool" to "find out what Operating Systems your clients are really using."
  - https://github.com/NikolaiT/zardaxt
  - https://incolumitas.com/ → great blog 💯
- Antoine Vastel's **Bot IPs API**, "a free API to get information about IP addresses used by bots."
  - https://antoinevastel.com/bots/ui-ip
- **IP Intelligence**, "a service that determines how likely an IP address is a proxy / VPN / bad IP using advanced mathematical and modern computing techniques."
  - https://getipintel.net
- More → <FlUxlu5/p0f3plus>, <Ivan-Markovic/proxyCheck>, <z0ccc/locatejs>, <Umkus/ip-index>

"Free" to $$ vendor tech includes **Cloudflare Bot Management**, **Google reCAPTCHA supplements** …

Behavioral analysis

What is it?

- Look at input-derived biometrics, action velocity, etcetera
- Dependent on observing a minimum amount of user activity before acting
  - More direct and/or anonymous (unauthenticated) a user can be = **less accuracy**

Notable open source works to help include …

- Prescience's **dark-knowledge**, "papers and presentations for counter-detection and web privacy enthusiasts."
  - https://github.com/prescience-data/dark-knowledge
- Xinyu Wang's **fraud-detection-papers**, "a collection of research and survey papers of fraud detection[,] mainly in advertising [but does get into anomaly detection!]."
  - https://github.com/IPL/fraud-detection-papers
- SafeGraph's **DGFraud**, "a deep graph-based toolbox for fraud detection."
  - https://github.com/safe-graph/DGFraud
- Benedek Rozemberczki's **awesome-fraud-detection-papers**, "a curated list of data mining papers about fraud detection."
  - https://github.com/benedekrozemberczki/awesome-fraud-detection-papers
- AWS "Fraud Detection Using Machine Learning" lab setup.
  - https://github.com/awslabs/fraud-detection-using-machine-learning

… but this category is where you should consider DIY. 🛠️

"Free" to $$ vendor tech includes **Callsign**, **TypingDNA**, **Kount**, **Google reCAPTCHA supplements** …

Threat profiling

Threat profiling

What are bad bots trying to accomplish?

Login attacks

Payment attacks

Destructive or for-ransom attacks

Content scraping

Threat profiling

What does the biggest, baddest attacker look like?

A whole lot of …

- **Time and/or $$$**
  - Enablers of the following additional points
- **Network and/or device resources**
  - Residential proxies
  - Device farms
  - Private botnet
    - Network of compromised computers/devices
- **Attack fodder**
  - Freshly compromised credentials → login attacks
  - Manual workers or labor farm credits → full attack behavior or CAPTCHA solving
- **Understanding of *your* application and/or "normal" traffic**
  - "Act natural"
- **Programming skills or resources**
  - No OpenBullet "script kiddies"
  - More custom = more effective

Tactical maneuvers

Tactical maneuvers

Ad hoc blocks

Statically block unwanted IP addresses, subnets, ASNs, usernames, request header orders, TLS and/or other device fingerprints, "WAF patterns" ...

Adaptive blocks

Dynamically labeling IP addresses, subnets, ASNs, usernames, other identifiers as good/bad, by region, traffic type (VPN, proxy, etc.) ...

Rate limits

Measuring and then blocking based on observed velocity from an IP address, session, username, email, payment method, other unique identifiers.

In-client bot detection

Fingerprinting and/or CAPTCHA followed by scrutiny on the backend.

Bot traps

Missing bot detection payloads from the client, honeypots (via hostname, paths, or fields), Client Puzzle Protocol (i.e. Hashcash) ...

Fluffing up your blocks

Obvious HTTP blocks → tarpit, spoof, or random confusion.

Fool + frustrate attackers. 😉

Non-block actions

Various challenges, locks, logging, partial blocks, flags, waiting rooms, adjustments, notifications, user requests, limits ...

Product security

Magic links*, WebAuthn, multi-factor authentication (MFA), email verification, one-time passcodes (OTP), device and/or network anomaly detection ...

* or magic.link !

Implementation patterns

Implementation patterns

Fairly immature

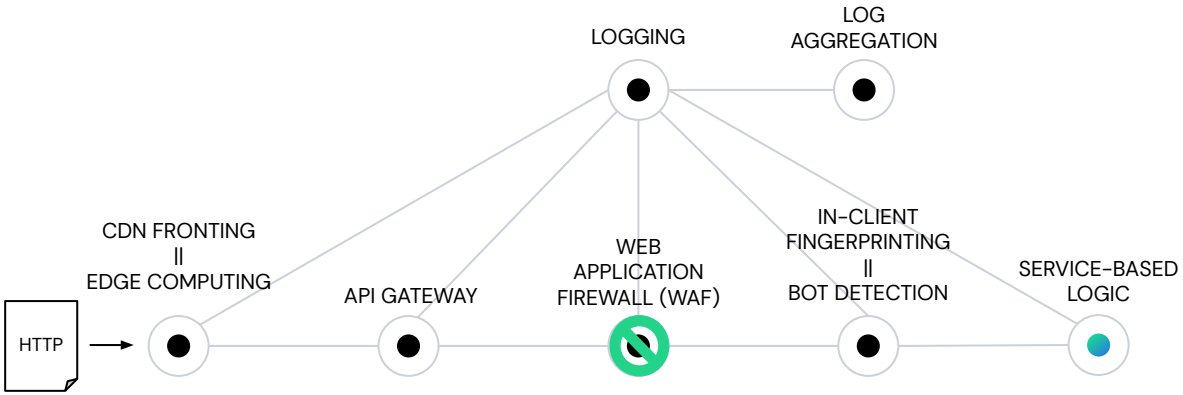
Fairly mature

“One heavy lifter near origin.”

“Edge it and forget it.”

“Manage at origin, move to edge.”

Implementation patterns

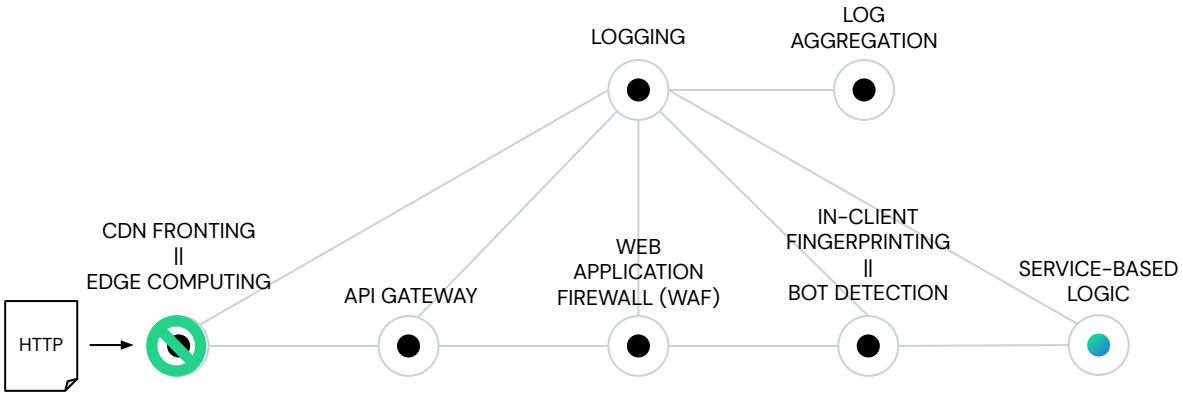


"One heavy lifter near origin."

"Edge it and forget it."

"Manage at origin, move to edge."

Implementation patterns

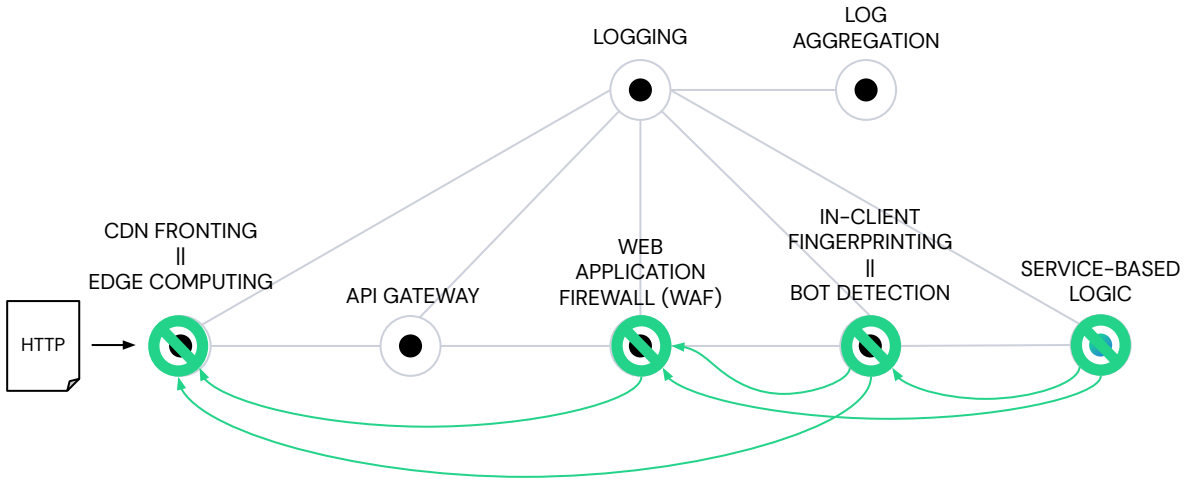


"One heavy lifter near origin."

"Edge it and forget it."

"Manage at origin, move to edge."

Implementation patterns



"One heavy lifter near origin."

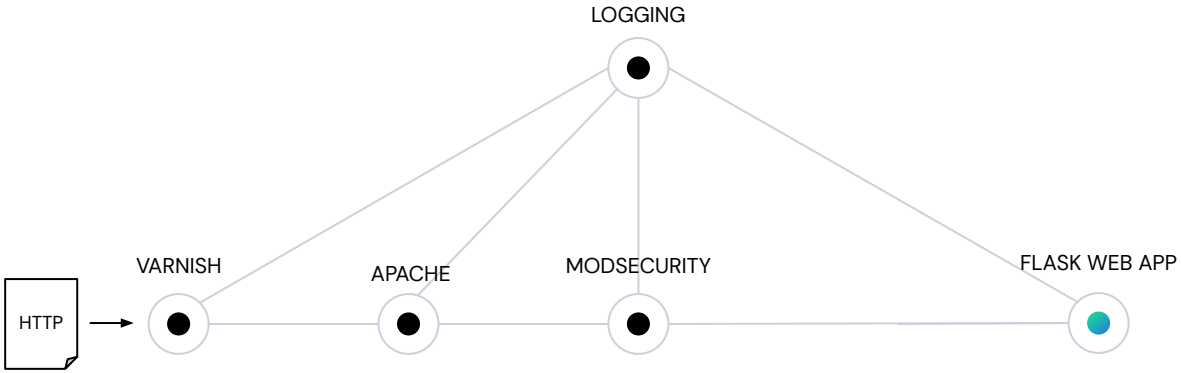
"Edge it and forget it."

"Manage at origin, move to edge."

Demo

Demo

https://warandcode.com/2021-global-appsec



406s to 401s via Varnish config

Continuing on

Continuing on

Risk scoring

Vendor-agnostic

Product security

Ever-evolving field

Bullish

THANK YOU 🙂

randy.gingeleski@owasp.org 📩
gingeleski.com (general)
warandcode.com (infosec)

We're hiring ❤️ bullish.com/careers