AMERICAS DevSecCon

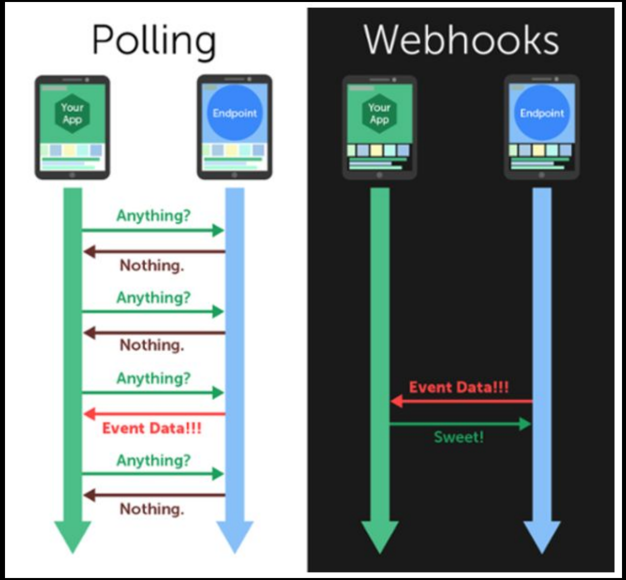
Simple ways to make webhook security better

Fred

peace, love and good software @ ngrok

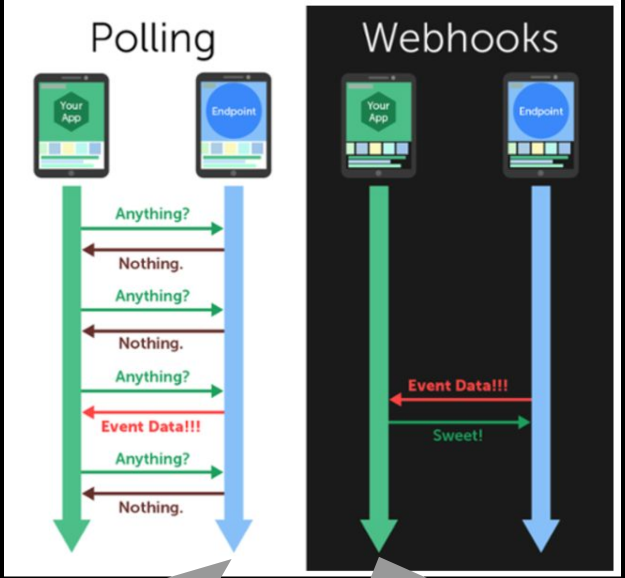
Webhooks

You see them
You ❤️ them



Caveat 1

Most security responsibilities on the listener

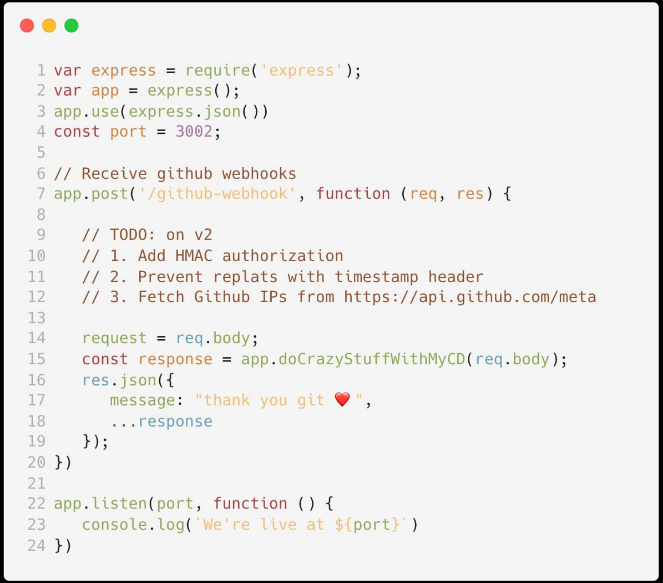


Confidentiality: Authn, Authz, **Integrity:** Message Integrity, Queueing
Availability: DDoS

Confidentiality: Authn, Authz, **Integrity:** Message Integrity, Queueing
Availability: DDoS

Caveat 2

Security doesn't block success



```
var express = require('express');
var app = express();
app.use(express.json())
const port = 3002;

// Receive github webhooks
app.post('/github-webhook', function (req, res) {

   // TODO: on v2
   // 1. Add HMAC authorization
   // 2. Prevent replats with timestamp header
   // 3. Fetch Github IPs from https://api.github.com/meta

   request = req.body;
   const response = app.doCrazyStuffWithMyCD(req.body);
   res.json({
      message: "thank you git ❤️",
      ...response
   });
})

app.listen(port, function () {
   console.log(`We're live at ${port}`)
})
```

Caveat 3

Lots of different ways to secure webhooks!

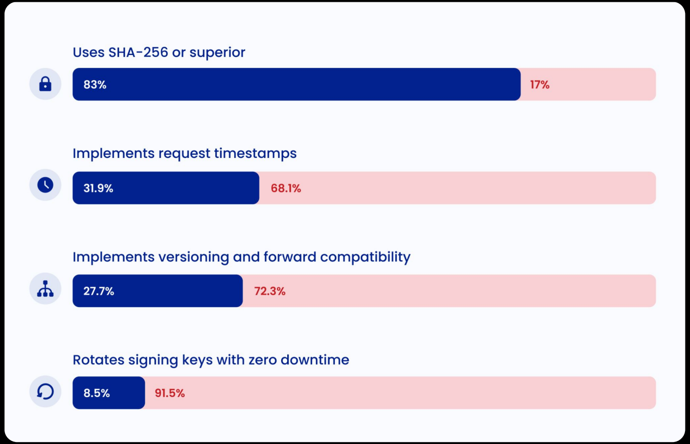
After seeing
100+ webhooks

1. Implementations + challenges

TL;DR

7 of 10 webhooks will present differences

4% of webhooks implement complete controls



https://blog.ngrok.com/posts/get-webhooks-secure-it-depends-a-field-guide-to-webhook-security

1. Implementations + challenges

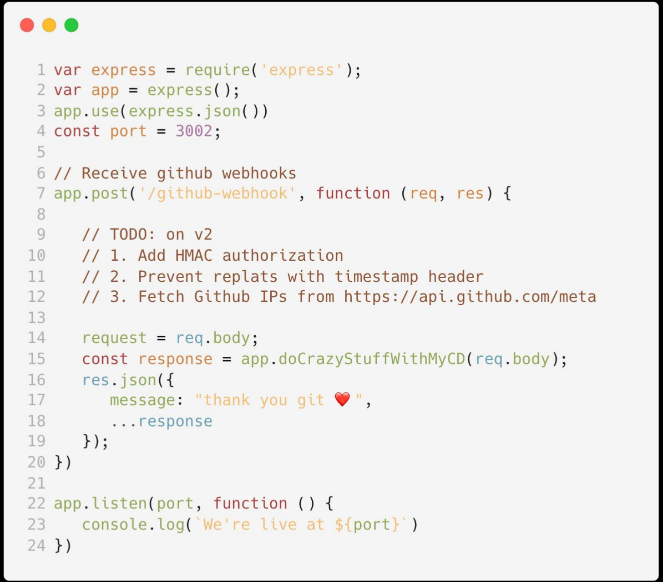
Responsible developers

Tasks:

- read a bunch of docs

- implement beyond the happy path

- don't take it for granted



```
var express = require('express');
var app = express();
app.use(express.json())
const port = 3002;

// Receive github webhooks
app.post('/github-webhook', function (req, res) {

   // TODO: on v2
   // 1. Add HMAC authorization
   // 2. Prevent replats with timestamp header
   // 3. Fetch Github IPs from https://api.github.com/meta

   request = req.body;
   const response = app.doCrazyStuffWithMyCD(req.body);
   res.json({
      message: "thank you git ❤️",
      ...response
   });
})

app.listen(port, function () {
   console.log(`We're live at ${port}`)
})
```

https://blog.ngrok.com/posts/get-webhooks-secure-it-depends-a-field-guide-to-webhook-security

How we can fix this?

catalog!

Webhook Providers

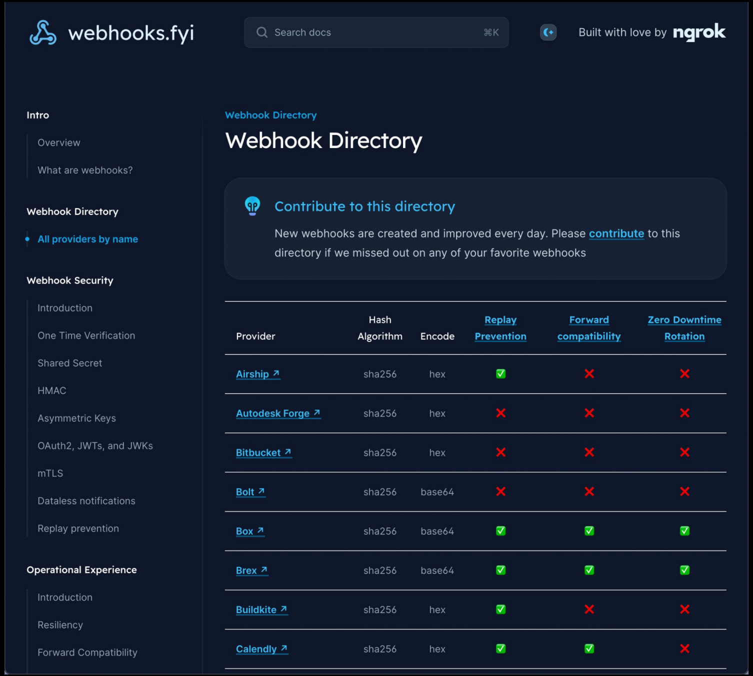
Best Practices

- Provide amazing documentation
- Implement security on egress
- Improve secret keys
- Use strong Encryption & hashing

- Leverage Signature Payload
- Replay Prevention
- Versioning
- Add compensatory controls

Webhook Providers

Easier: Copy the Greats!



webhooks.fyi

Search docs ⌘K

Built with love by ngrok

Intro

- Overview
- What are webhooks?

Webhook Directory

- All providers by name

Webhook Security

- Introduction
- One Time Verification
- Shared Secret
- HMAC
- Asymmetric Keys
- OAuth2, JWTs, and JWKs
- mTLS
- Dataless notifications
- Replay prevention

Operational Experience

- Introduction
- Resiliency
- Forward Compatibility

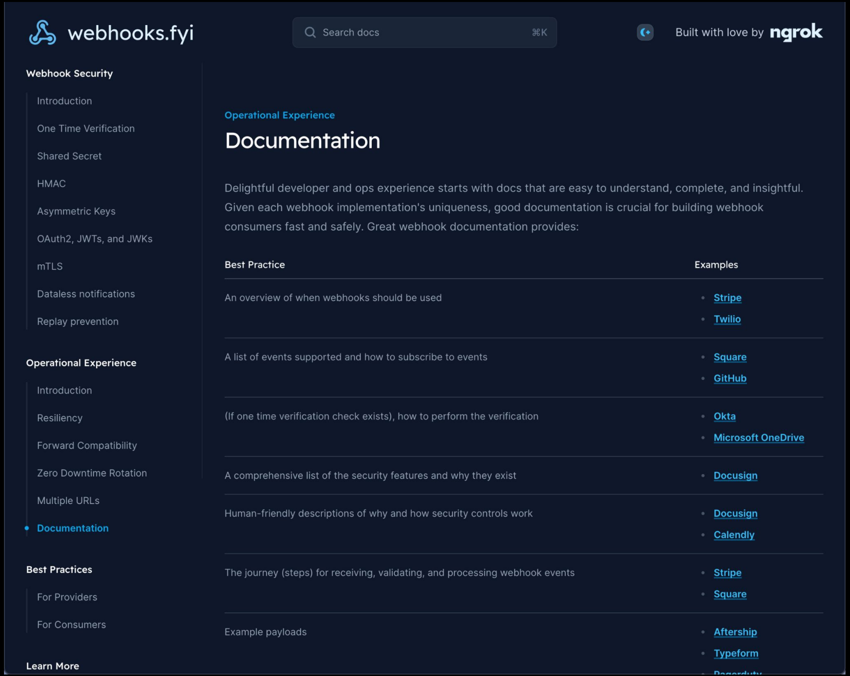
Webhook Directory

Webhook Directory

Contribute to this directory

New webhooks are created and improved every day. Please contribute to this directory if we missed out on any of your favorite webhooks

Provider	Hash Algorithm	Encode	Replay Prevention	Forward compatibility	Zero Downtime Rotation
Airship ↗	sha256	hex	✅	❌	❌
Autodesk Forge ↗	sha256	hex	❌	❌	❌
Bitbucket ↗	sha256	hex	❌	❌	❌
Bolt ↗	sha256	base64	❌	❌	❌
Box ↗	sha256	base64	✅	✅	✅
Brex ↗	sha256	base64	✅	✅	✅
Buildkite ↗	sha256	hex	✅	❌	❌
Calendly ↗	sha256	hex	✅	✅	❌



webhooks.fyi

Search docs ⌘K

Built with love by ngrok

Webhook Security

- Introduction
- One Time Verification
- Shared Secret
- HMAC
- Asymmetric Keys
- OAuth2, JWTs, and JWKs
- mTLS
- Dataless notifications
- Replay prevention

Operational Experience

- Introduction
- Resiliency
- Forward Compatibility
- Zero Downtime Rotation
- Multiple URLs
- Documentation

Best Practices

- For Providers
- For Consumers

Learn More

Operational Experience

Documentation

Delightful developer and ops experience starts with docs that are easy to understand, complete, and insightful. Given each webhook implementation's uniqueness, good documentation is crucial for building webhook consumers fast and safely. Great webhook documentation provides:

Best Practice	Examples
An overview of when webhooks should be used	Stripe Twilio
A list of events supported and how to subscribe to events	Square GitHub
(If one time verification check exists), how to perform the verification	Okta Microsoft OneDrive
A comprehensive list of the security features and why they exist	Docusign
Human-friendly descriptions of why and how security controls work	Docusign Calendly
The journey (steps) for receiving, validating, and processing webhook events	Stripe Square
Example payloads	Aftership Typeform Pagerduty

Webhook Listeners

Best Practices

- Use HTTPS with a strong ciphers
- Ensure you're using security
- Restrict requests by IP
- Storing secrets
- Segmenting secrets
- Rotating secrets
- Use robust signature algorithms
- Call back the service

Webhook Listeners

Easier: Learn from your Web App/API

- Use HTTPS with a strong ciphers
- Ensure you're using security
- Restrict requests by IP
- Storing secrets
- Segmenting secrets
- Rotating secrets
- Use robust signature algorithms
- Call back the service

As an Individual / Industry

Some standards

IETF HTTP Message Signatures

Spec for Signing HTTP messages
Applicable to webhooks
Part of the IETF Extensions Working Group

OpenID's Shared Signals and Events (SSE)

Establishing a security framework for event notification.
Focus on security solutions exchanging info.
Relies heavily on webhooks as proto for events.

CloudEvents

Specification for standardizing event data.
The specification includes webhooks.
Simplifying event declaration and delivery across systems.
Active effort at Cloud Native Computing Foundation (CNCF).

REST Hooks

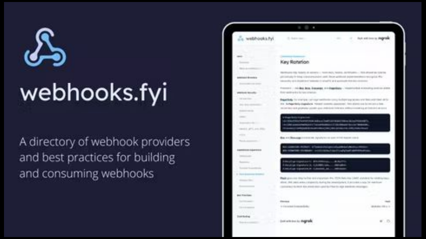
REST Hooks are an initiative ran by Zapier from 2013-2017.
Goal was to create a collection of patterns for treating webhooks
like subscriptions with a minimum implementation walkthrough.

As an Individual / Industry

Easy-ish: Build Awareness

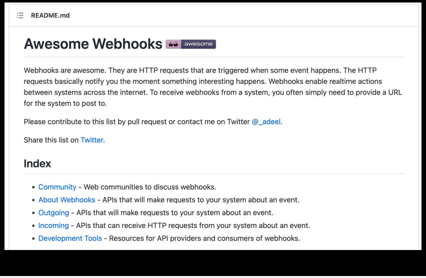
As an Individual / Industry

Ways to help us



A directory of webhook providers and best practices for building and consuming webhooks

https://webhooks.fyi



README.md

Awesome Webhooks awesome

Webhooks are awesome. They are HTTP requests that are triggered when some event happens. The HTTP requests basically notify you the moment something interesting happens. Webhooks enable realtime actions between systems across the internet. To receive webhooks from a system, you often simply need to provide a URL for the system to post to.

Please contribute to this list by pull request or contact me on Twitter @_adeel.

Share this list on Twitter.

Index

- Community - Web communities to discuss webhooks.
- About Webhooks - APIs that will make requests to your system about an event.
- Outgoing - APIs that will make requests to your system about an event.
- Incoming - APIs that can receive HTTP requests from your system about an event.
- Development Tools - Resources for API providers and consumers of webhooks.

https://github.com/realadeel/awesome-webhooks

Read, Contribute, Star, Share
List your implementation or a provider you know

Thank you ❤️

@sudobinbash
@ngrokHQ