Securing Serverless - By Breaking In

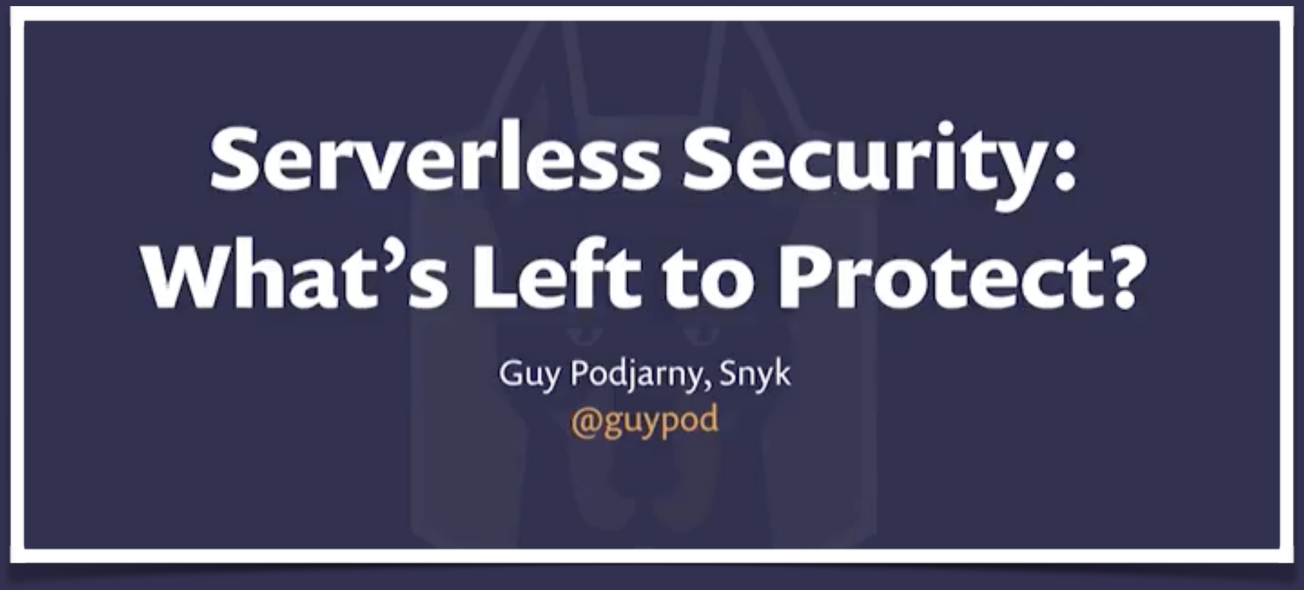
Guy Podjarny, Snyk
@guypod

About Me

- Guy Podjarny, **@guypod** on Twitter
- CEO & Co-founder at Snyk
- History:
  - Cyber Security part of Israel Defense Forces
  - First Web App Firewall (AppShield), Dynamic/Static Tester (AppScan)
  - **Security**: Worked in Sanctum -> Watchfire -> IBM
  - **Performance**: Founded Blaze -> CTO @Akamai
- O'Reilly author, speaker

Serverless Security: The Theory

(talk from ServerlessConf)



https://snyk.io/blog/serverless-security-implications-from-infra-to-owasp/

https://www.youtube.com/watch?v=CiyUD_rI8D8

Today - straight to practice!

Agenda

- Show a demo serverless app
- Hack it
- Explain the security flaws and how to fix them
- Summary
- Q&A

Going Terminal…

Vulnerable Libraries

Example: Fetch file & store in s3 (Serverless Framework Example)

```
'use strict';

const fetch = require('node-fetch');
const AWS = require('aws-sdk'); // eslint-disable-line import/no-extraneous-dependencies

const s3 = new AWS.S3();

module.exports.save = (event, context, callback) => {
  fetch(event.image_url)
    .then((response) => {
      if (response.ok) {
        return response;
      }
      return Promise.reject(new Error(
            `Failed to fetch ${response.url}: ${response.status} ${response.statusText}`));
    })
    .then(response => response.buffer())
    .then(buffer => (
      s3.putObject({
        Bucket: process.env.BUCKET,
        Key: event.key,
        Body: buffer,
      }).promise()
    ))
    .then(v => callback(null, v), callback);
};
```

19 Lines of Code

```
"dependencies": {
  "aws-sdk": "^2.7.9",
  "node-fetch": "^1.6.3"
}
```

2 Direct dependencies

19 dependencies (incl. indirect)

191,155 Lines of Code

EQUIFAX DATA BREACH

Equifax's Mega-Breach Was Made Possible by a Website Flaw It Could Have Fixed

Security

Equifax's disastrous Struts patching blunder: THOUSANDS of other orgs did it too

Those are just the ones known to have downloaded outdated versions

Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/13/2017, 11:12 PM

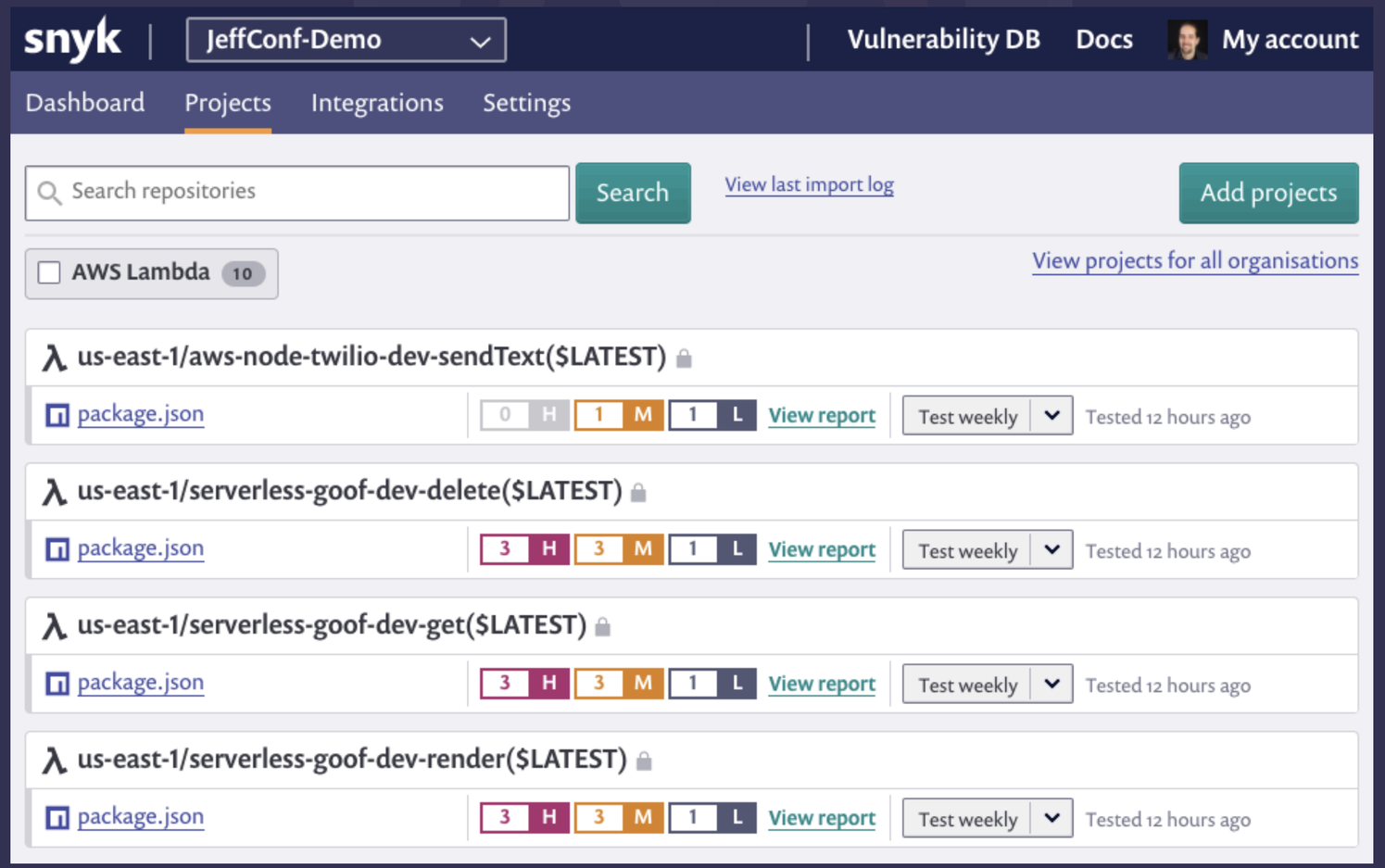
Serverless does secure

OS dependencies

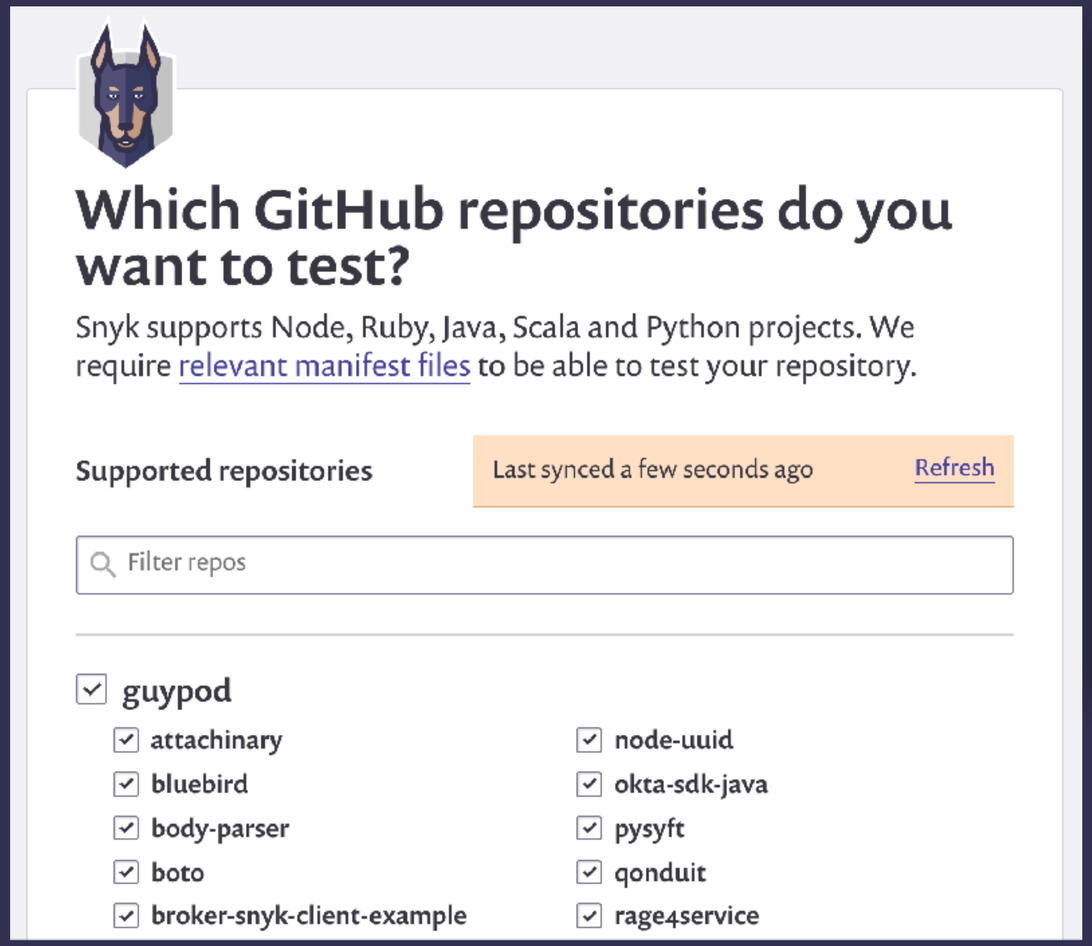
Just not app dependencies

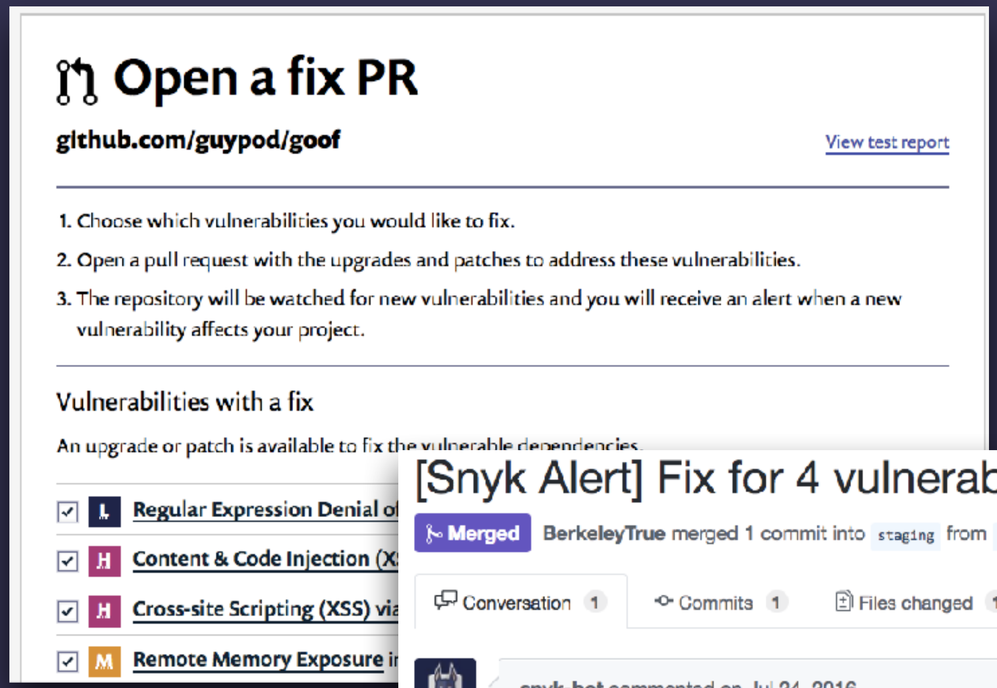
1. Beware Vulnerable Libraries

(test during dev, monitor over time)



Side Note:
Snyk isn't only for Serverless





snyk.io

Denial of Service

2. ReDoS can still be costly

(won't take you down, but can hike up bill)

Beware
Resource Exhaustion Attacks

Not all your services elastically scale

Secrets

3. Avoid secrets in deployed code

(env variables aren't enough - Use a KMS!)

```
const adminSecret = "ea29cbdb-a562-442a-8cc2-adbc6081d67c";

module.exports.api = (event, context, callback) => {

  if (!event.queryStringParameters ||
    !event.queryStringParameters.secret ||
    event.queryStringParameters.secret != adminSecret) {
      // Return an unauthorized response
```

Serverless platforms offer a Key Management System

Just use it!

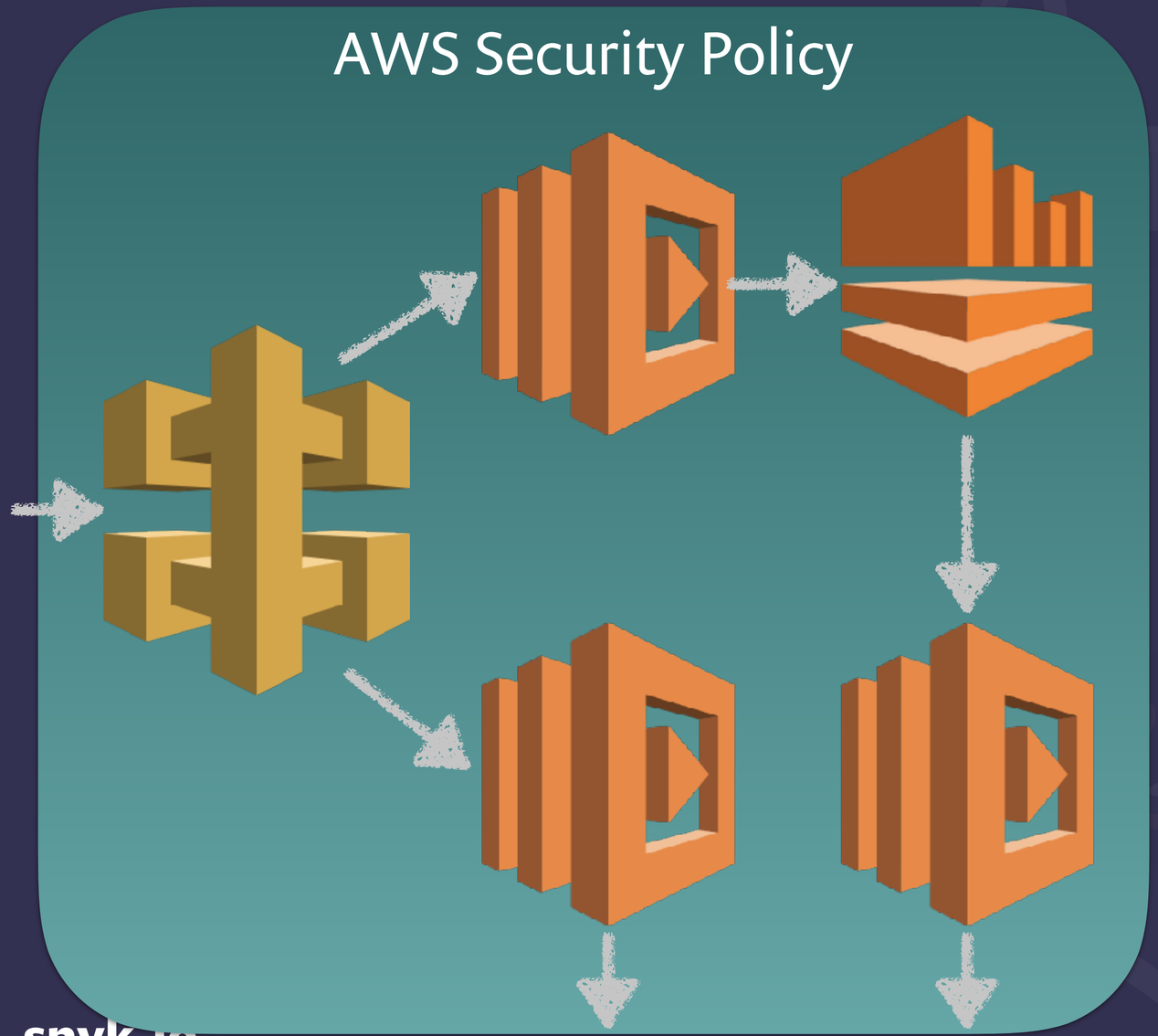
Granularity

4. Deploy granular functions

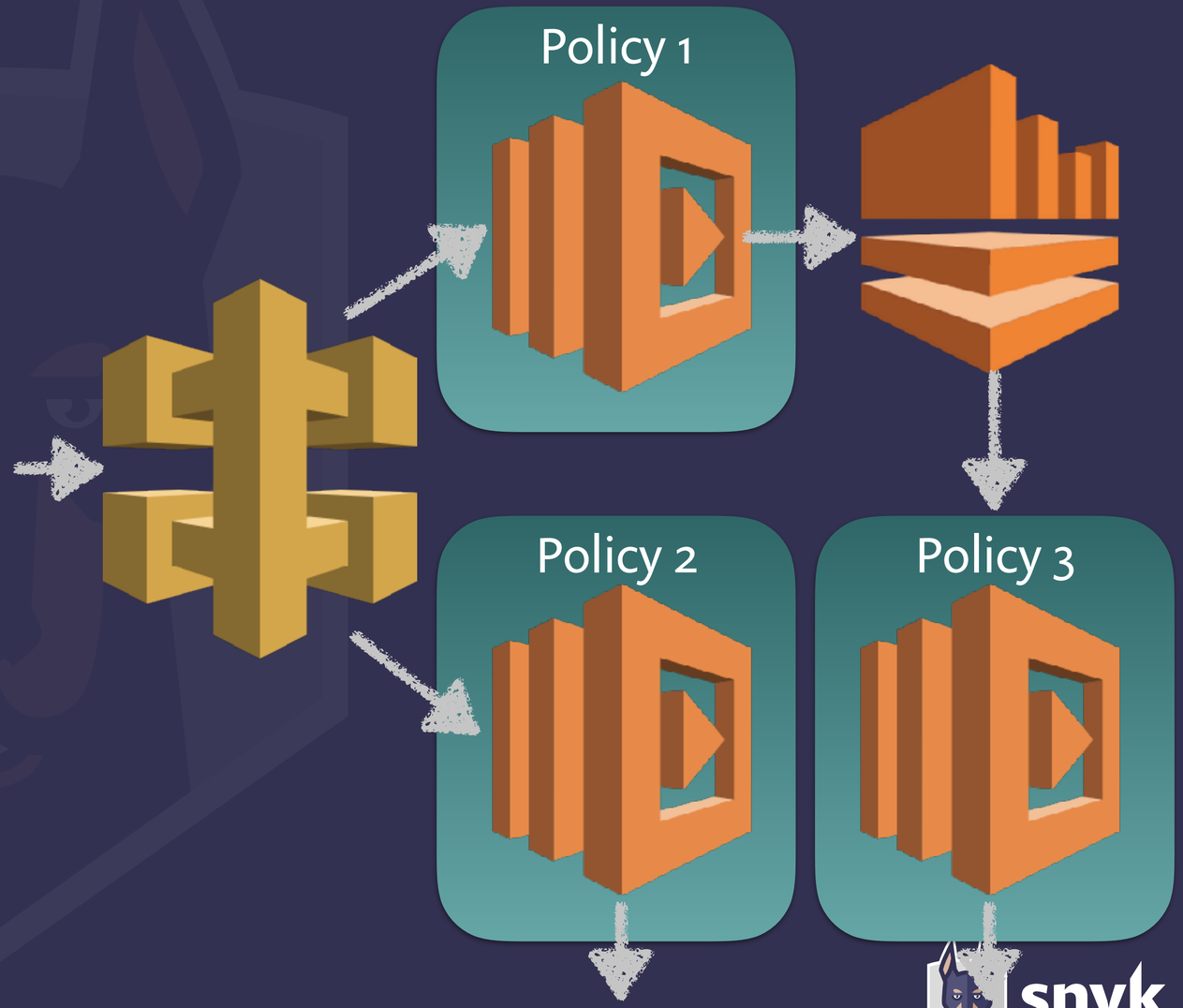
(shared function code = greater exposure)

```
functions:
  create: serverless-goof-dev-create
  list: serverless-goof-dev-list
  render: serverless-goof-dev-render
  get: serverless-goof-dev-get
  update: serverless-goof-dev-update
  delete: serverless-goof-dev-delete
  internalBackup: serverless-goof-dev-internalBackup
  internalRestore: serverless-goof-dev-internalRestore
  adminApi: serverless-goof-dev-adminApi
```

Easier



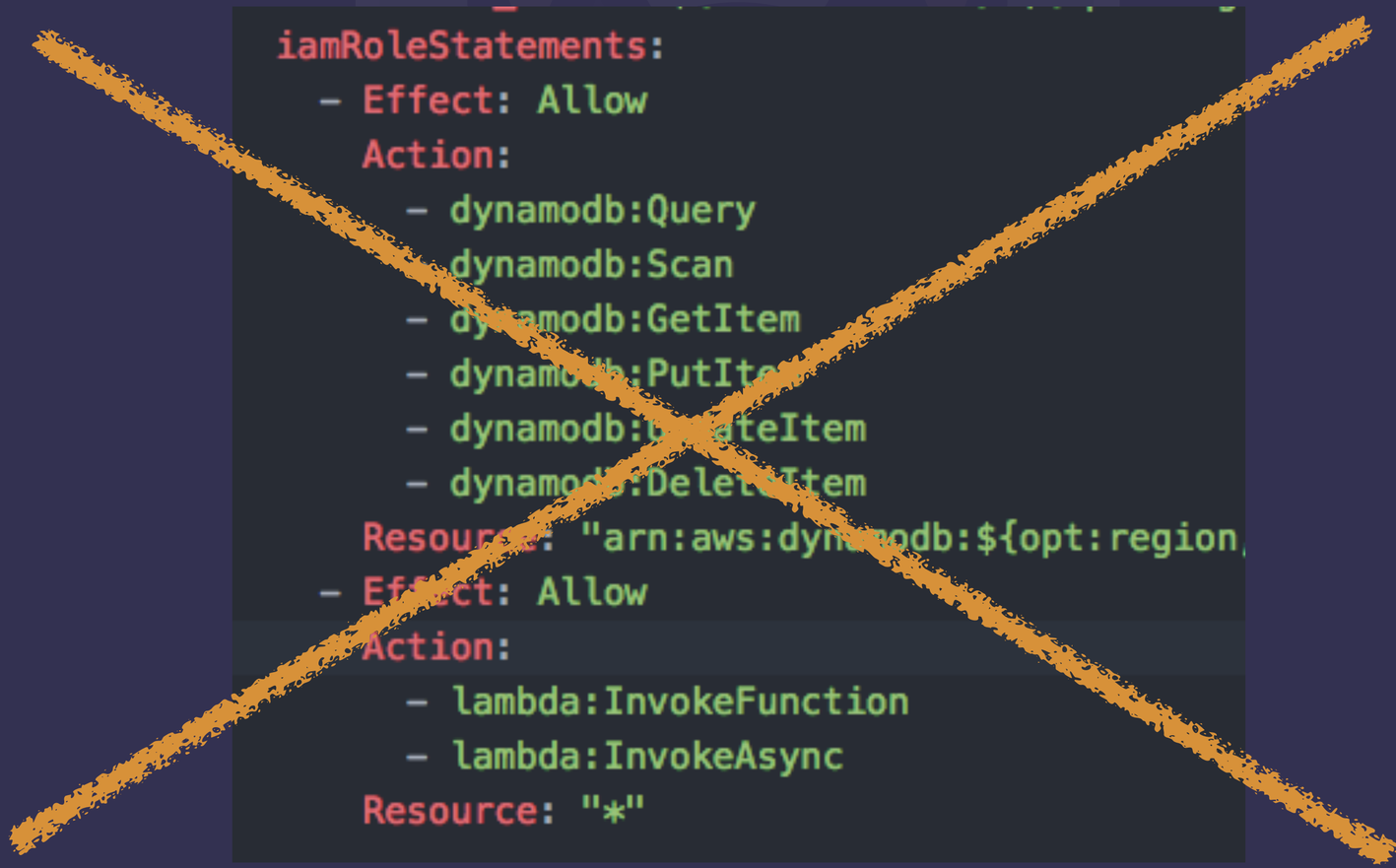
Safer



Permissions

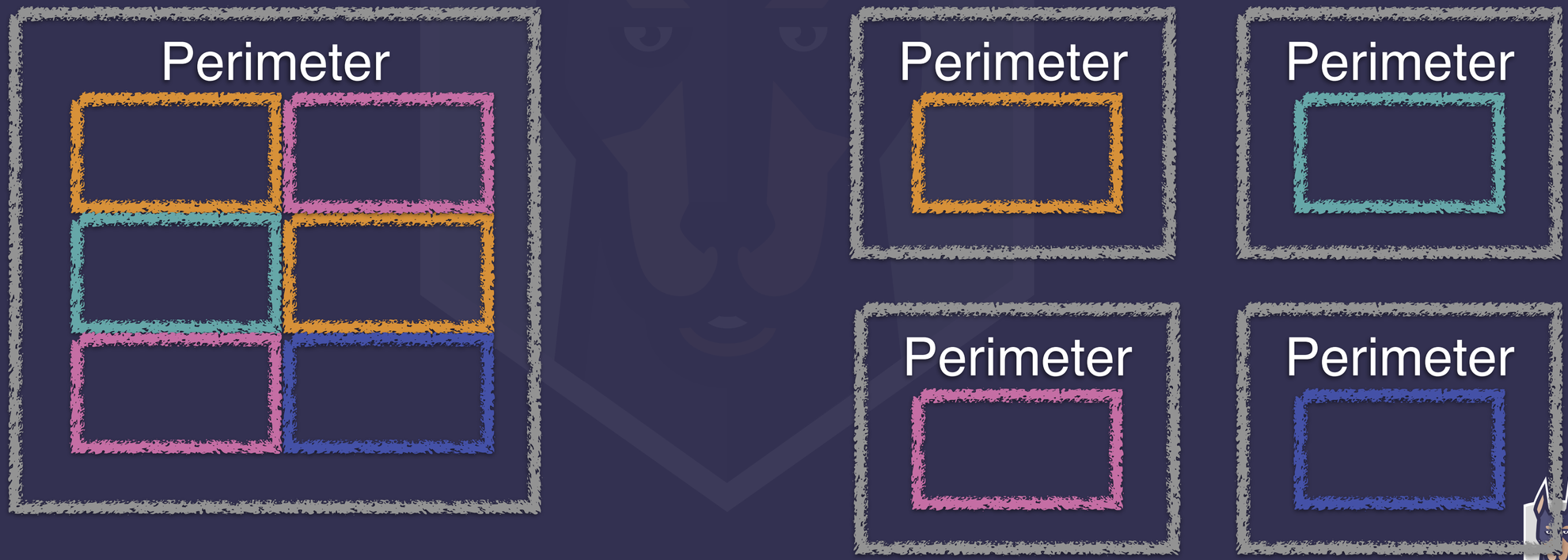
5. Use Granular Policies

(only allow each function its minimum permissions)



A function is a perimeter

That needs to be secured



Immutability

6. Don’t rely on immutability

(Lambda - and others - reuse servers)

```
total 20
-rw-rw-r-- 1 sbx_user1060 486 1859 Jul  6 21:41 goof-todos.0.3685130574740469 5
-rw-rw-r-- 1 sbx_user1060 486 1859 Jul  6 21:41 goof-todos.0.4484891316387802 4
-rw-rw-r-- 1 sbx_user1060 486 1859 Jul  6 21:41 goof-todos.0.4663676058407873
-rw-rw-r-- 1 sbx_user1060 486 1859 Jul  6 21:41 goof-todos.0.581618001218885 2
-rw-rw-r-- 1 sbx_user1060 486 1859 Jul  6 21:41 goof-todos.0.7717393657658249
```

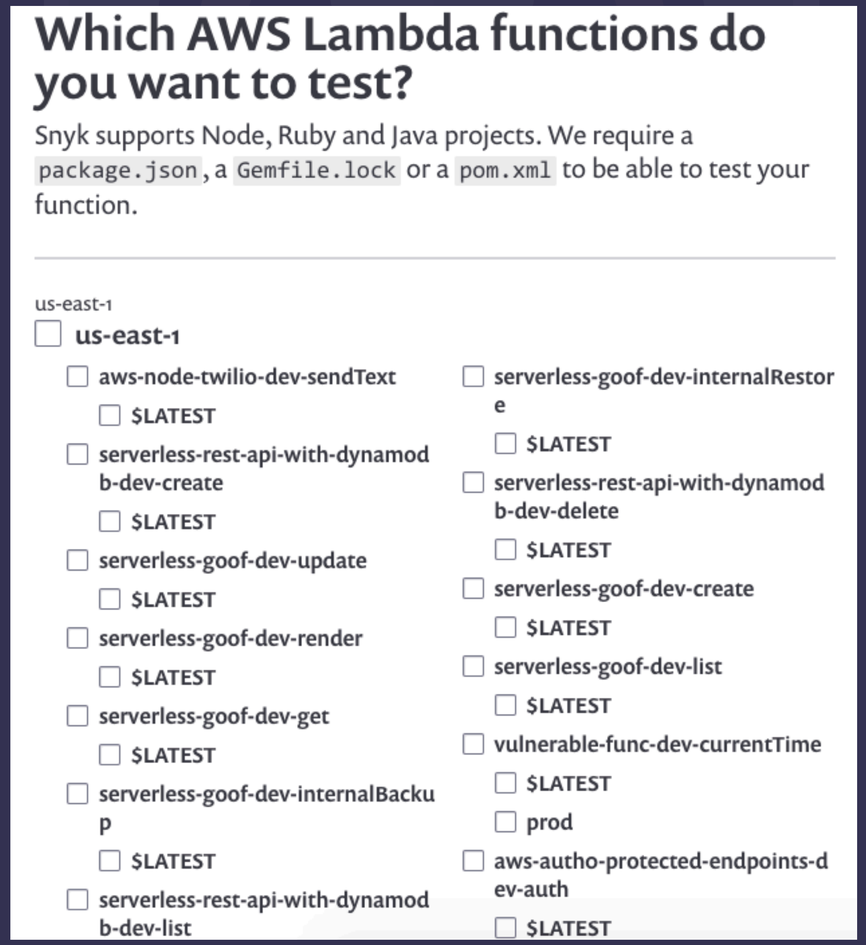
Serverless user is typically

Low Privilege

Reducing impact substantially, but not eliminating it

7. Worry about *all* functions

(Every available function increases your attack surface)



Security in Serverless

Better	Neutral	Worse
Vulnerable OS Dependencies	Permissions	Third Party Services
Denial of Service	Securing Data at rest	Attack Surface
Long-lived Compromised Servers	Vulnerabilities in your code	Security Monitoring
	Vulnerable App Dependencies	

Serverless is defined now.
Let's build Security in.

Thank You!

Don't forget:
OSS Security AMA
2:55pm, Waterfront CDE

Guy Podjarny, Snyk
@guypod