Serverless Security
New Risks Require New Approach

Itay Rozenman | Sr Director
Cloud Native Engineering

CONTRAST SECURITY

```
user@host:~
```

Itay Rozenman

CONTRAST SECURITY

cloudessence

CisCo

CloudLock

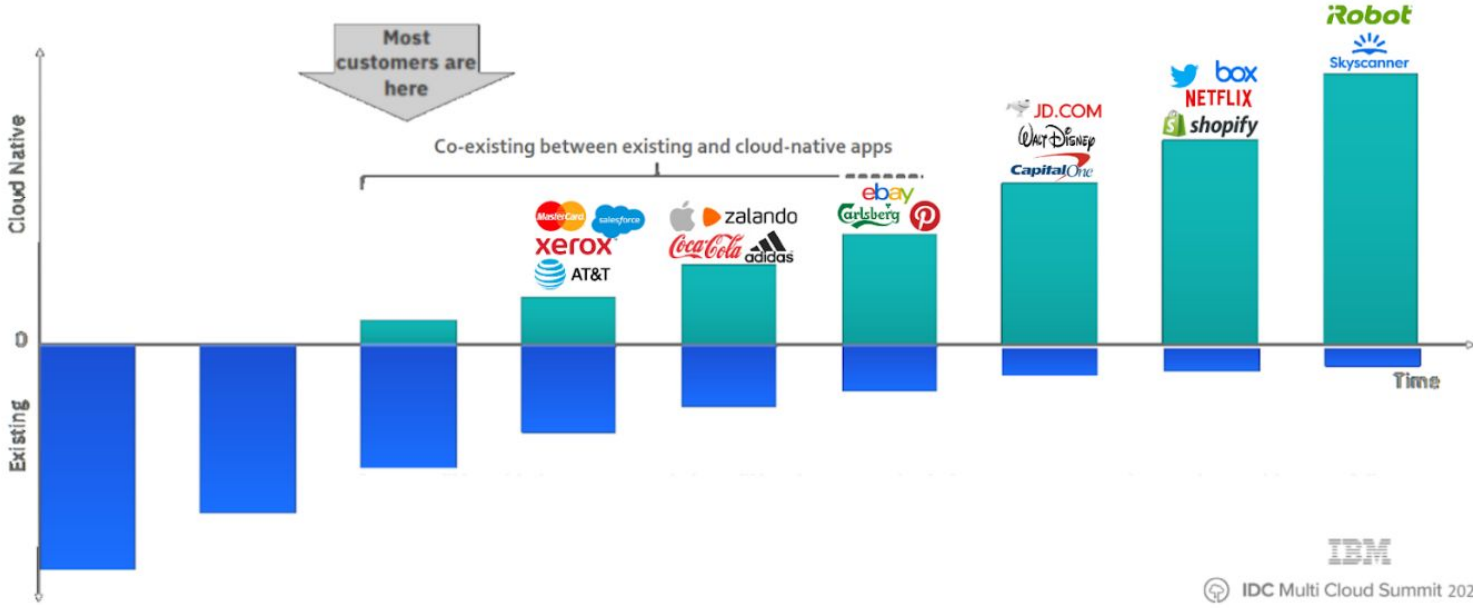
Cisco Cloudlock

Cloud Native is the **Future** of **Application development**

By 2021, **70% of newly developed enterprise applications will be developed as cloud-native applications**, based on a hyper-agile architecture, which utilizes containers, microservices, and serverless functions.

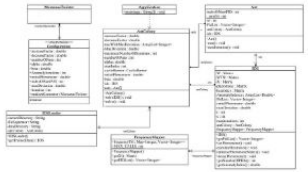
IDC
ANALYZE THE FUTURE

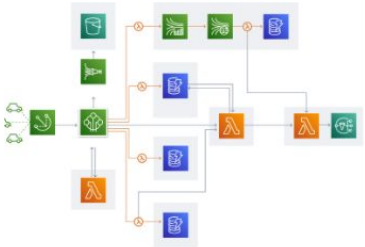
The Cloud Native Transformation Has Begun



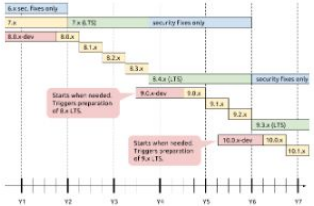
More Than A Technology Shift

Architecture

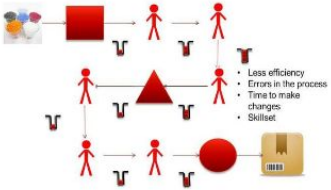




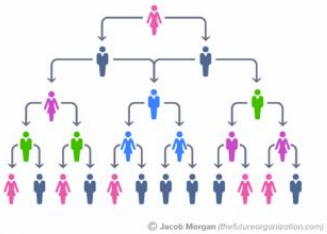
Cycles

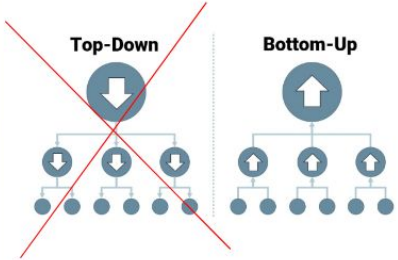


Process

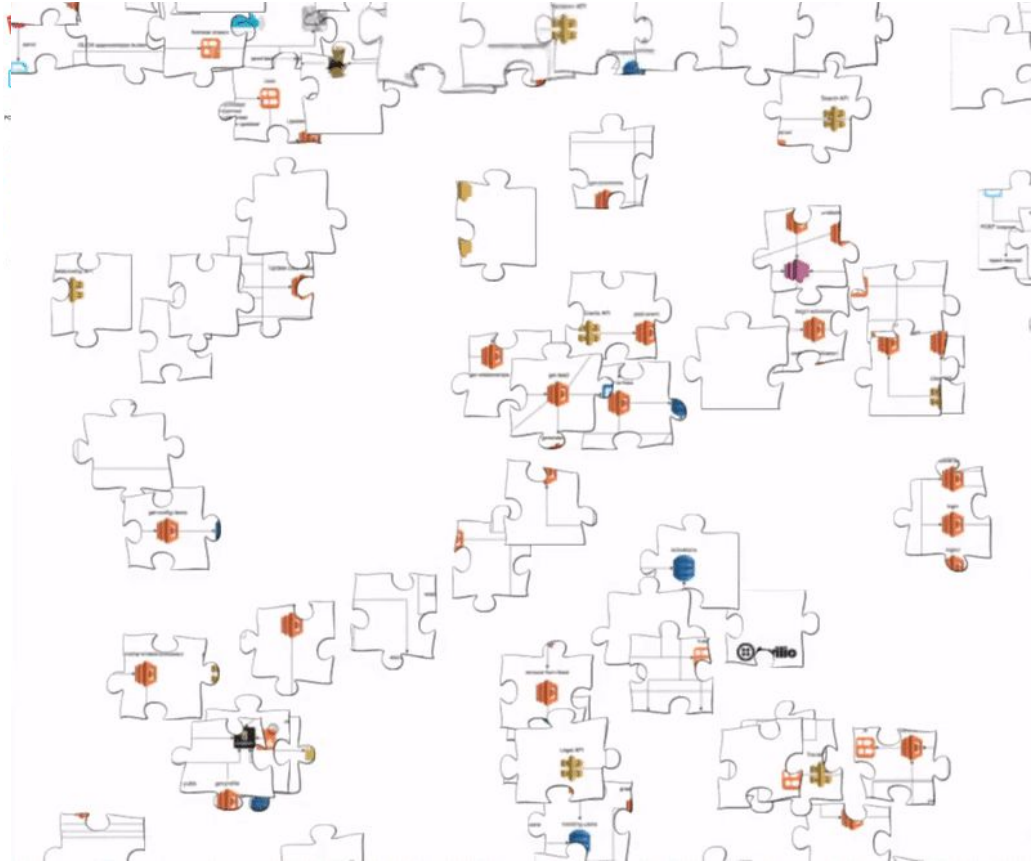


Decision





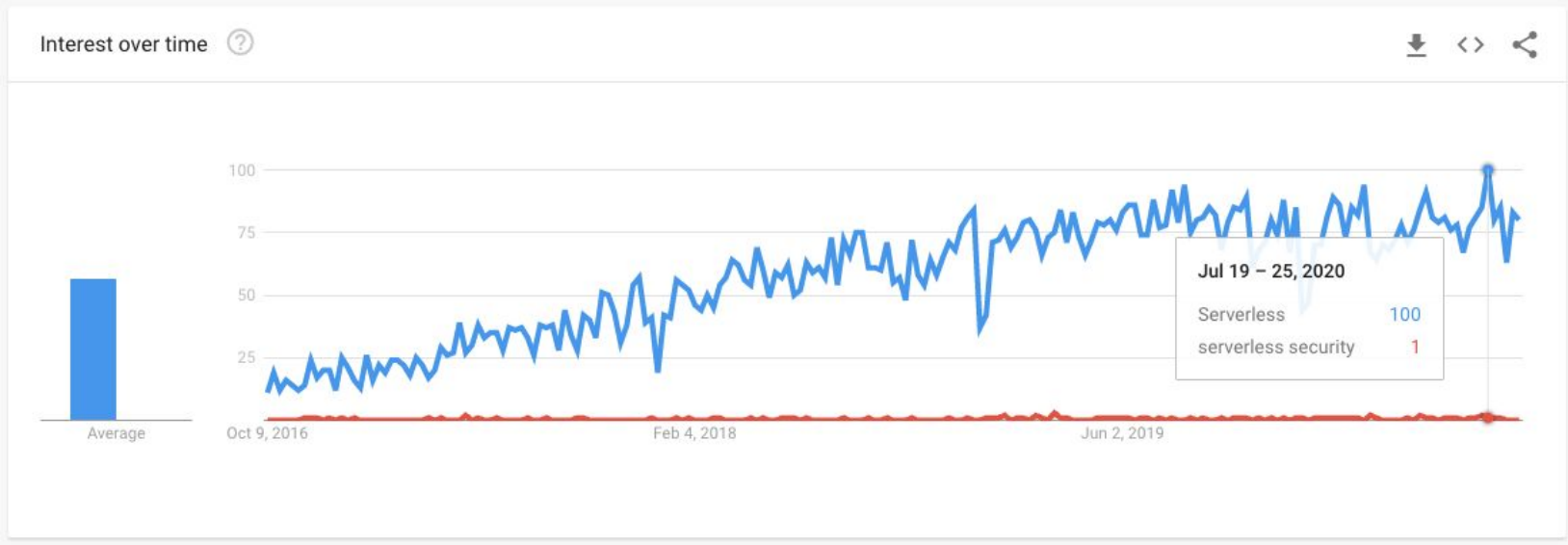
Serverless Architecture



What is Serverless?

- Container spins up when required
- Triggered via events
- Read-only environment, except for */tmp*
- Terminates when code execution
- Not wired to the internet*
- Data is temporary**
- Code reside in environment
- Keys are available as environment variables

What About Security?



https://trends.google.com/trends/explore?date=today%205-y&q=%2Fg%2F11c0q_754d

Can traditional AppSec Be **Applied to** Serverless?

Resource-Based IAM

DVSA-ORDER-NEW

```
def lambda_handler(event, context):
    orderId = str(uuid.uuid4())
    itemList = event["items"]
    status = 100

    userId = event["user"]
    address = "{}"
    ts = int(time.time())
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table(os.environ["ORDERS_T
    response = table.put_item(
        Item={...}
    )

    if response['ResponseMetadata']['HTTPStatus
        res = {"status": "ok", "msg": "order cr
    else:
        res = {"status": "err", "msg": "uld n

    return res
```

Execution role

Role name

serverlessrepo-DVSA-OrderNewFunctionRole-N65M2RQ1B6QS

Resource summary

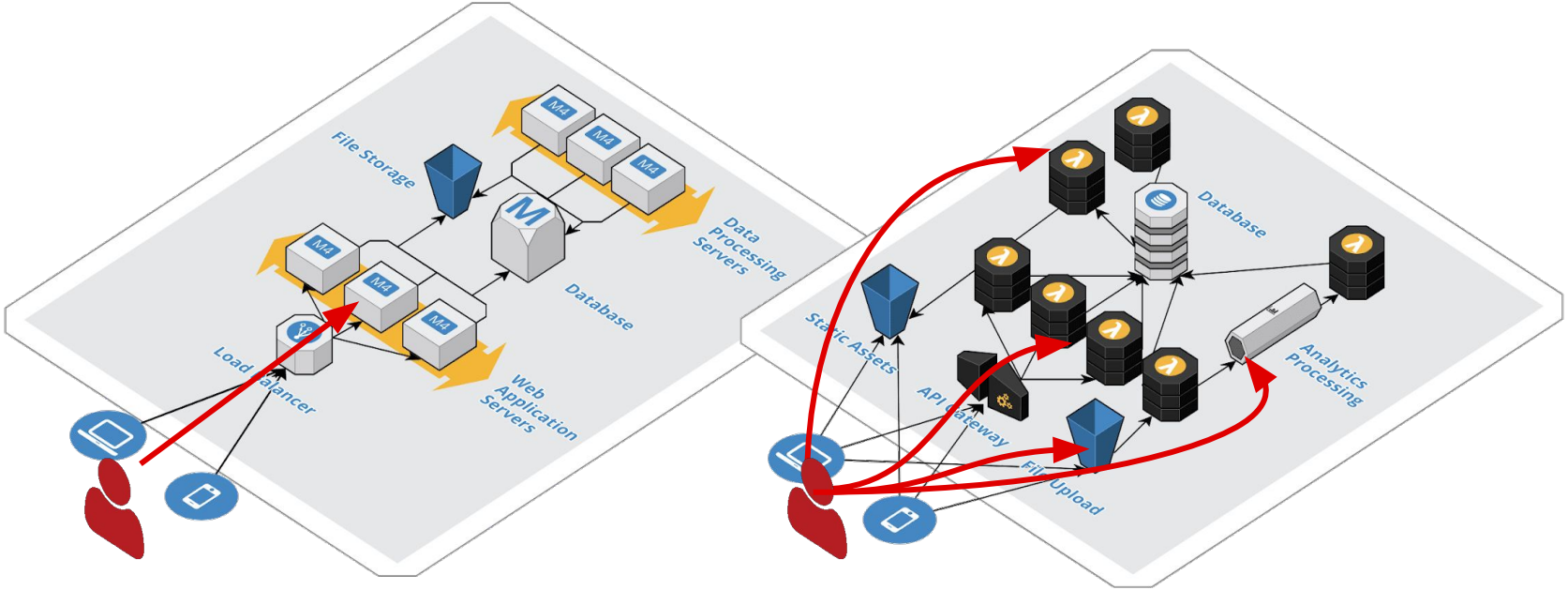
Amazon DynamoDB
1 action, 2 resources

To view the resources and actions that your function has permission to access, choose a service.

By action | By resource

Action	Resources
dynamodb:PutItem	Allow: arn:aws:dynamodb:us-east-1:402181209224:table/*

Loss of Perimeter



Serverless Risks

Event Injection

Broken Authentication

Sensitive Data Exposure

Over-Privileged Functions

Vulnerable Dependencies

Insufficient Logging & Monitoring

Open Resources

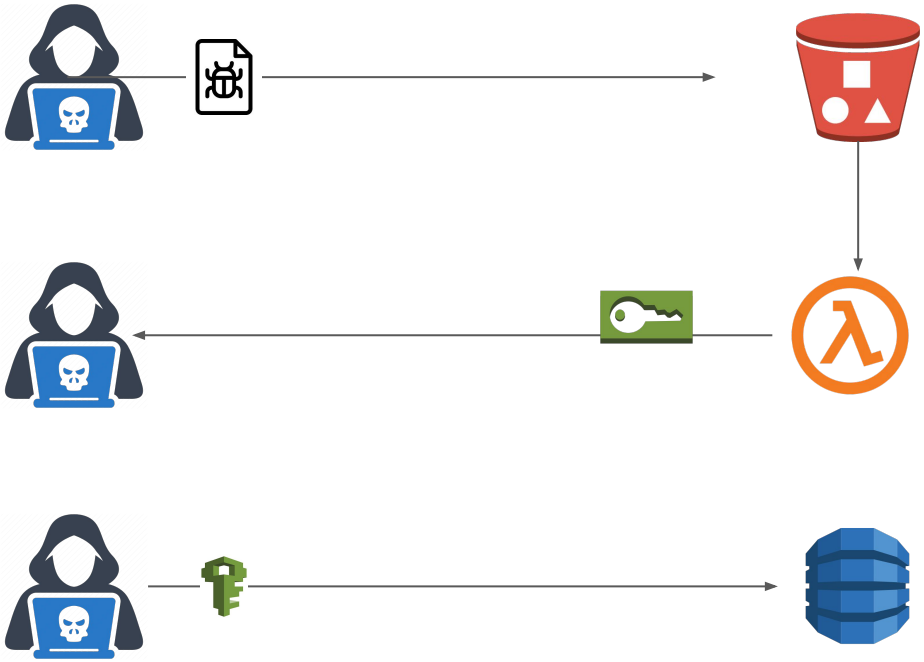
DoW / DoS

Insecure Secret Management

OWASP Serverless Top

- Current project state:
  - Interpretation of Top 10
  - Open Data Call: http://tiny.cc/slscall
- Goal: Serverless-tailored Top 10

Demo

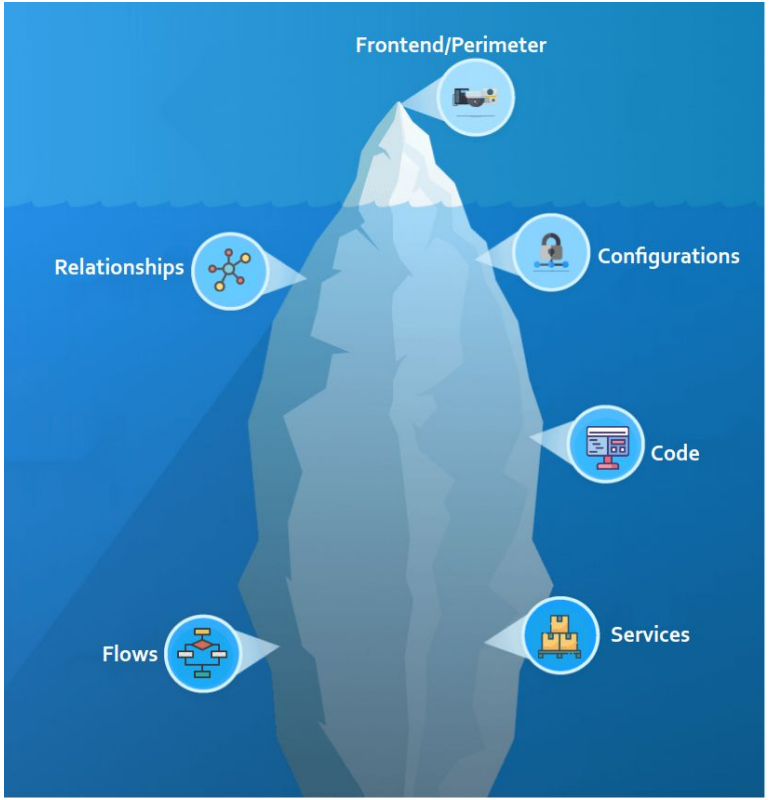


Scale

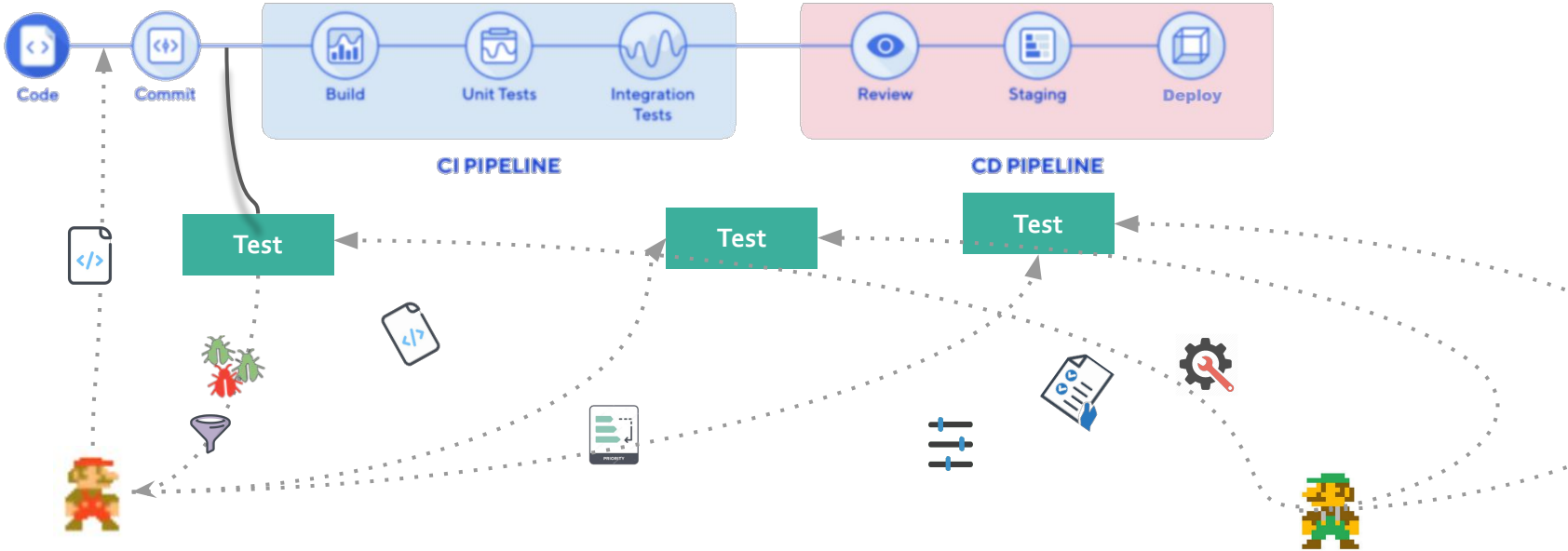
- Lots of services
- Frequent deployments
- Who's talking to who?
- Many developers
- Few AppSec
- What's important?
- Is security even the same?
- Who takes care of Infra?

Traditional **AppSec Testing** for **Cloud Native**

- Ignorant to the environment and context
- Completely blind to no-edge services
- Block developers, disruptive to CI/CD
- Hard to scale



Traditional Testing in **Modern** CI/CD Pipelines

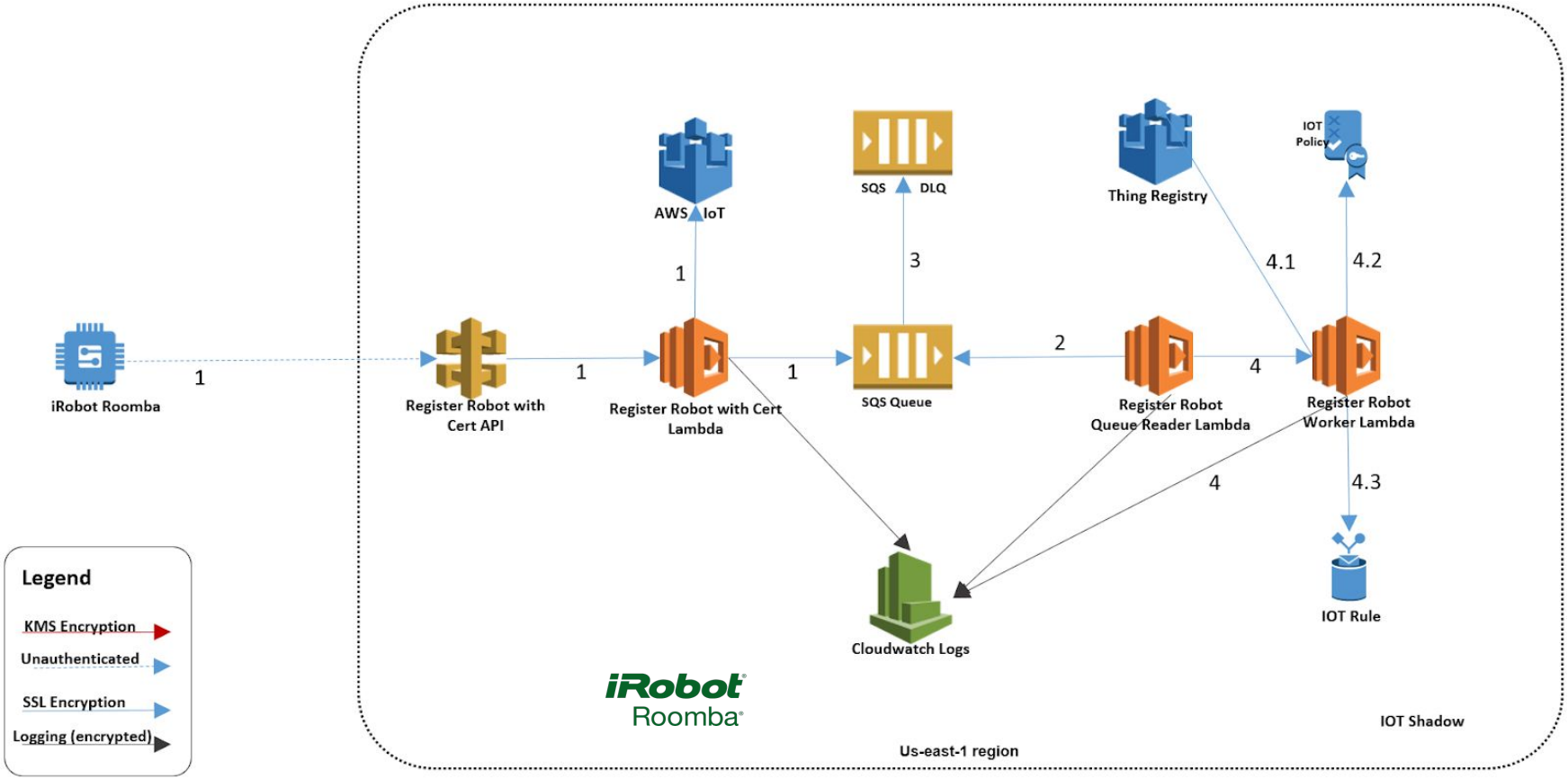


Mario,
Sr. Developer, Toadstool Inc.

Luigi,
AppSec Team

How do **Security** Test Serverless **Apps**?

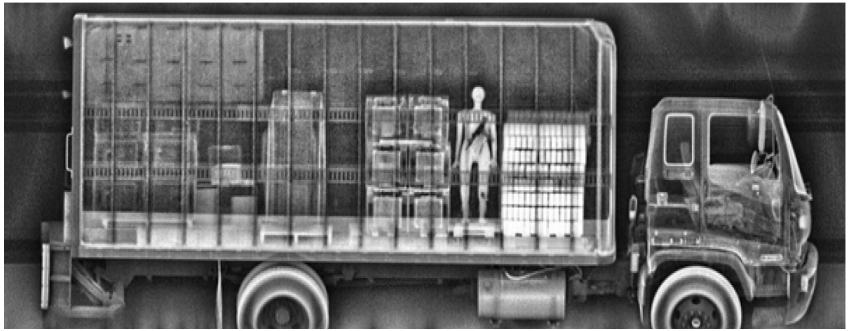
iRobot Serverless App



Source: https://aws.amazon.com/solutions/case-studies/irobot/

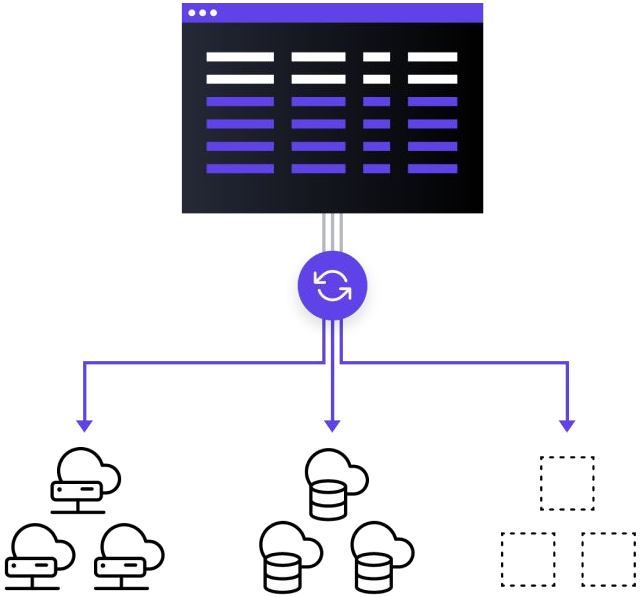
Easy... **SCA** & Image Scanning

- Covers ~10% of your app
- Just fixing problems you imported
- What about your **CODE**, your services & your configurations?
- Provided by the cloud provider and many OS Project



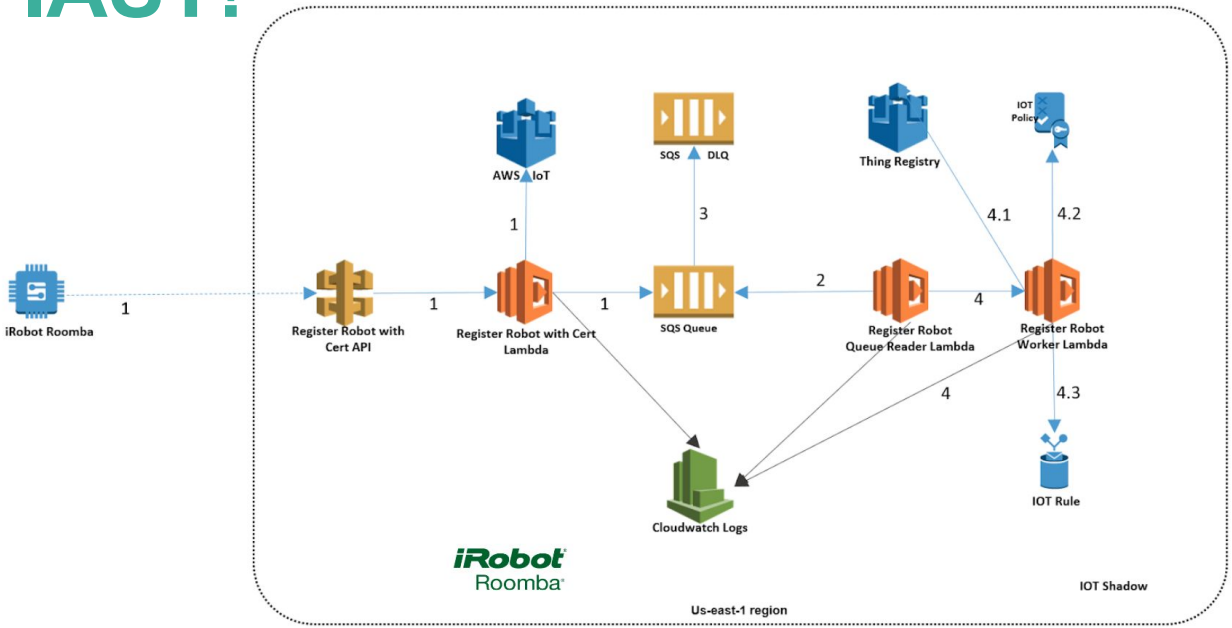
I know… IaC

- Really?! Blaming it on the DevOps?
- As left as it can get, but…
  - Limited visibility
  - **Zero code coverage**
  - No logic, no prioritization
  - IaC dependant



IAST!

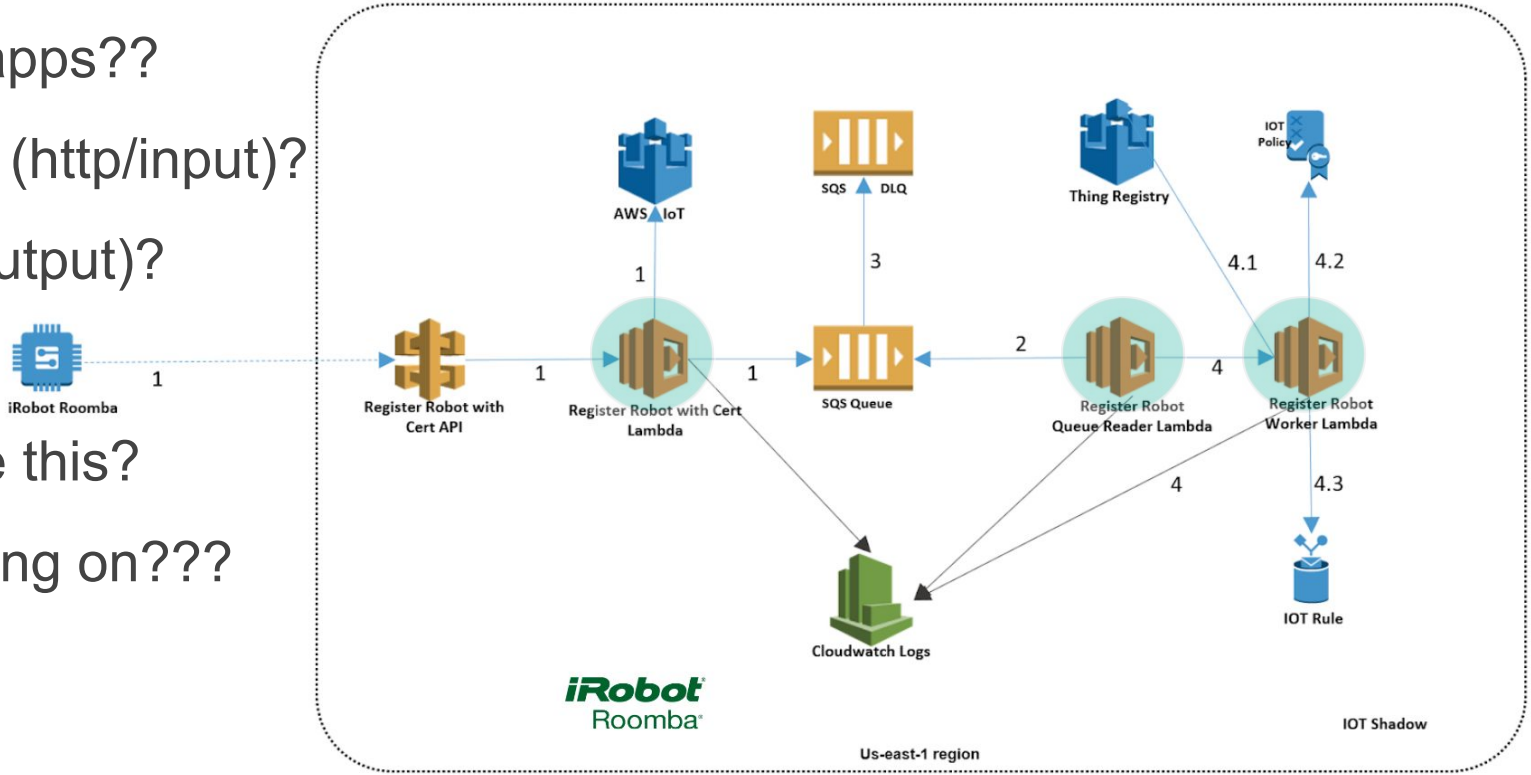
- Modern AppSec
- Accurate & Reliable
- Enables DevOps



But! No <u>Server</u> to instrument...

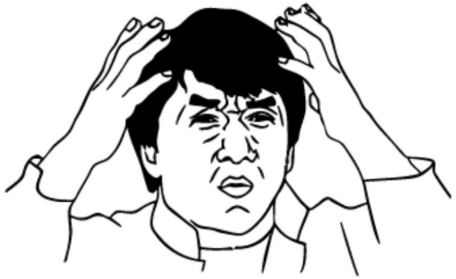
SAST

- So many apps??
- No source (http/input)?
- No sink (output)?
- Who wrote this?
- What's going on???

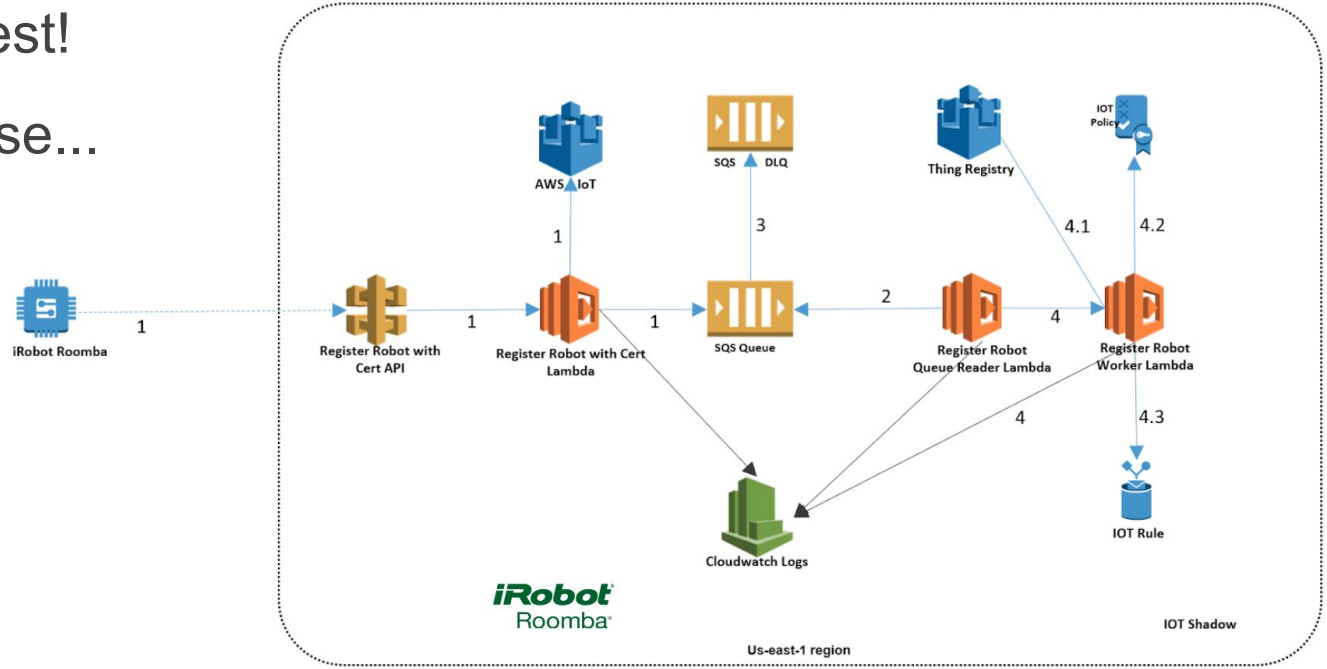


DAST

- No HTTP/S Request!
- No (sync) Response...
- Nothing to test?

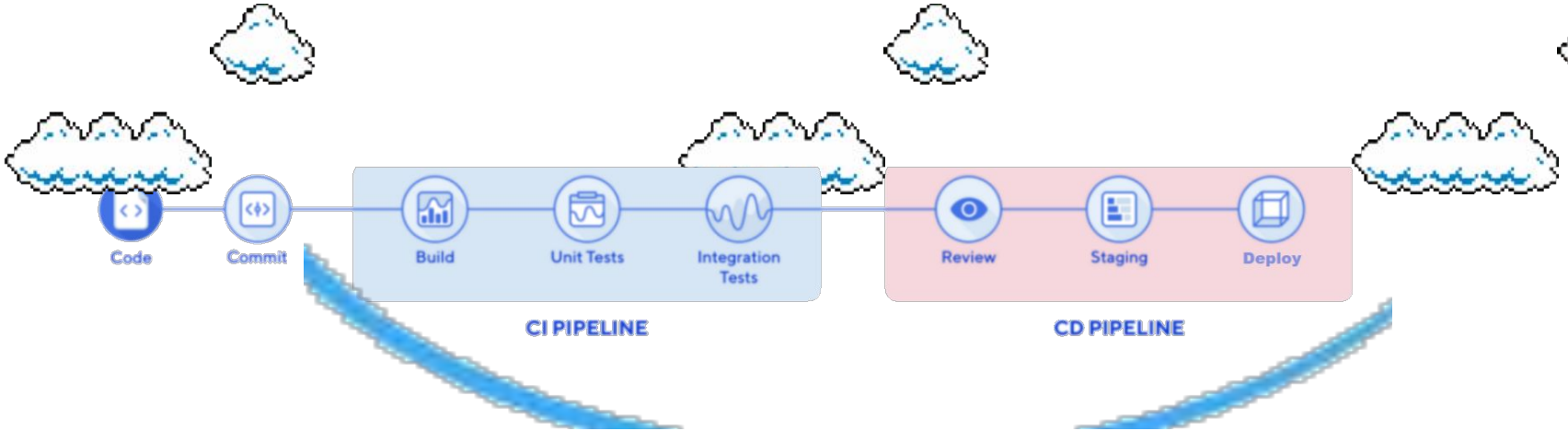


Where do I even start?!



I give up!

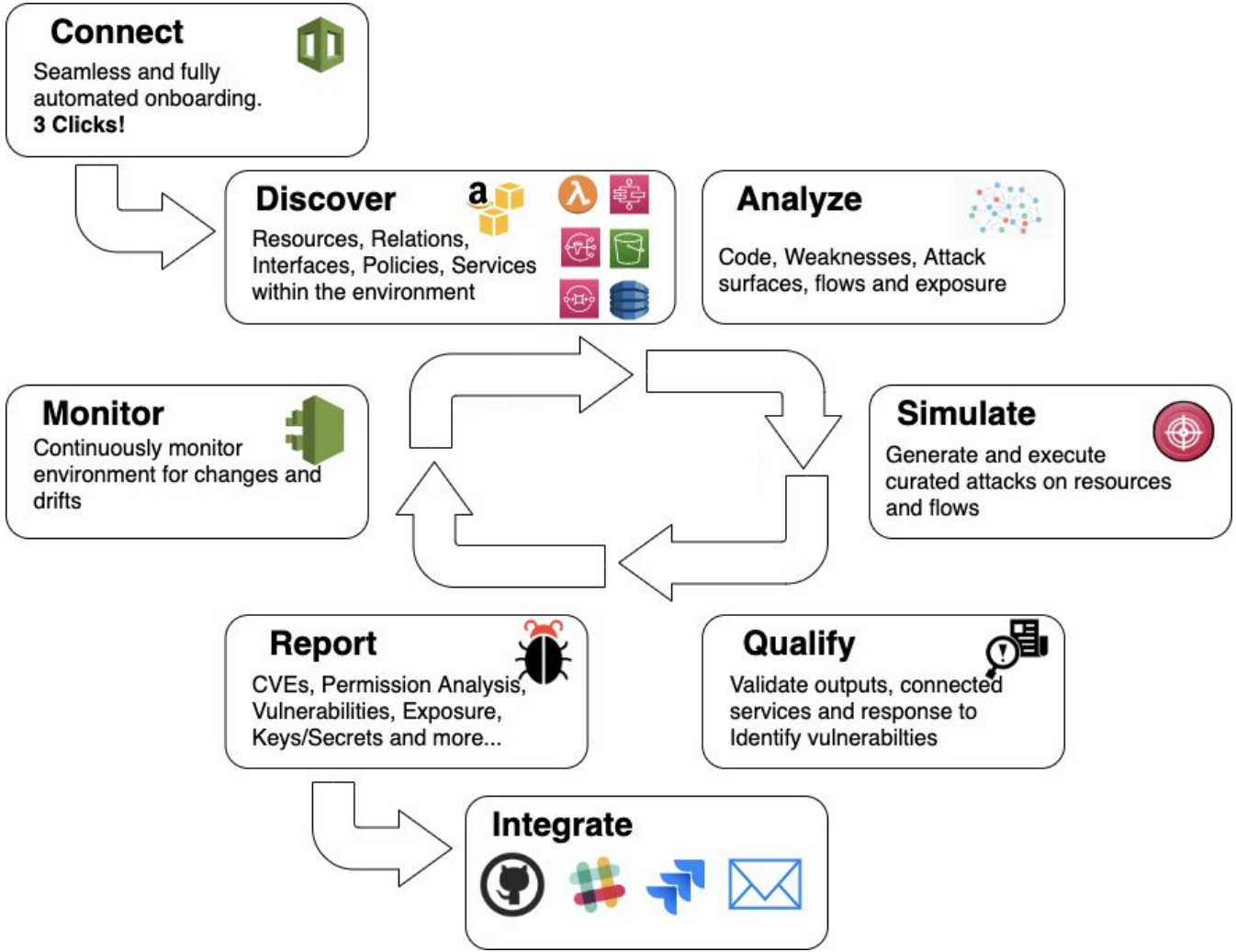
AppSec Testing, **Redefined** for the Cloud



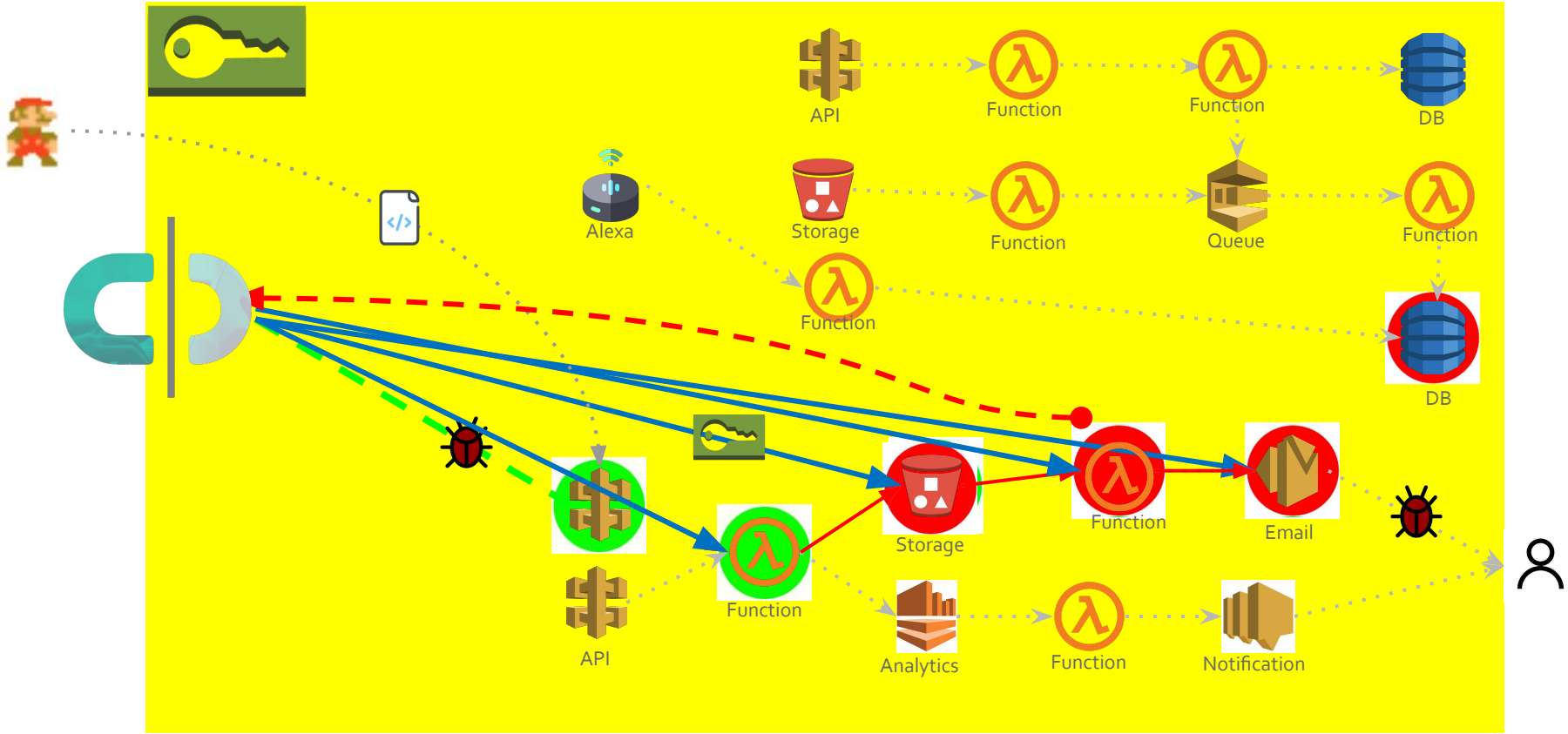
Mario,
Sr. Developer, Toadstool Inc.

Luigi,
AppSec Team

Continuous. Frictionless. **Complete.**


Connect
Seamless and fully automated onboarding.
3 Clicks!
Discover
Resources, Relations, Interfaces, Policies, Services within the environment
Analyze
Code, Weaknesses, Attack surfaces, flows and exposure
Monitor
Continuously monitor environment for changes and drifts
Simulate
Generate and execute curated attacks on resources and flows
Report
CVEs, Permission Analysis, Vulnerabilities, Exposure, Keys/Secrets and more...
Qualify
Validate outputs, connected services and response to Identify vulnerabilties
Integrate

Example Use Case



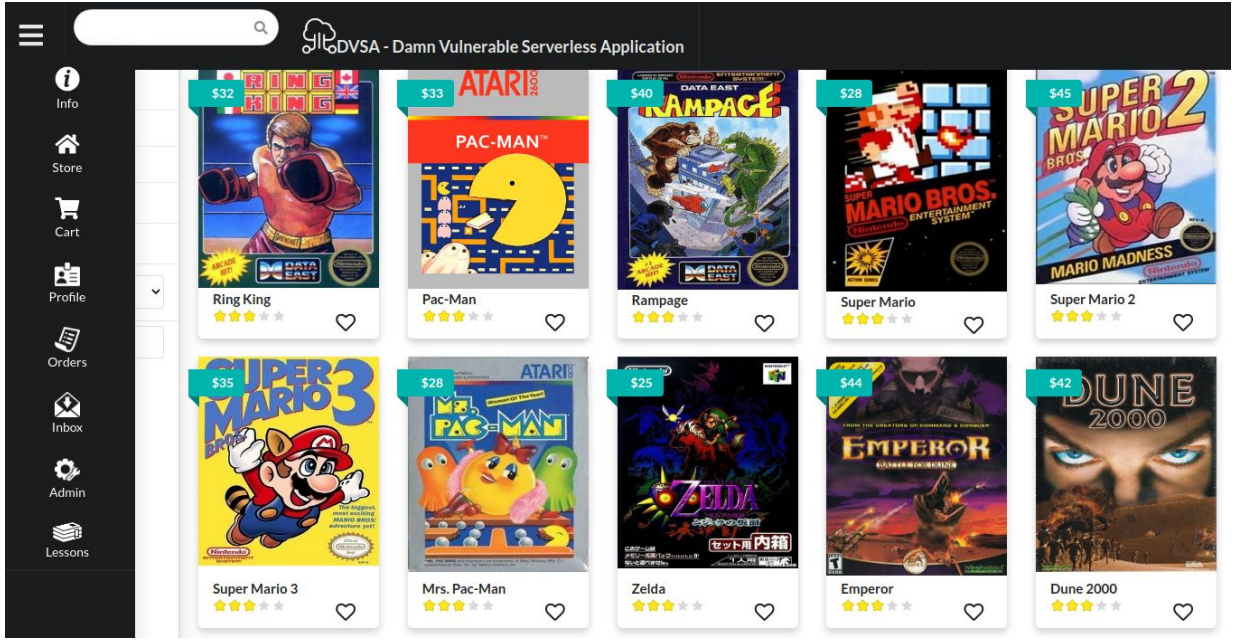
github.com/owasp/dvsa

@DVSAowasp

! NOT in PRODUCTION !

DVSA

DAMN VULNERABLE SERVERLESS APPLICATION



CAST?

OS Command Injection

HIGH | Category: Exploits | Function: arn:aws:lambda:us-east-1:733980656228:function:DVSA-FEEDBACK-UPLOADS

Description

Command injection (CWE-77, CWE-78) is an attack in which the goal is execution of arbitrary commands on the host operating system via a vulnerable application. Command injection attacks are possible when an application passes unsafe data to a system shell. In this attack, the attacker-supplied commands is executed with the privileges of the vulnerable function. Command injection attacks are possible largely due to insufficient input validation. This attack differs from Code Injection, in that code injection allows the attacker to add his own code that is then executed by the application. In Command Injection, the attacker extends the default functionality of the application, which execute system commands, without the necessity of injecting code.

What happened

cloudessence has identified evidence in the function logs (CWL) that indicates of was successfull exploit

```
logGroup: /aws/lambda/DVSA-FEEDBACK-UPLOADS
logStream: 2021/11/02/[$LATEST]45c9e5cafd5842eba1d14815437c3b31
payload: |echo $((4+4854071629342))
requestId: 5b604244-cc03-4f48-861f-7937f645809f
type: Log
vector: arn:aws:s3:::dvsa-feedback-bucket-733980656228
```

Remediation

If at all possible, use library calls rather than external processes to recreate the desired functionality. Additionally, it is recommended to sanitize all inputs from untrusted sources.

Impact

Least Privilege Violations

MEDIUM | Category: Permissions | Function: arn:aws:lambda:us-east-1:733980656228:function:DVSA-ADMIN-GET-ORDERS

Description

The Attached Role arn:aws:iam::733980656228:role/DVSA-AdminRole-dev has an over permissive policy that may violate the principle of least privilege. For more information on aws least privilege docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege.

What happened

- DVSA-AdminRolePolicy-dev: Permissions for service s3 should be removed
- DVSA-AdminRolePolicy-dev: Use of wildcard in service dynamodb is inadvisable

Violating policies

DVSA-AdminRolePolicy-dev

> View violating policy

Remediation

Replace the existing policy with one of the following:

JSON YAML TERRAFORM

```
{
    "PolicyName": "DVSA-AdminRolePolicy-dev",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "1",
                "Effect": "Allow",
```

ONE DEVSECOPS PLATFORM

DEV

Write secure software.
Empower developers to write secure software quickly.

TEST

Validate, fix and assure secure software development.

PROD

Run software securely.
Stop attacks.

CONTRAST SCAN NEW

Accurately identify & remediate vulnerabilities based on code scans

CONTRAST ASSESS

Accurately identify & remediate vulnerabilities based on runtime analysis

CONTRAST PROTECT

Prevent known and unknown app exploits in production

CONTRAST OSS

Secure the use of open-source software across your portfolio

CONTRAST for SERVERLESS NEW

Accurately identify & efficiently remediate vulnerabilities in Serverless functions

Thank you!

itay.rozenman@ContrastSecurity.com

https://reinvent.awsevents.com/learn/jams-and-gamedays/