BSIDES MILANO

Contrast SECURITY

Hacking Serverless Applications

A Treasure Map for Uncharted Waters

Matteo Rosi | Cloud Native Security Researcher
matteo.rosi@**contrastsecurity.com**

Whoami

ottimo

@ottimorosi

matteorosi

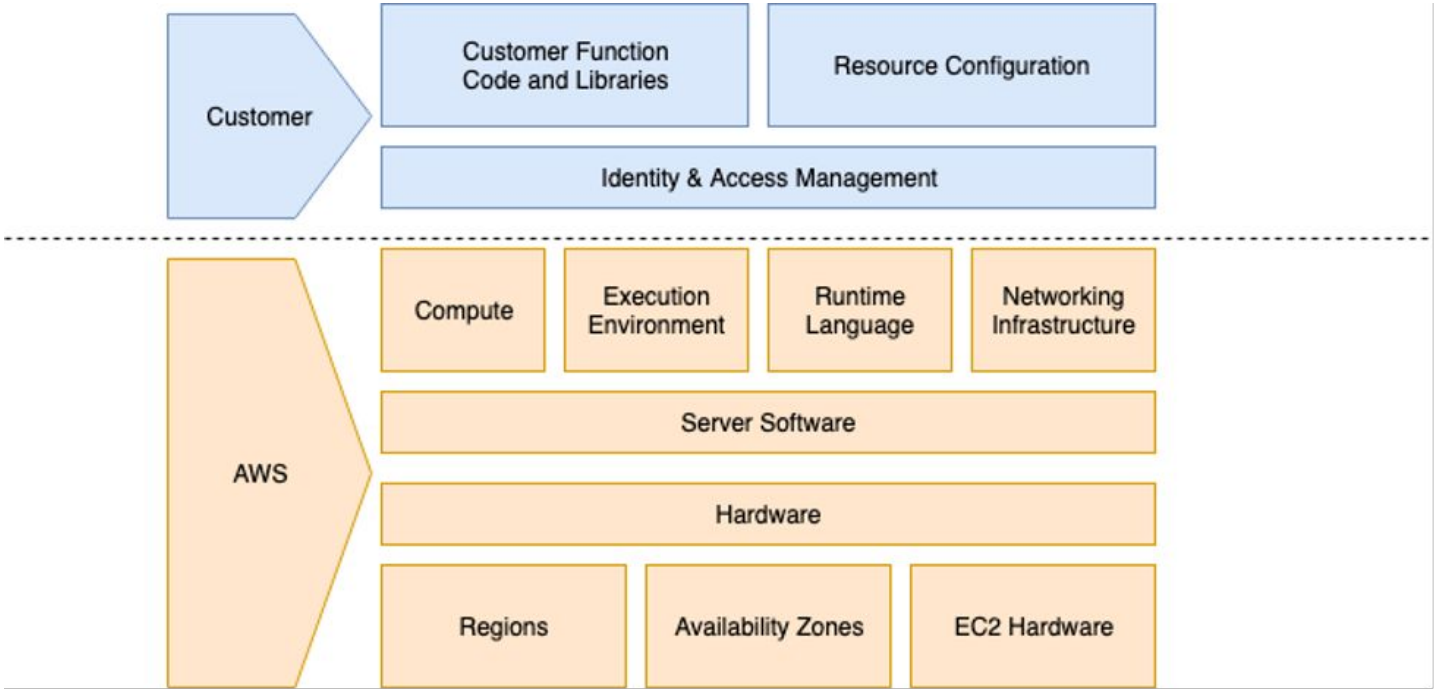
ottimo.me

matteo.rosi@contrastsecurity.com

Security Researcher @ Contrast Security

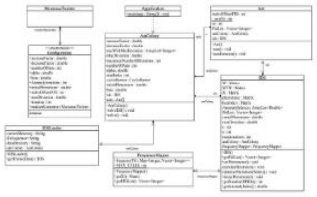
Head of Cyber Defense & CISO Deputy @ Telepass spa
SOC Manager @ 7Layers srl

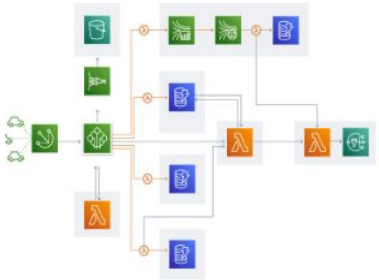
AWS Serverless Shared Responsibility Model



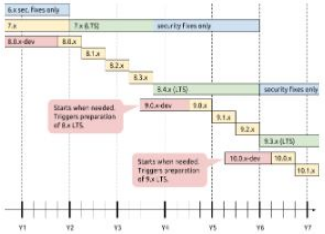
Serverless for DevSecOps

Architecture

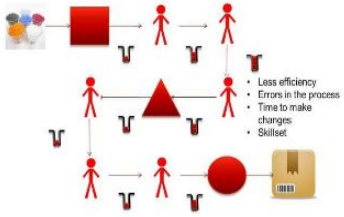




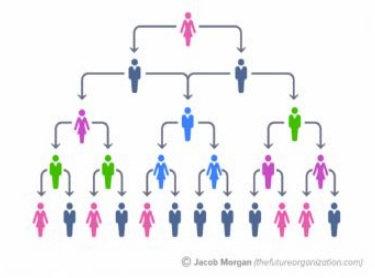
Cycles

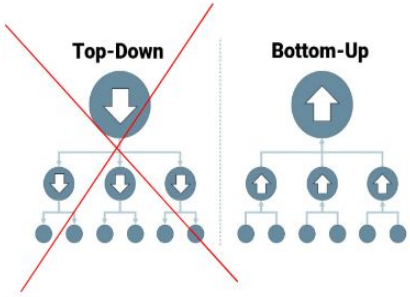


Process



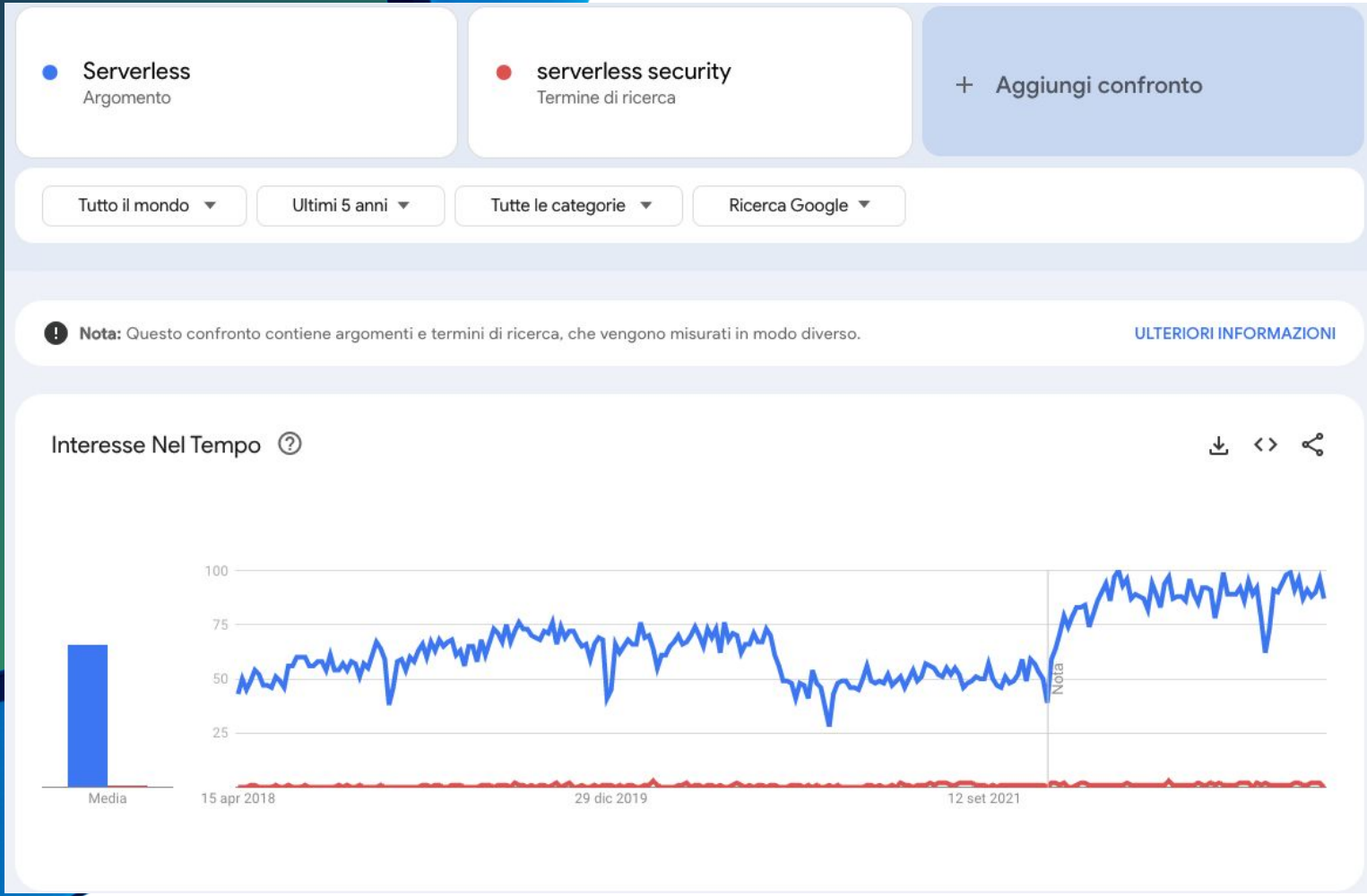
Decision



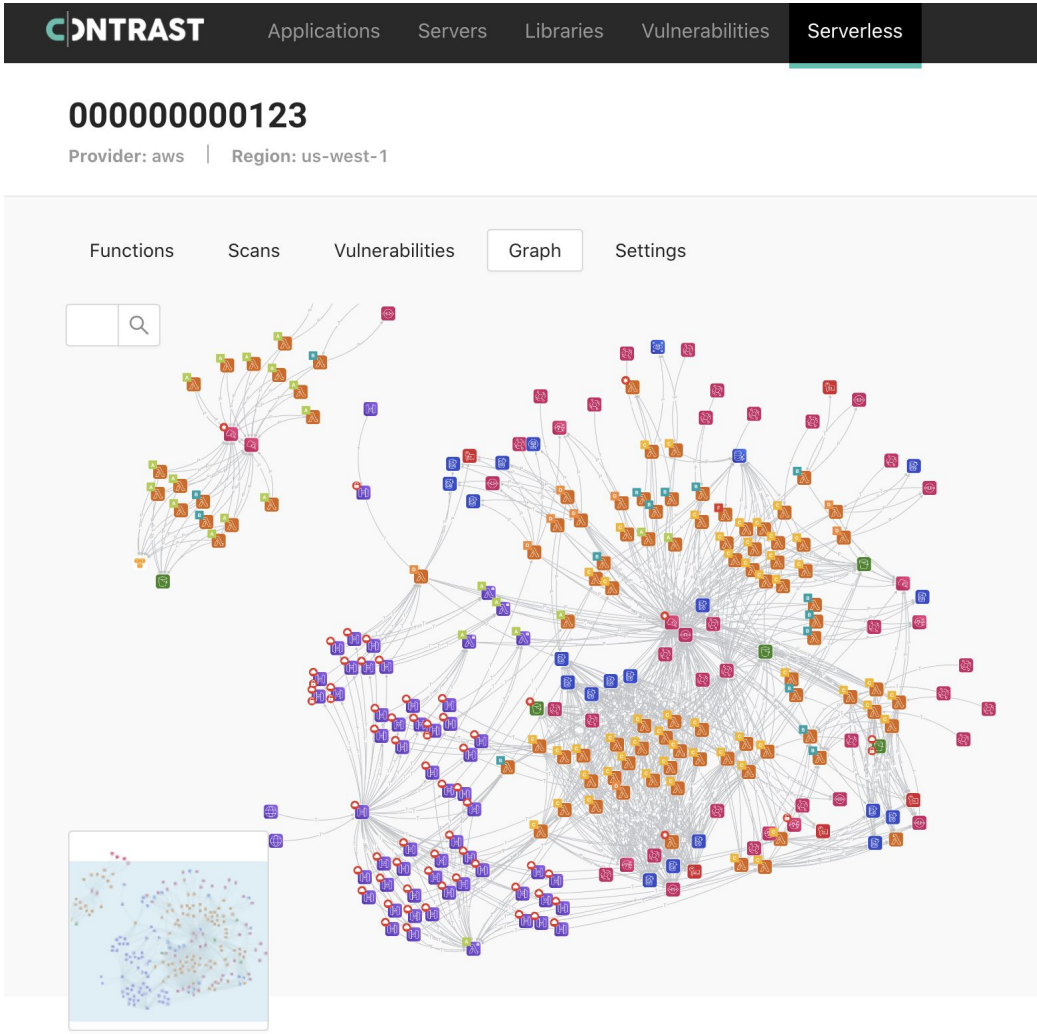


Serverless

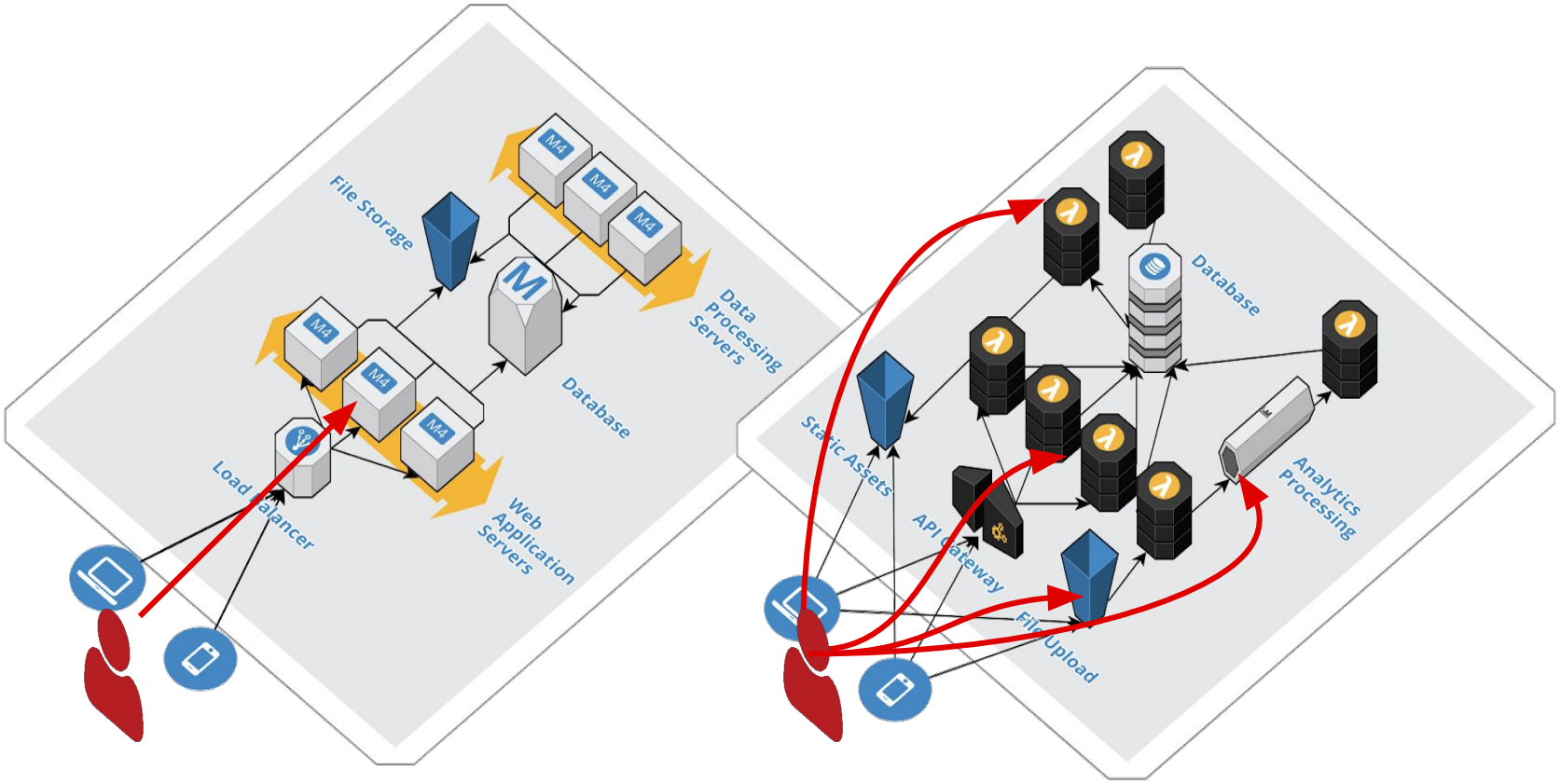
Trends



Serverless Architecture



Loss of Perimeter



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": ...r cr
    else:
        res = {"status": "err", "msg": ...ld n

    return res
```

Execution role

Role name

serverlessrepo-DVSA-OrderNewFunctionRole-N65M2RQ1B6QS

Resource summary

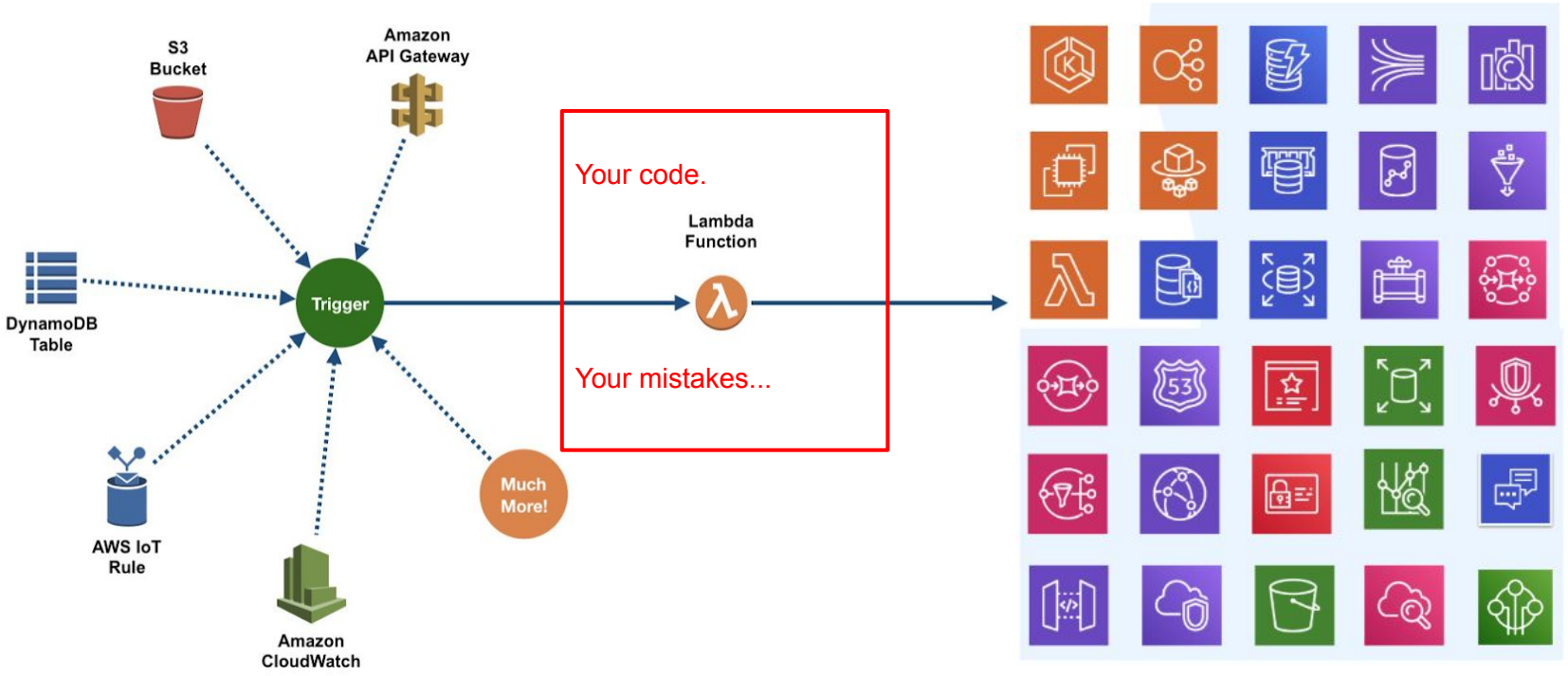
Amazon DynamoDB
1 action, 2 resources

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/*

Event-Driven Architecture

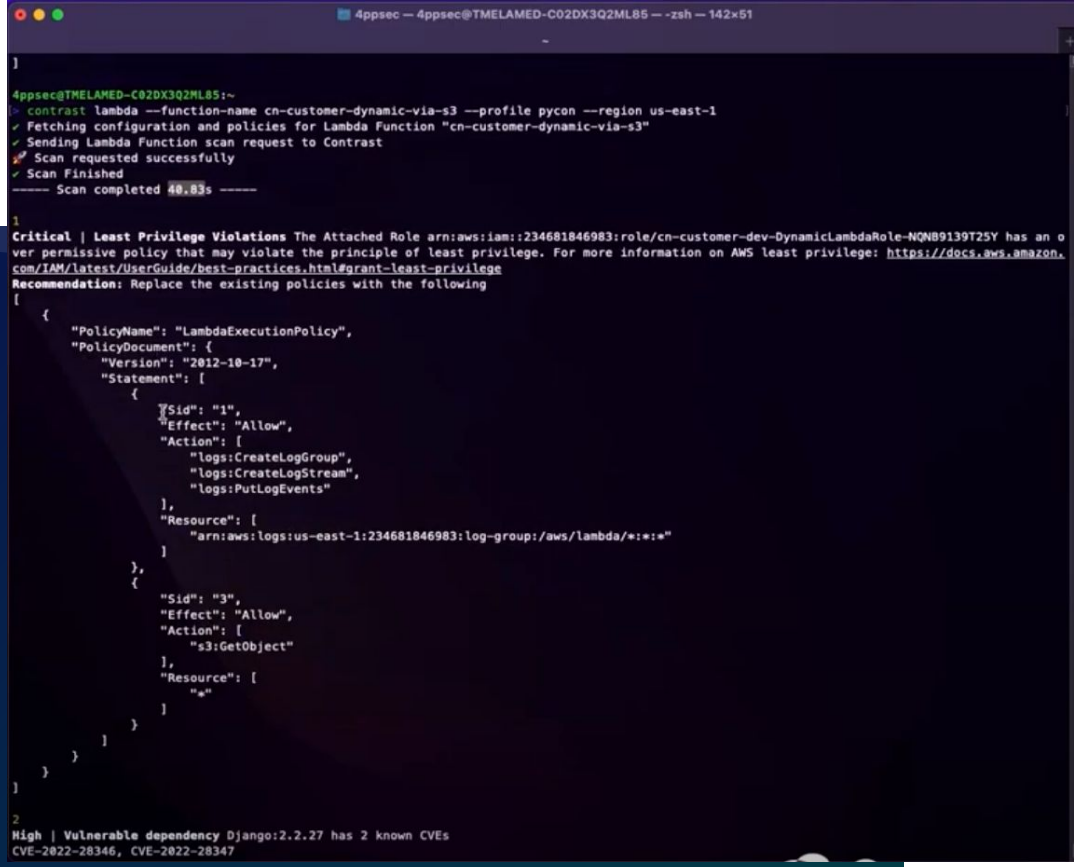


1. Triggered via events
2. CSP spins up a container with code
3. Container terminates after code executes**

Serverless Risks

- Event Injection
- Broken Authentication
- Sensitive Data Exposure
- Over-Privileged Functions
- Vulnerable Dependencies
- Insufficient Logging & Monitoring
- Open Resources
- DoW / DoS
- Insecure Shared Space
- Insecure Secret Management

BSIDES MILANO



CodeSec by Contrast

https://www.contrastsecurity.com/developer

DVSA

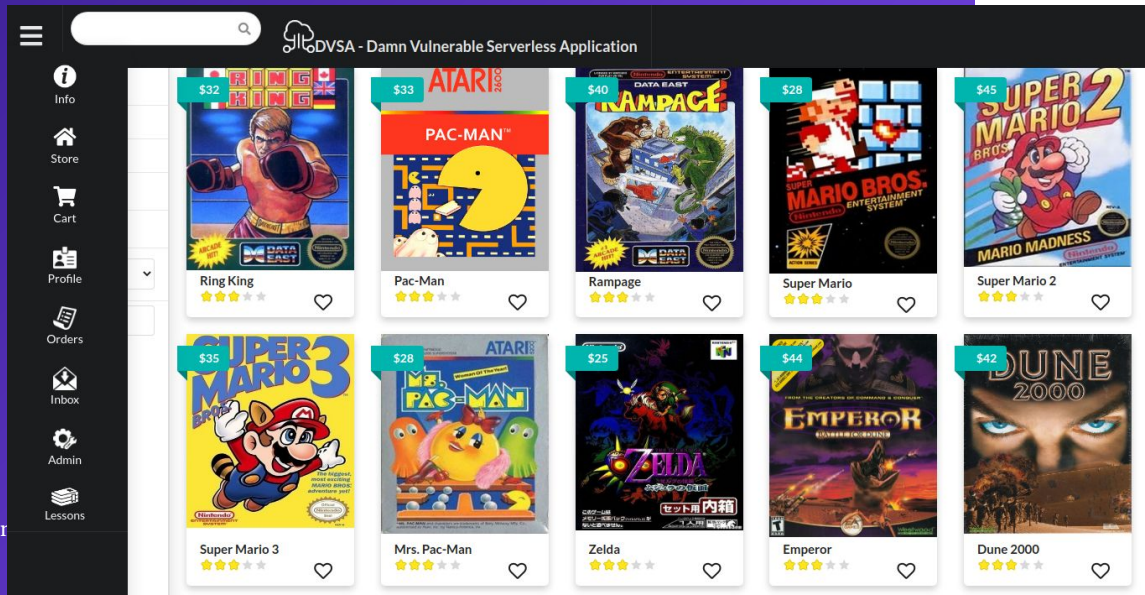
DAMN VULNERABLE SERVERLESS APPLICATION

https://owasp.org/www-project-dvsa/

github.com/owasp/dvsa

Twitter: @DVSAowasp

! NOT in PRODUCTION !



Security Aspects

AWS Lambda

- Read-only environment, except for /tmp
- Not wired to the internet*
- Data is temporary**
- Code reside in environment
- Keys are available as environment variables

Demo 1

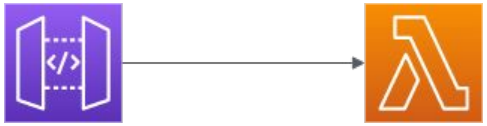
SpEL: CVE-2022-22963

Lambda Runtime: Java

Framework involved: Spring Cloud Function

Vulnerability: CVE-2022-22963

Trigger: API Gateway



Demo 2

Insecure Deserialization

Lambda Runtime: Python

Library involved: PyYAML

Vulnerability: CVE-2020-14343

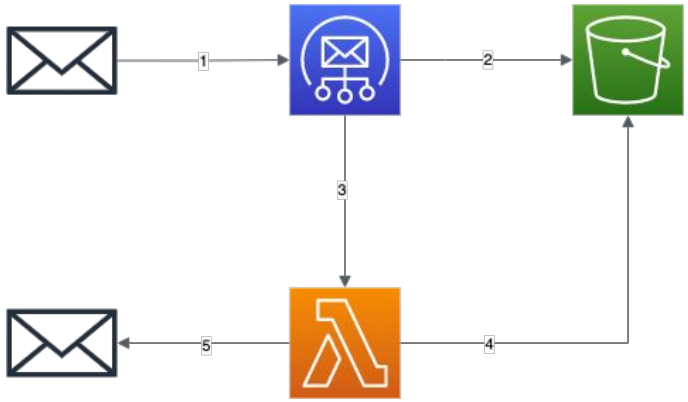
Trigger: S3 Bucket ObjectCreated

Demo 3

SES Injection

Lambda Runtime: Python

Trigger: AWS SES (email)



Demo 4

Alexa Injection

Lambda Runtime: Python

Trigger: Alexa Skill