# Security Uno: a fun way to threat model

**Kendra Ash @securelykash**

# The Portland Chapter

Twitter: @PortlandOWASP

Website: pdxowasp.org

Meetup: OWASP-Portland-Chapter

Slack: owasp.slack.com #chapter-pdx

# Who are you?
# QA, DEV, Security?...

# Are we in the cloud?

# Agenda

Kendra Ash

Introduction and Background

Threat Modeling

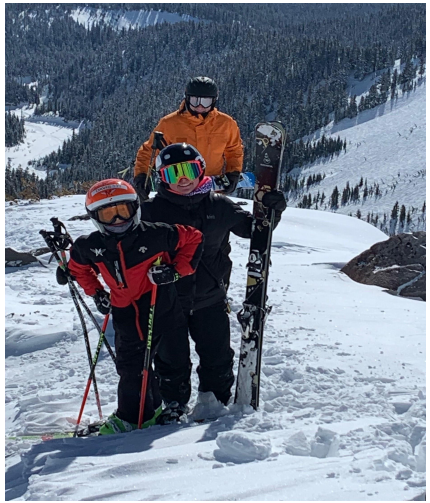
Support / Adoption

Card Game

Automation

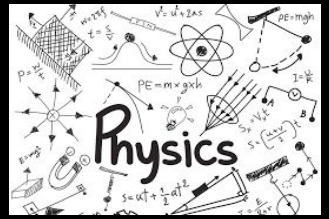
Future State

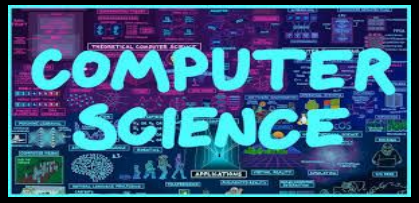
# Introduction
# Who am i?



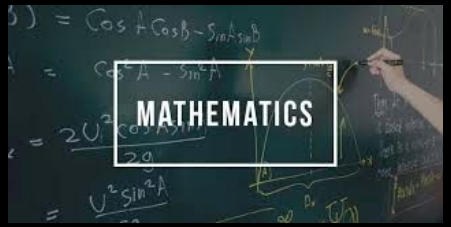
# A Little about me

@securelykash

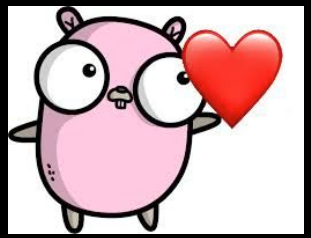



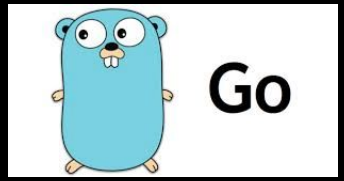


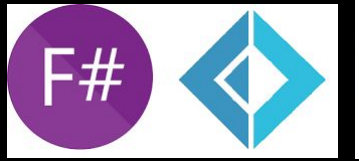


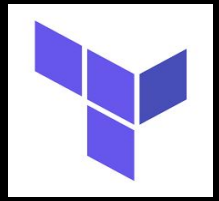






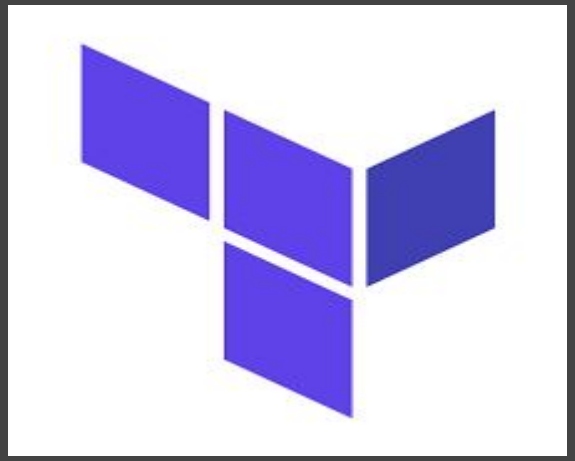




Intern * QA Engineer * Backend Engineer * Security Engineer


vacasa
Unbeatable revenue
Advanced rate optimization
Local caretaking teams
Dedicated property managers
Vacation rentals you can
Love your job.
vacasa
Better management means better vacations.
vacasa
vacasa
24/7 guest support
Professional care + housekeeping
3D virtual tours
15,000+ vacation rentals around the world
@securelykash

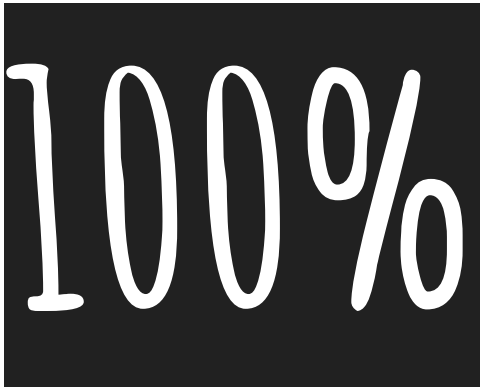




TypeScript



# Environment at Vacasa

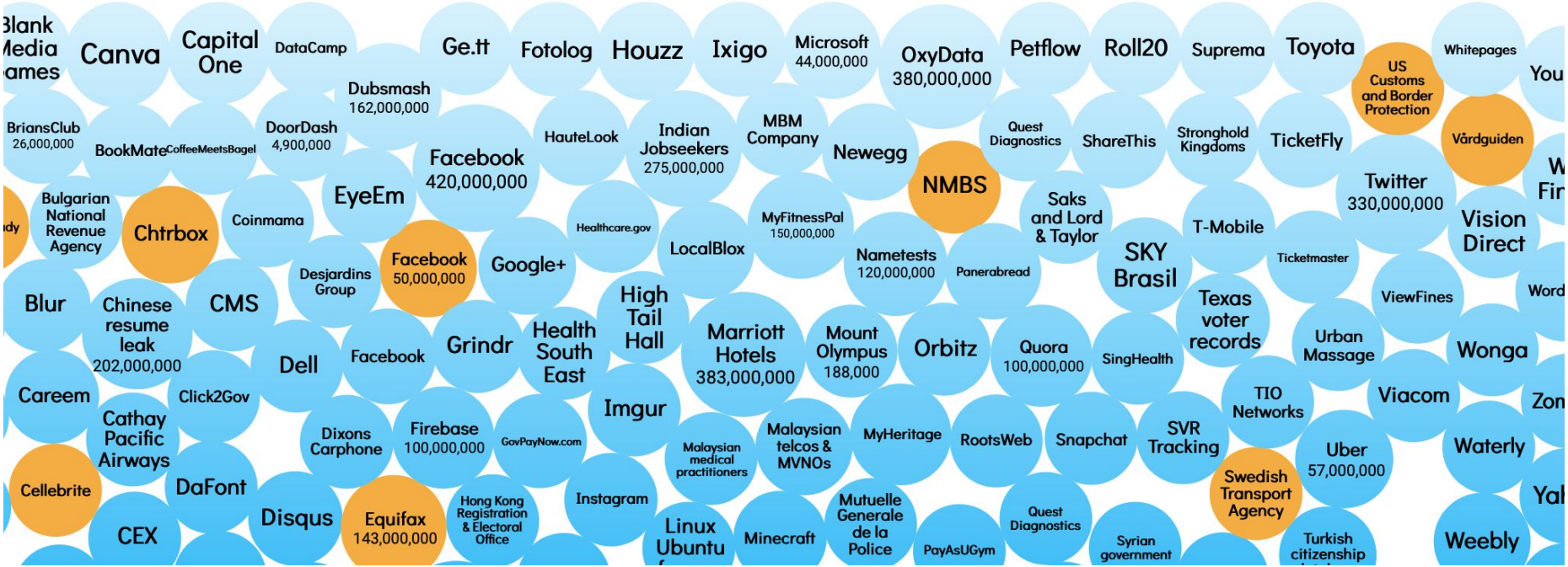
@securelykash




Attempting to make Vacasa a more secure place!

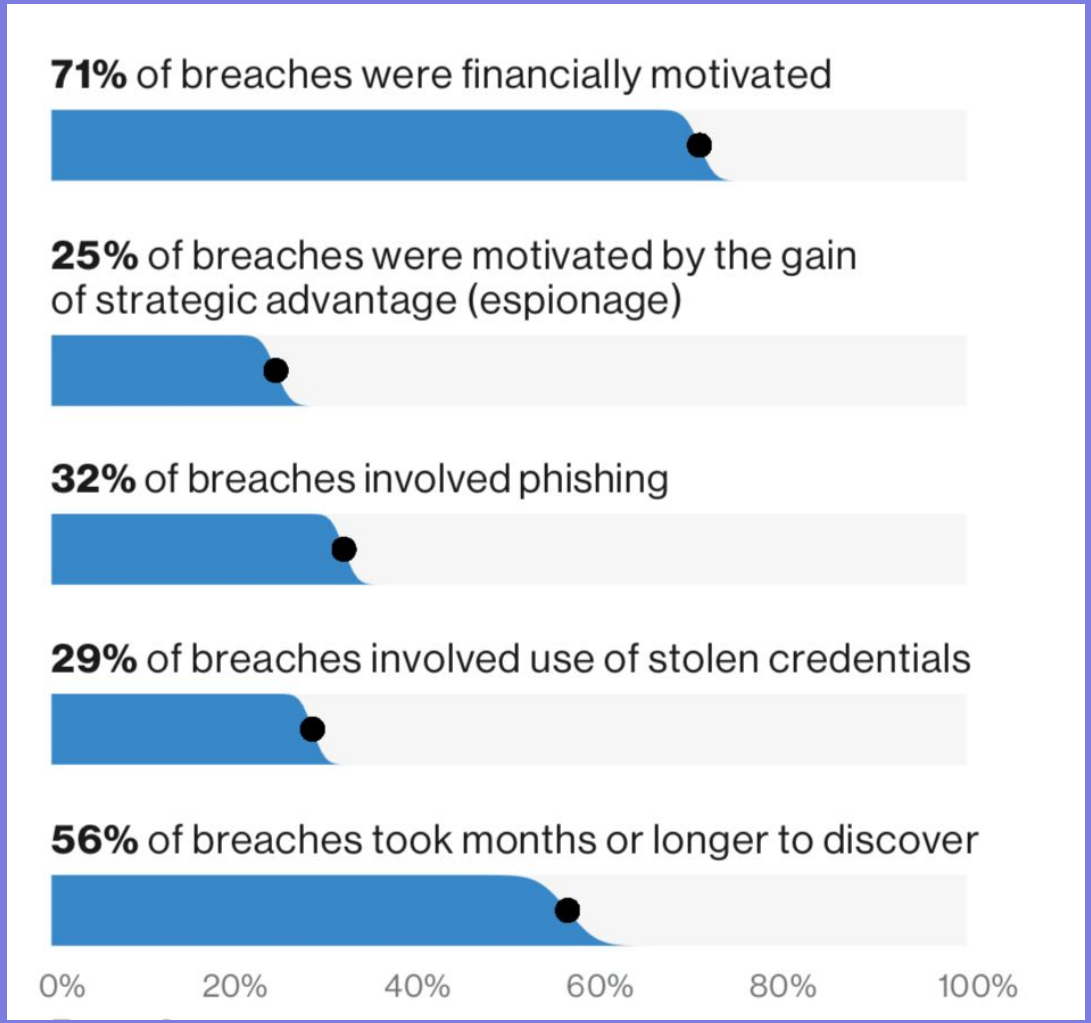

Verizon data breach investigation report for 2019.

2,013 of the 41,686 incidents were confirmed data breaches.




# Worlds biggest data breaches 2016-2019

@securelykash



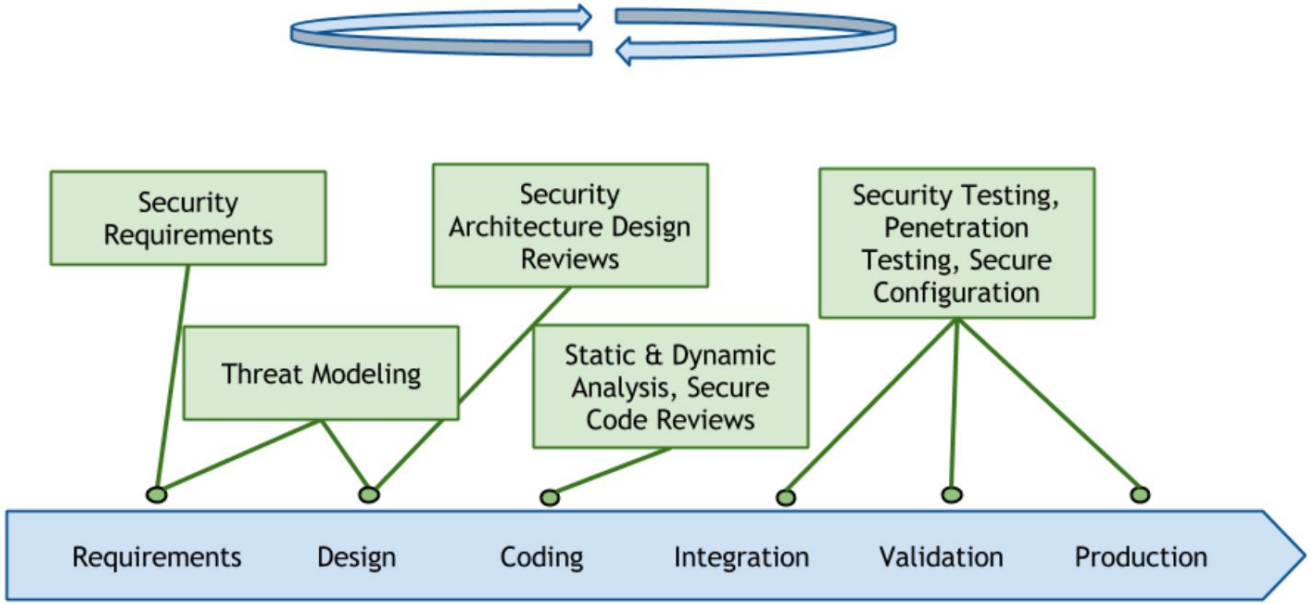
# Breaches 2019

Capital One (banking company, S3 bucket).

Facebook (Elasticcache Cluster)

Marriott Hotels (acquisition)

# Security in the SDLC Process



# Threat

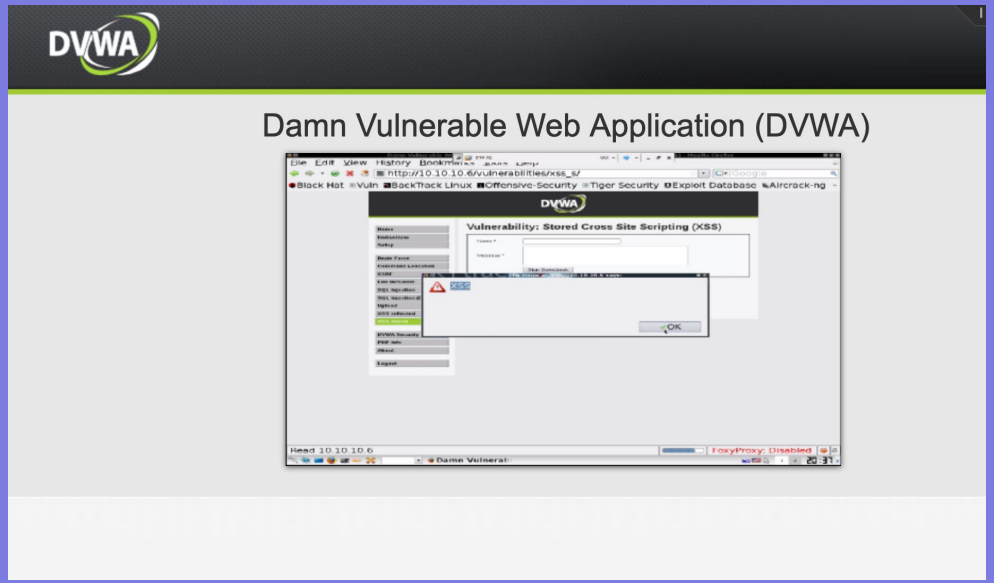
Vary depending on technologies

People

Robots

Mother Nature

THE POSSIBILITY OF A MALICIOUS ATTEMPT TO DAMAGE OR DISRUPT A COMPUTER NETWORK OR SYSTEM




http://www.dvwa.co.uk/


https://bkimminich.gitbooks.io/pwning-owasp-juice-shop/content/

@securelykash

# OWASP ServerlessGoat

OWASP ServerlessGoat is a deliberately insecure realistic AWS Lambda serverless maintained by OWASP.

Enter a URL of a Word Doc (.doc) file to convert:

https://www.puresec.io/hubfs/document.doc

PURESEC AND OWASP FOUNDATION.


STRIDE

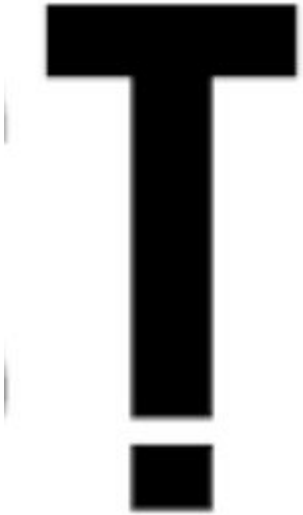
# History of STRIDE

Created at Microsoft by: Praerit Garg and Loren Kohnfelder a

When: 1990's

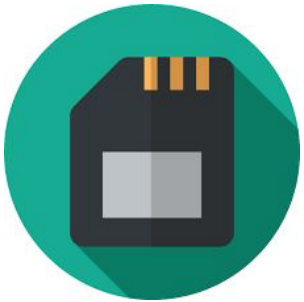

Email spoofing
Jean Quinn
From: Kelly Adams
Sent: 4/11/19 9:13 AM
To: Keith Jones
Subject: Data

# SPOOFING



Modifying something: code, configuration, or data.

Tampering with networks, memory, or files.



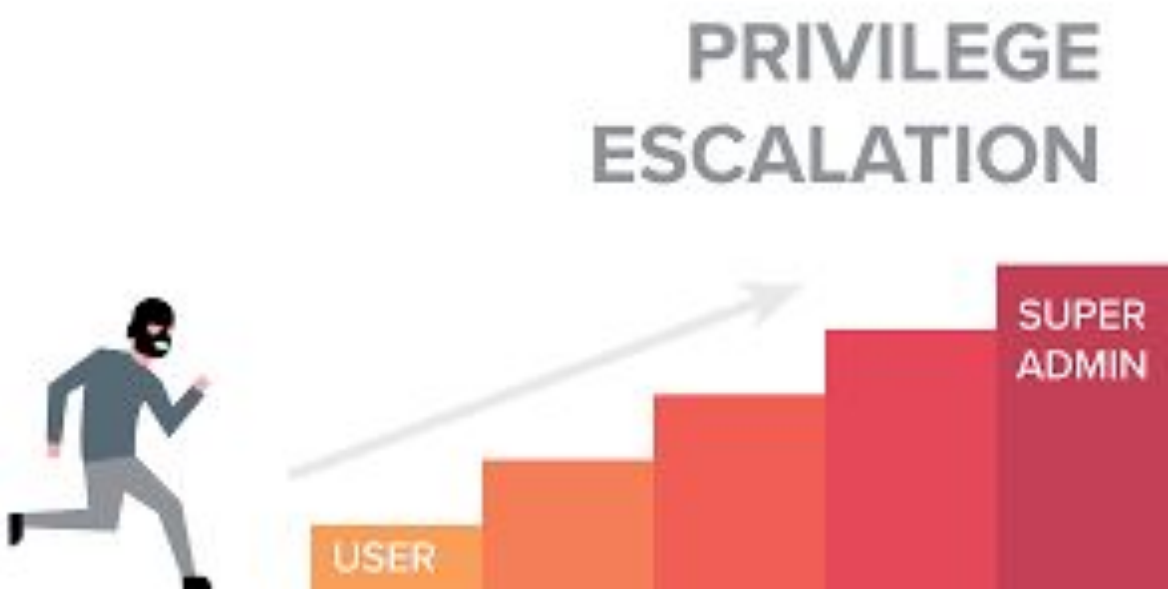
# Tampering

# R


Repudiation


# Information disclosure

# Denial of service




# Escalation of privileges

| OWASP Top 10 - 2017 |
| --- |
| A1:2017-Injection |
| A2:2017-Broken Authentication |
| A3:2017-Sensitive Data Exposure |
| A4:2017-XML External Entities (XXE) |
| A5:2017-Broken Access Control |
| A6:2017-Security Misconfiguration |
| A7:2017-Cross-Site Scripting (XSS) |
| A8:2017-Insecure Deserialization |
| A9:2017-Using Components with Known Vulnerabilities |
| A10:2017-Insufficient Logging & Monitoring |

| OWASP TOP 10 – 2013 |
| --- |
| A1 – Injection |
| A2 – Broken Authentication and Session Management |
| A3 – Cross-Site Scripting (XSS) |
| A4 – Insecure Direct Object References |
| A5 – Security Misconfiguration |
| A6 – Sensitive Data Exposure |
| A7 – Missing Function Level Access Control |
| A8 – Cross-Site Request Forgery (CSRF) |
| A9 – Using Known Vulnerable Components |
| A10 – Unvalidated Redirects and Forwards |

# History

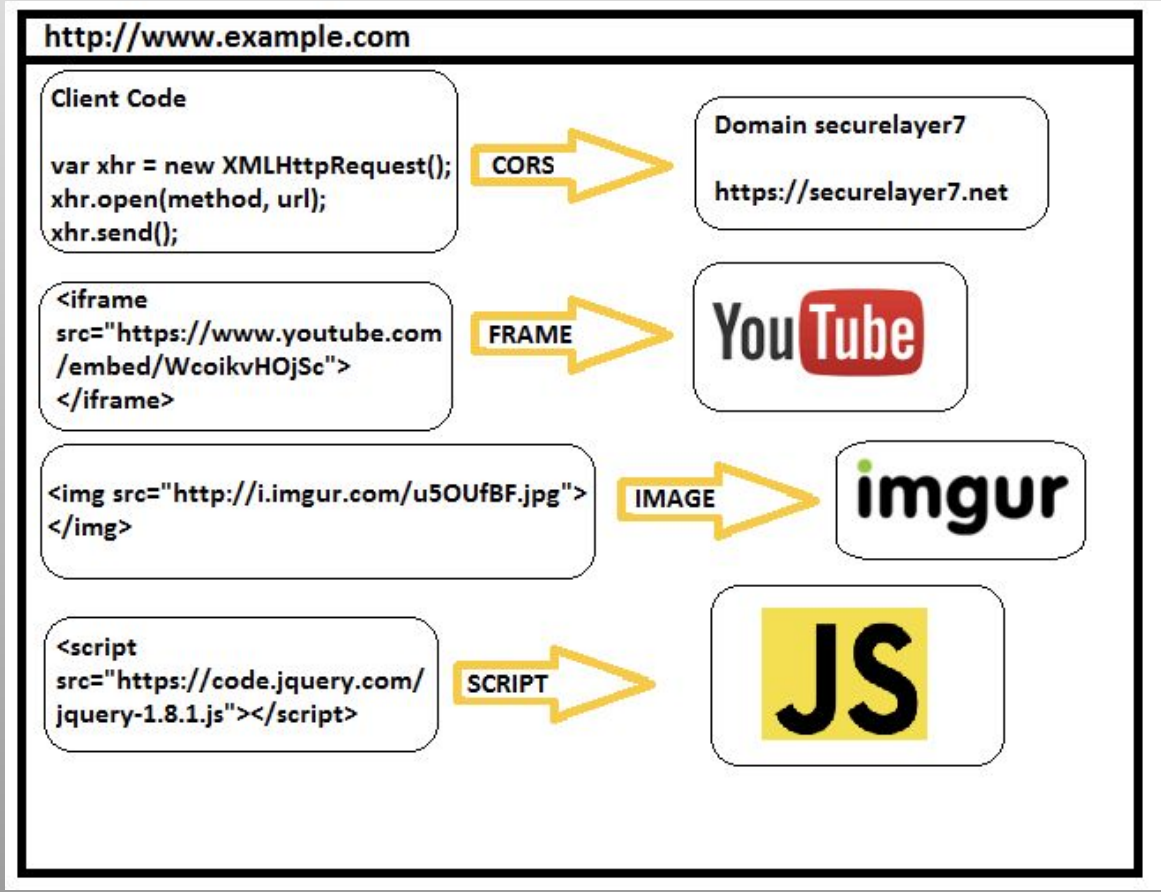
Of OWASP top 10

**When:** Fall of 2001

**Who:** Mark Curphey

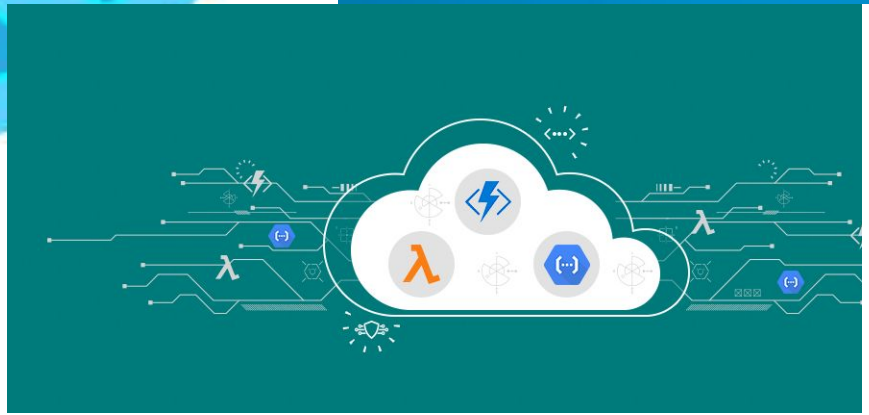
Non profit was started in 2004.

# Cross site scripting (xss)



# Logging and Monitoring


# What do we think this is?



# Serverless

Event Injection
Broken Authentication
Sensitive Data Exposure
Insecure Cloud Configurations
Broken Access Control
Denial of Service (DoS)
Overprivileged Functions
Logic Vulnerabilities
Vulnerable Dependencies
Unhandled Exceptions

# Serverless top 10


# Injection

# Code Injection via API Gateway

Ensure deserialization for the data in the request is secure.

*https://{string}.execute-api.{region}.amazonaws.com/{stage}/order*

# Command Injection via S3 Bucket

Filename

Subject

Messaging Protocol (MQTT)

# Injection in Queue System

Who are the messages coming from?

What are in the messages?

Idempotent?

# BROKEN AUTHENTICATION

Who has seen examples of broken auth?


Incidents found by Verizon data breach report 2019 are Web Application Attacks.

## Web application attack defined as:

Any incident in which a web application was the vector of attack.

This includes exploits of code-level vulnerabilities in the application as well as thwarting **authentication** mechanisms.

# EXAMPLES

Endpoints without authentication.

Identity provider tokens not being verified for expiration.

Missing verification of claims / permissions.

# 10 minute break
# If needed?

# Who deletes data?

# Get all endpoints?

Get all the orders in the database?

Get all the vacation rentals for a homeowner?

Get all reservations?

Get all accounts for the entire company?

# Sensitive Data Exposure

Exposing more data than the user needs to perform a task.

Giving all users admin level access to information.


Insecure cloud configuration

# S3 BUCKETS

Accidently given public access?

Public access

| | Group |
|---|---|
| ○ | Everyone |

# Capital one incident

Rumor has it, the S3 bucket was not the issue.
By: J Cole Morrison

Misconfigured firewall, waf, security group etc.

IAM Role allowed for S3 access to those 700+ Buckets

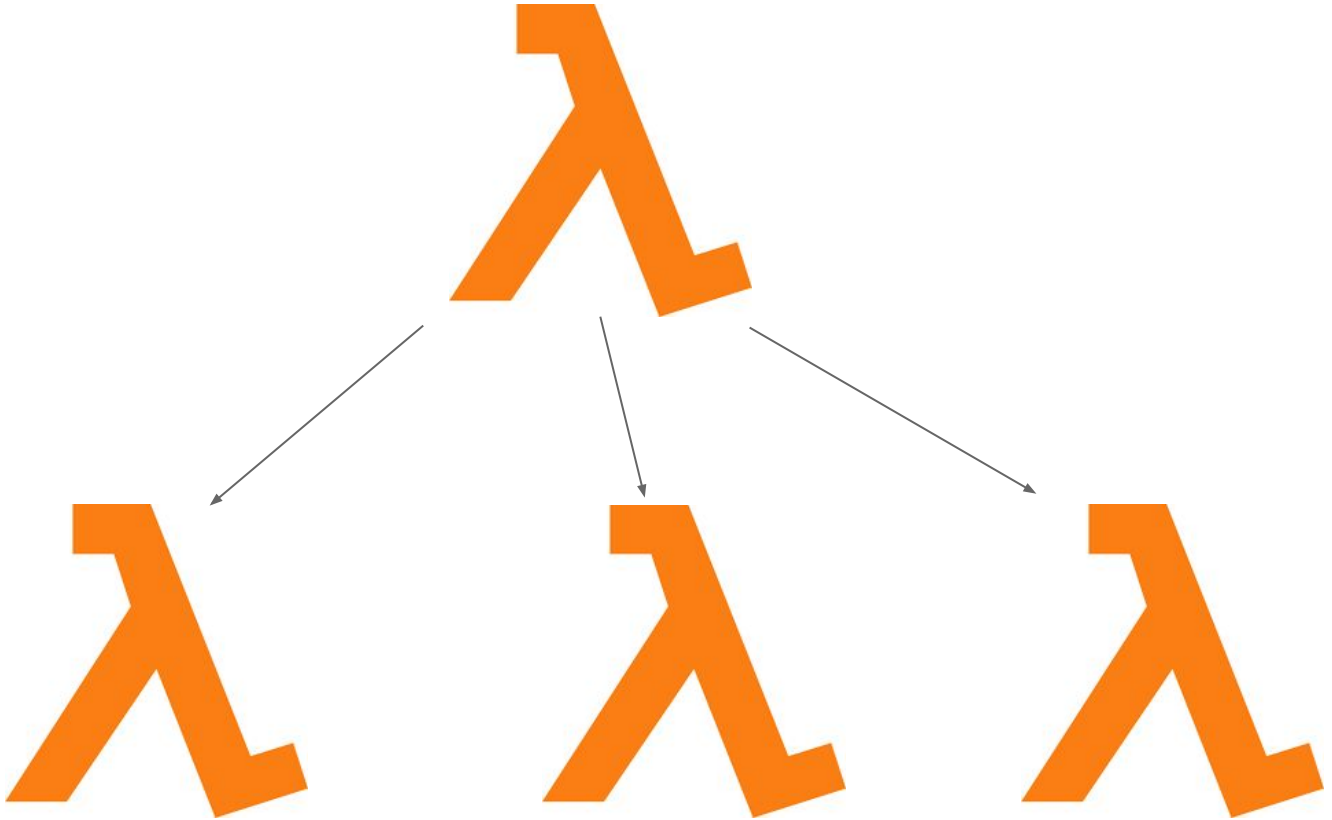
# SQS, Dynamo

Public? Encrypted?

Both of these serves depending on if the team is using Terraform, Cloudformation or Serverless framework.

Have bad defaults.

Lambda functions



How many lambdas does a lambda have permission to invoke?

@securelykash

# Broken Access Control

Limit access for all functions.

Follow least privilege.

Single IAM role for each lambda function.

# Overprivileged functions

Protego Labs “has found that almost all functions are configured with more permissions than what they actually need.”

Look at teams Terraform, cloudformation, serverless.yml

# Logic Vulnerabilities

When a system allows an attacker to make requests out of order for an established workflow.

Time of check to time of use (TOCTOU), also known as a race condition.

⚠ **We found potential security vulnerabilities in your dependencies.** Dismiss

Some of the dependencies defined in your `package-lock.json` have known security vulnerabilities and should be updated.

**Review vulnerable dependencies**

Only the owner of this repository can see this message.
Learn more about vulnerability alerts

# Vulnerable dependencies


Leaking stack traces, sensitive info, primary keys, etc.

# Unhandled exceptions

(error messages)

Not exposing queries, code, or information.

Test and review all error messages.

Debug mode

# Unsanitized Logging

Not exposing queries, code, or information.

Review all log messages.

Not exposing personal information.

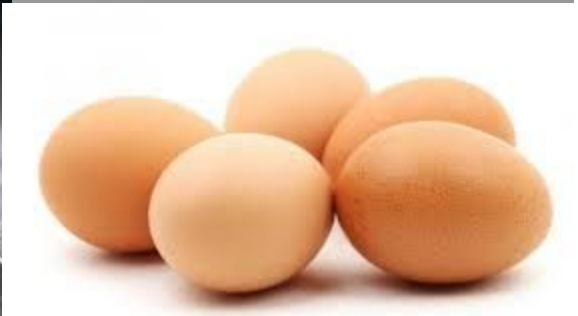
# Subdomain takeover

"A" records that point at Elastic IPs you no longer own.

{attacker choice}.domain.com

# 5-10 minute
# Break if needed


Who likes chocolate chip cookies?
@securelykash






# Ingredients

# Ingredients

# Dry ingredients

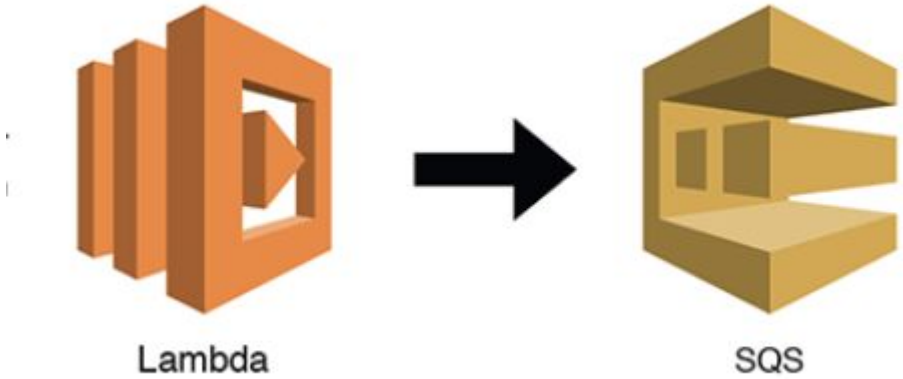
Flour

Baking Powder

# AWS model

Single lambda

SQS Queue



# Primary ingredients

Butter

Eggs

Vanilla

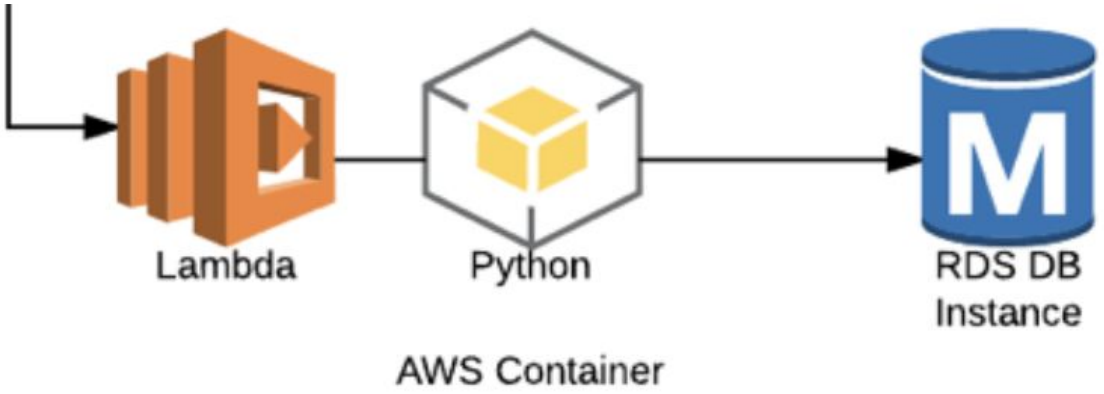
Brown Sugar

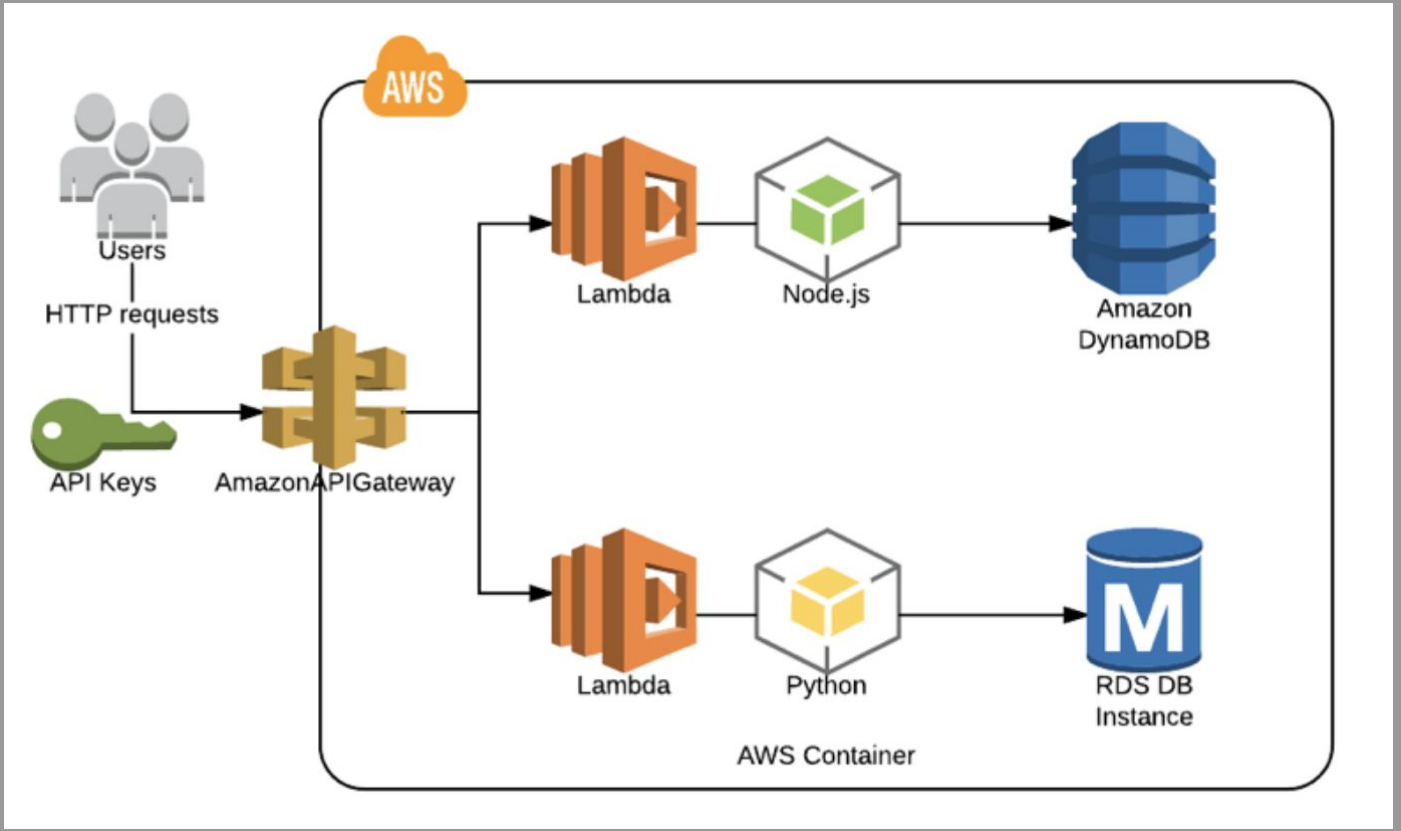
# AWS model

API Gateway

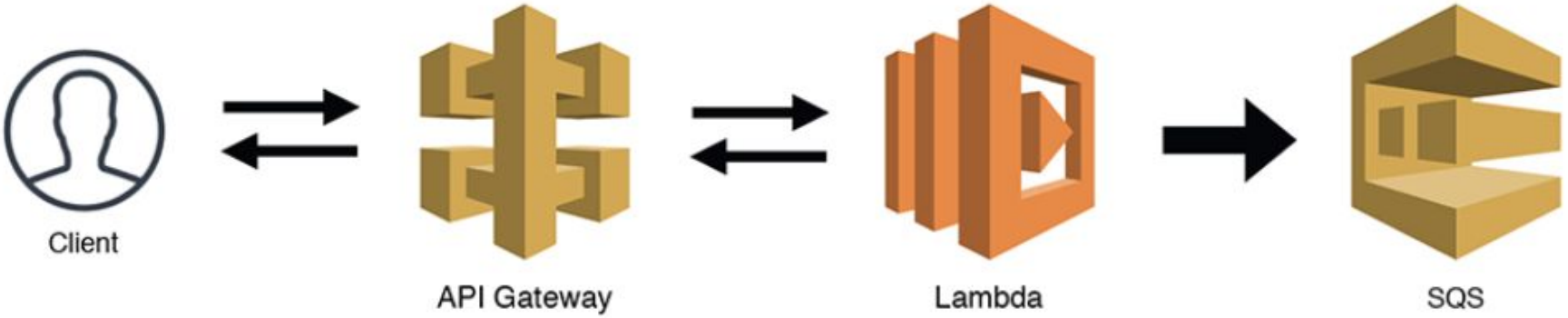
Single lambda

SQS Queue



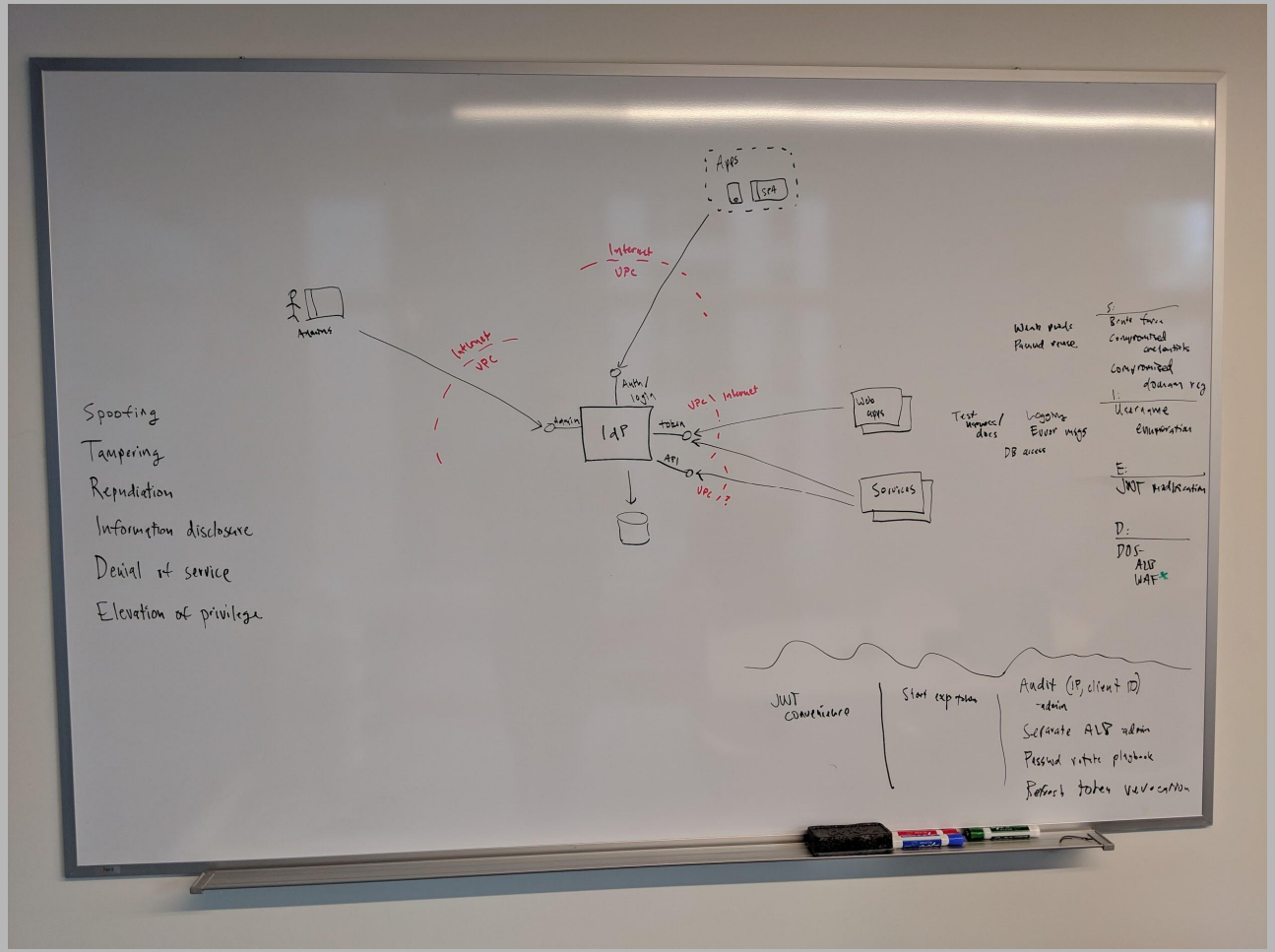



# System Level - all the ingredients



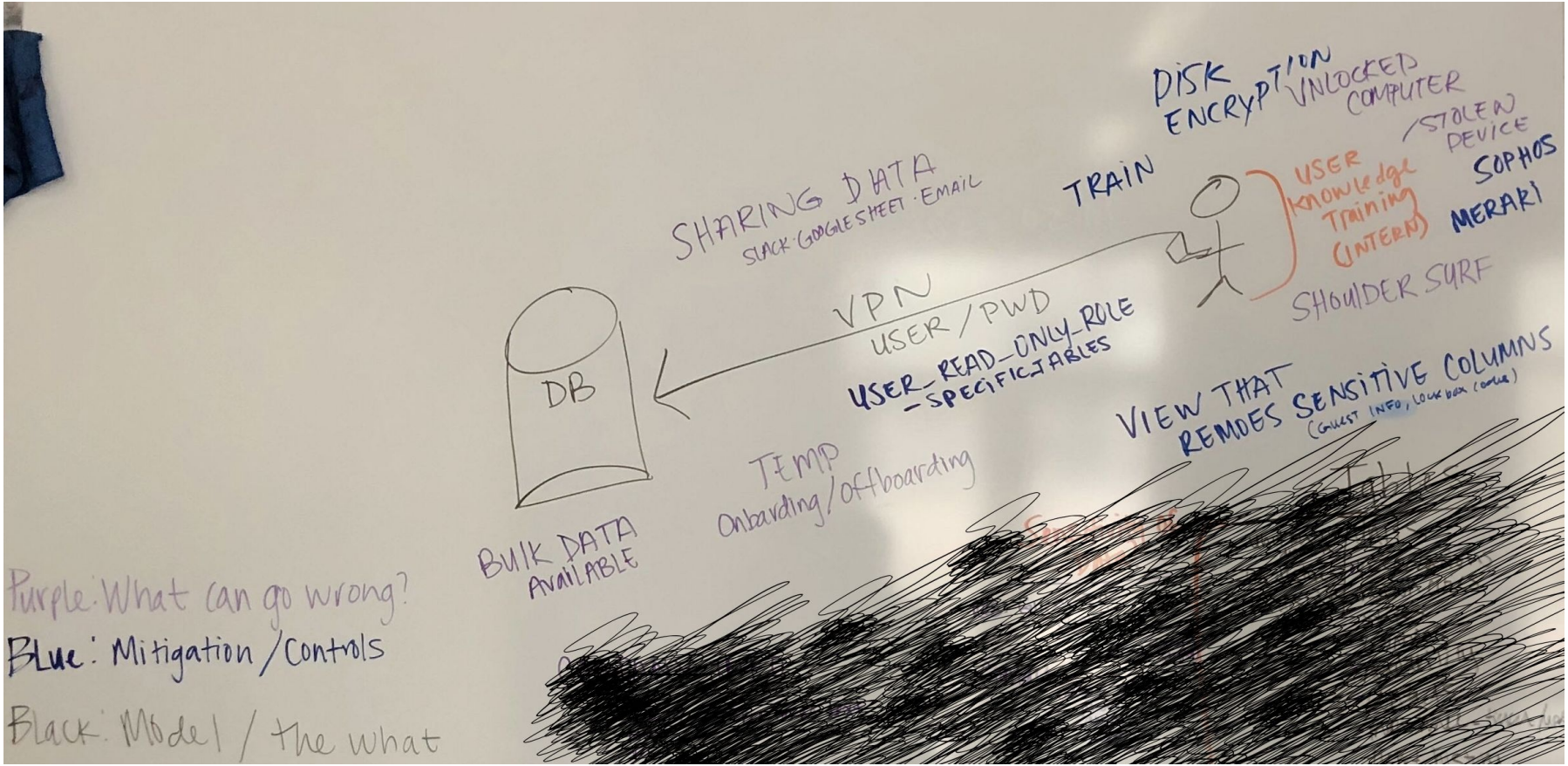

# System Level - all the ingredients

# Example Diagram




@securelykash

# Example Diagram



# Models

System Level

Feature

Epic

Domain

"Representation of a system"

# Threat + Model

"Threat Modeling
Is about using
models to find
security problems"

A way to optimize security by identifying threats, vulnerabilities and risks, working towards mitigating or preventing them.

# WHEN

Anytime

Design Phase

Every Major Architecture Change

Go Live

Epics

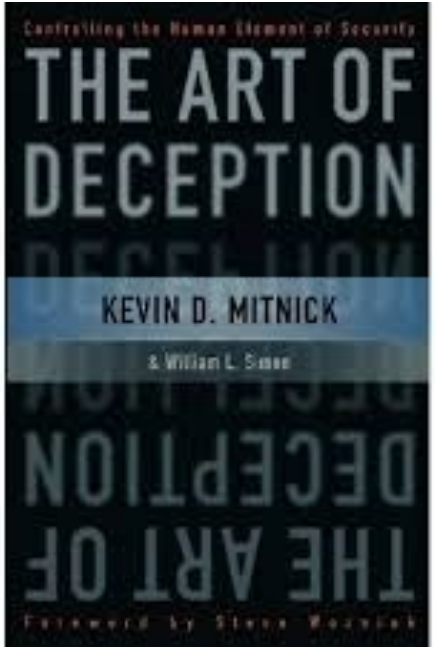

What data is at stake (specific elements)

# What are you building?




# What can go wrong?

@securelykash




# Phishing is more than email...Phones and forgotten passwords


Model
Found threats
Discovered Vulnerabilities
@securelykash

# Mitigation

Refers to policies and processes put in place by companies to help prevent security incidents and data breaches as well as limit the extent of damage when security attacks do happen.

# Mitigation

Reducing impact and likelihood to be discovered by an outsider.

Never trust input or make any assumptions about its validity.

Always validate or sanitize input.

Never pass user input directly to any interpreter.

**Nutrition Facts** Serv. Size: 1 oz (28 g/1 oz), Servings: 8, Amount Per Serving: **Calories** 100, Fat Cal. 35, **Total Fat** 4g (6%DV), Sat. Fat 0g (0%DV), *Trans* Fat 0g, **Cholest.** 20mg (7%DV), **Sodium** 35mg (1%DV), **Total carb.** 15g (5%DV), Fiber 1g (4%DV), Sugars 8g, **Protein** 3g, Vitamin A (0%DV), Vitamin C (0%DV), Calcium (4%DV), Iron (2%DV). Percent Daily Values (DV) are based on a 2,000 calorie diet.

INGREDIENTS: Gluten Free Flour:( Brown rice flour, sweet rice flour, tapioca starch, cornstarch, potato starch), Pure Cane Sugar, Fresh Eggs, Almond Flour, Almonds, (Dried Unblanched), Cranberries (Dried Sweetened), Fresh Orange Juice, Baking Powder; Aluminum Free(Sodium Acid Phosphate), Pure Vanilla Extract (water, alcohol, vanilla extractives), Fresh Orange Peel, Almond Extract

Contains: Eggs, Almonds

# What if I was allergic to peanuts(or event injection)?

# Mitigation by Thorough logging

Logging of API access keys related to successful/failed logins (**authentication**)

Attempts to invoke serverless functions with inadequate permissions (**authorizations**)

Successful/failed deployment of new serverless functions or configurations (**change**)

Changes to function permissions or execution roles (**change**)

Examples:

Validate all input.

Review all error messages.

## Requirements...and product

# Making threat modeling fun!



Card games with the whole team!

# Escalation of privileges

By: Adam Showstack

Start with a model of the system.

Facilitator, notes keeper and team.

Follow instructions.

# Escalation of Privilege

Example cards.

Available on Github



# Escalation of privileges

Simplified version

Play like it was UNO.

Match the color.

Discuss each threat (card)

# Time

How long does it take to play?

Recommend blocking out 2 hours for a new application.

# Take 10 minutes to look at the serverless Goat app

# Some ideas

Individually look at the serverless goat https://www.serverless-hack.me/

XSS

Open web inspect in the browser

Upload a file

Upload a fake file

# What did you find?

## Response Headers

| | |
|---|---|
| X-Firefox-Spdy | h2 |
| content-length | 985 |
| content-type | application/json |
| date | Thu, 20 Feb 2020 02:10:52 GMT |
| x-amz-apigw-id | ILGa7GGBvHcFbnw= |
| x-amzn-requestid | c0661e76-491f-45c9-8b12-f3c1268c753a |
| x-amzn-trace-id | Root=1-5e4deaac-5cdbd57ba925ec64960f6440 |

## Request Headers

# OWASP ServerlessGoat

OWASP ServerlessGoat is a deliberately insecure realistic AWS Lambda serverless application, maintained by OWASP.

Enter a URL of a Word Doc (.doc) file to convert:

https://foo.com; env

Submit

Try to find as many issues as possible, from the following list:

# Break up into groups of 5

# Security UNO

Demonstrate

Use previous discovered threats.

OWASP top 10

Serverless top 10

Play the same number or color.

# Tools

Review of things we covered

Security Uno

Threat Model

**Automation**

# Automation

Many options!

CLoudMapper by Duo-Labs

Vacabot (internal)

Devops Report (internal)

# Cloudmapper

Duo-Labs
https://github.com/duo-labs/cloudmapper

Open Source!

Scan all your aws accounts, produces a nice HTML report.

Slack web hooks.

# Contents



| Firehose streams | Glacier vaults | KMS keys | Lambda functions |
|---|---|---|---|
| 7 | 0 | 20 | 4 |
| 0 | 0 | 6 | 29 |
| 0 | 0 | 7 | 545 |
| 0 | 2 | 19 | 4 |
| 0 | 0 | 8 | 6 |
| 0 | 0 | 7 | 7 |
| 0 | 1 | 13 | 15 |
| 1 | 0 | 9 | 19 |

Example findings from the report

# Vacabot

Internal tool built with Golang

Scans every PR

Look for key words and add a *Security label*

Tag Security team as a reviewer.

**Vacabot** added the **security** label 19 days ago

**Vacabot** requested a review from **Vacasa/security** 19 days ago

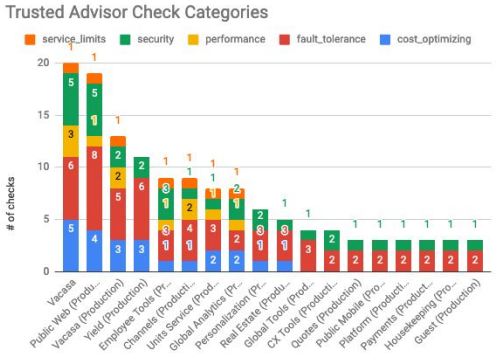
**Vacabot** added the **change/terraform** label 2 days ago

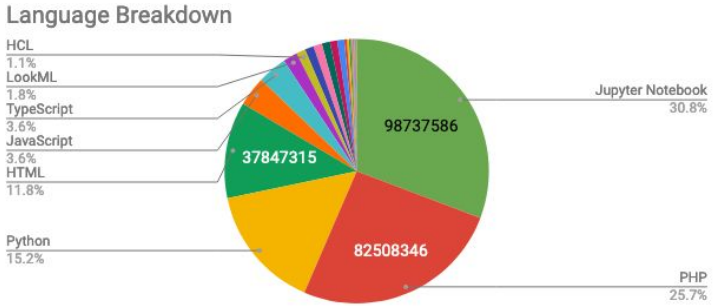
**Vacabot** commented 2 days ago

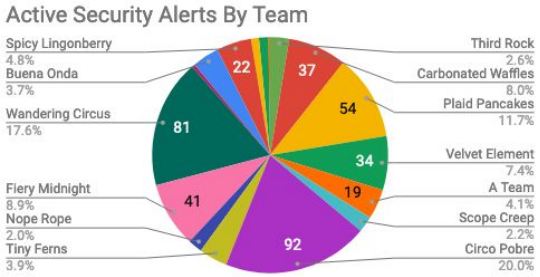
## Pull Request Audit Notes

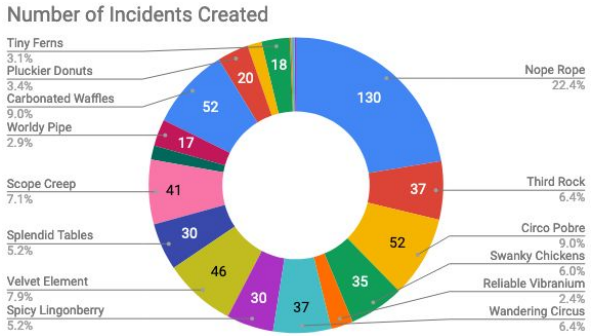
> This PR changes terraform files.
>
> - When testing this PR, run terraform plan and verify the results are what you expect.
> - When releasing this PR, inform team DevOps members that terraform changes need to run prior to deploying this PR.

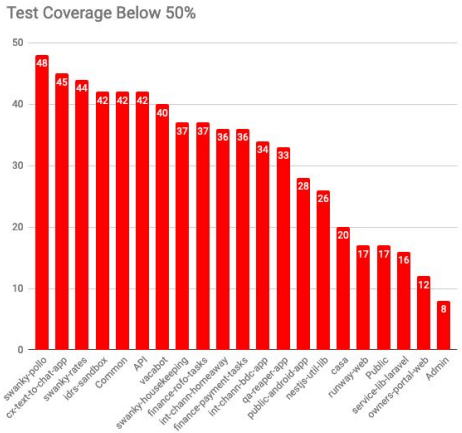
# Build a "Health of the Org" Report

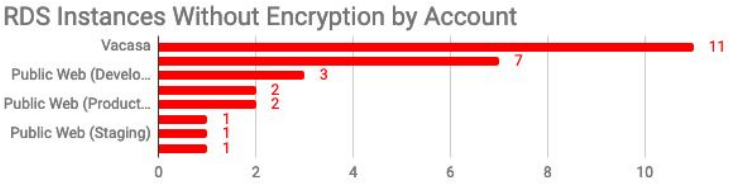



You need top-down buy in.

# Security should be fun and easy

All automation is easily consumable. Security **helps** teams find solutions

# Review

Things to remember!

Everyone can threat model on the team!

Make it fun and a habbit.

Automate where you can!

# Questions?

# References and Callouts

https://vacasait.atlassian.net/wiki/spaces/ENG/pages/827719808/Secure+Software+Development+Life+Cycle

https://cheatsheetseries.owasp.org/cheatsheets/Access_Control_Cheat_Sheet.html

https://github.com/OWASP/DVSA/blob/master/AWS/LESSONS/README.md

https://www.protego.io/protego-labs-finds-nearly-all-serverless-application-functions-at-risk/
https://start.jcolemorrison.com/the-technical-side-of-the-capital-one-aws-security-breach/

# Contact

https://github.com/kendraash

kendra.ash32@gmail.com