RSAConference2019
San Francisco | March 4–8 | Moscone Center

SESSION ID: ASD-W03

Security Precognition: Chaos Engineering in Incident Response

Aaron Rinehart
Chief Technology Officer
Verica.io
@aaronrinehart

Kyle Erickson
Director of IoT Security
Medtronic

“Resilience is the story of the outage that never happened.”

- John Allspaw

@aaronrinehart

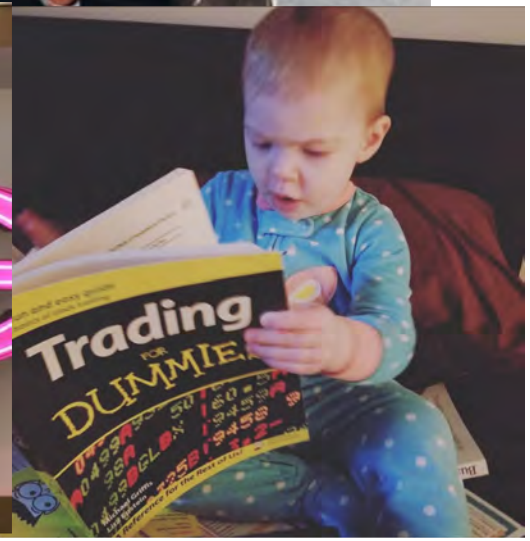
Security PreCognition

About A.A.Ron

- CTO of Stealthy Startup in Chaos Engineering
- Former Chief Security Architect @UnitedHealth responsible for strategy
- Led the DevOps and Open Source Transformation at UnitedHealth Group
- Extensive enterprise experience (DOD, NASA, DHS, CollegeBoard)
- Frequent speaker and author on Chaos Engineering & Security
- Pioneer behind Security Chaos Engineering
- Led ChaoSlingr team at UnitedHealth

About Kyle E>

- Cybersecurity Director of IOT Security @ Largest Medical Device Company in the World
- Former Director of Security Incident Response @UnitedHealth
- Former Director of Cloud and Application Security Engineering @Optum
- Startup and enterprise experience as an engineer and leader
- Build and break all the things!
- Wannabe financial analyst
- Baseball stadium aficionado



In this Session we will cover

Takeaways

★Problems with Complex Adaptive Systems

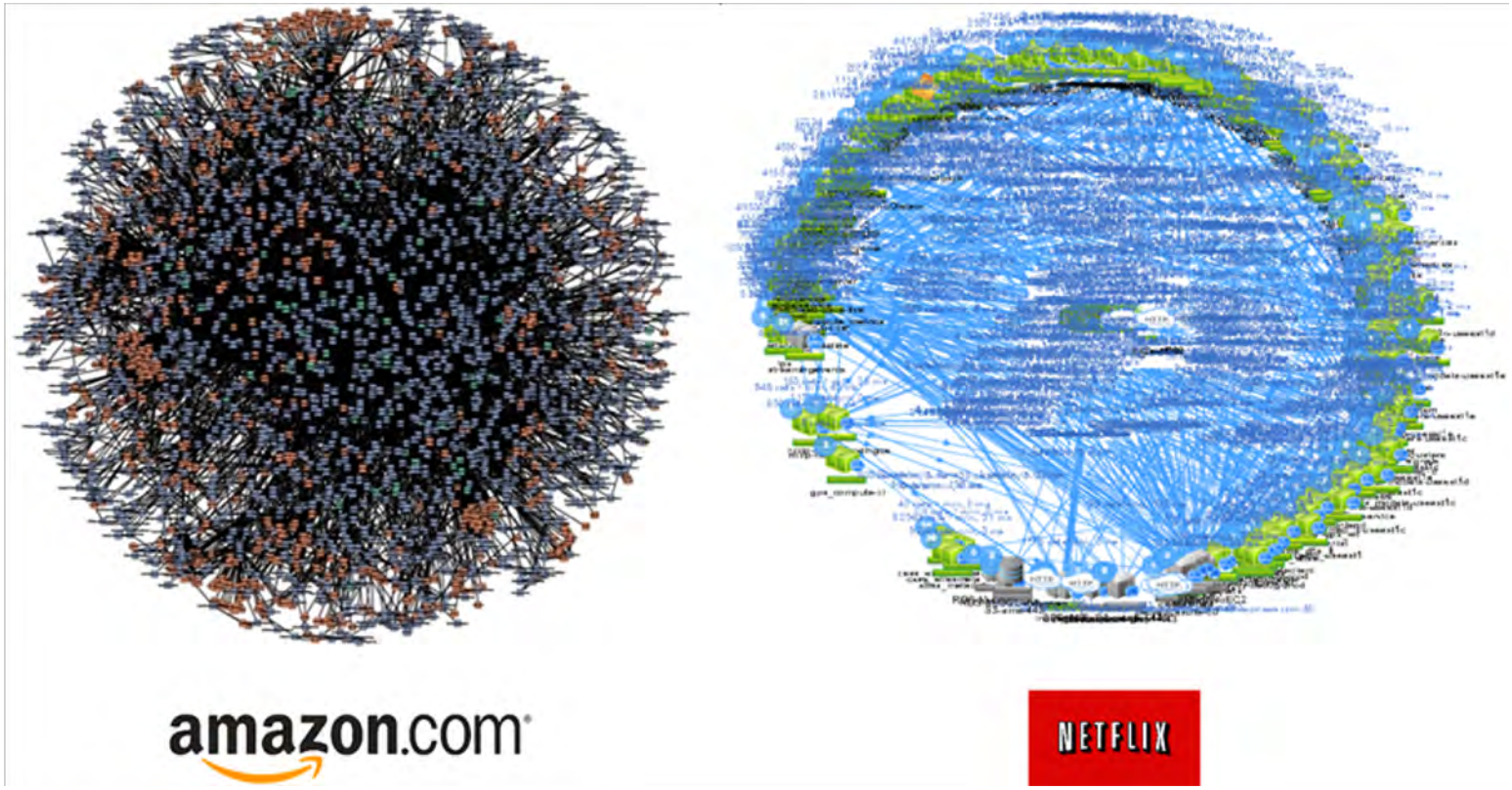
Takeaways

- ★Problems with Complex Adaptive Systems
- ★Security Chaos Engineering

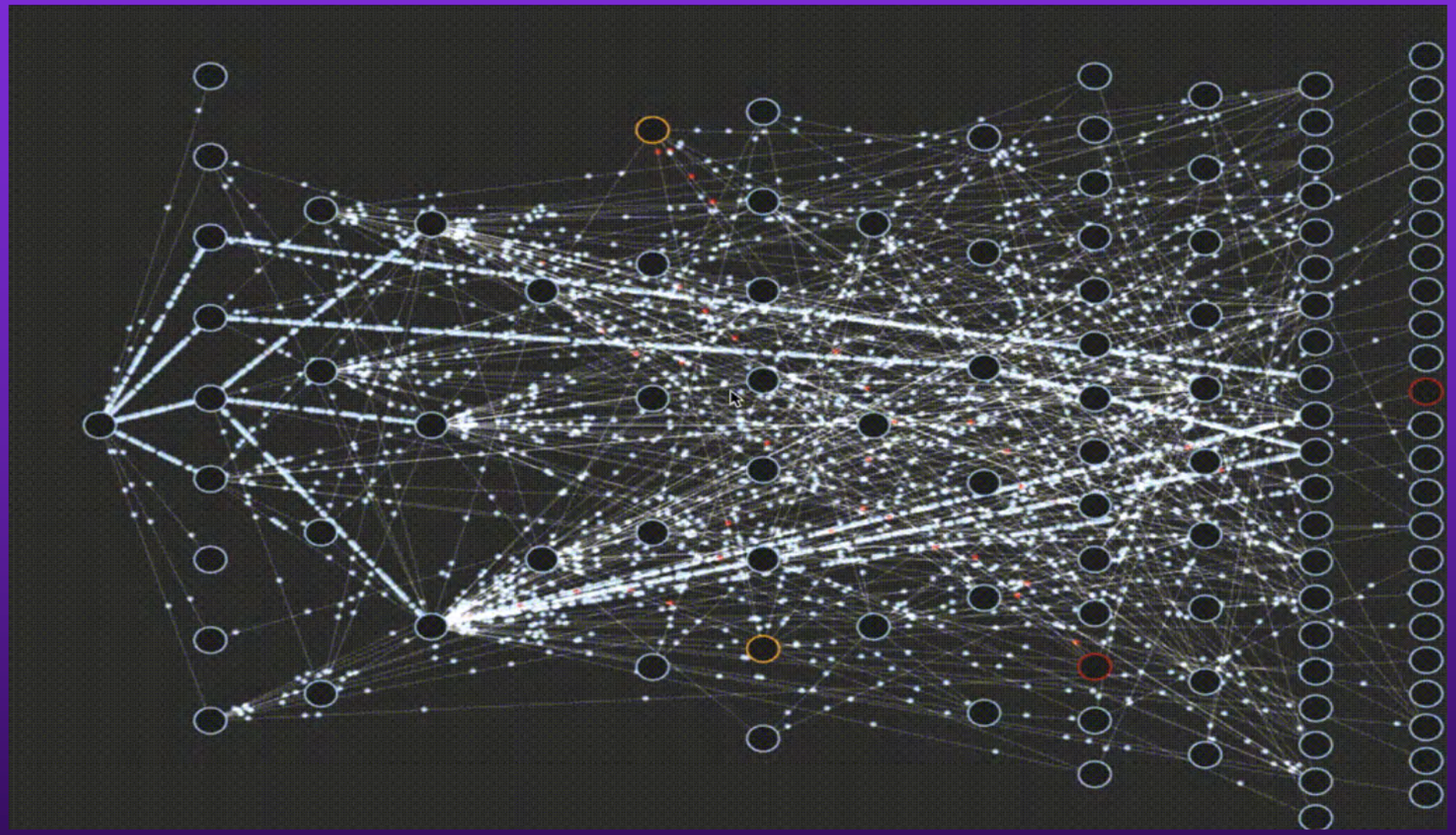
Takeaways

- ★Problems with Complex Adaptive Systems
- ★Security Chaos Engineering
- ★How it works: Security IR Use Case

Our systems have evolved beyond human ability to mentally model their behavior.

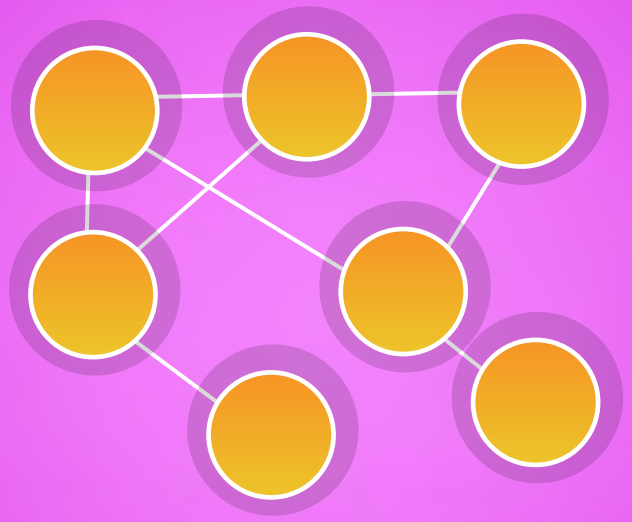


A Complex Dynamic Problem

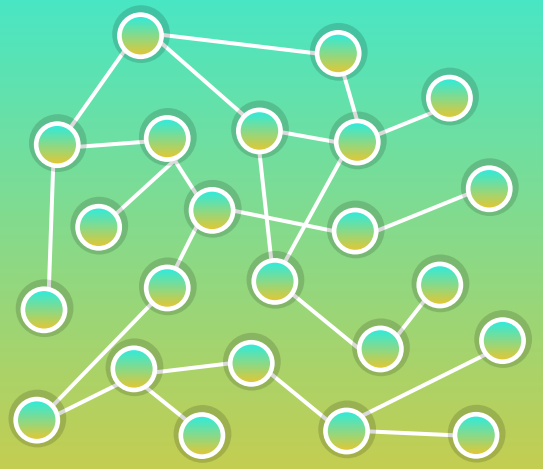


Evolution of Modern Architecture

Monolith



Microservices



Functions

Complex?

Continuous Delivery

Distributed Systems

Microservice Architectures

Automation Pipelines

Blue/Green Deployments

Containers

DevOps

Continuous Integration

Immutable Infrastructure

Infracode

CI/CD

Cloud Computing

Service Mesh

API

Auto Canaries

Circuit Breaker Patterns

Security?

Mostly Monolithic

Expert Systems

Stateful in nature

Prevention focused

Adversary Focused

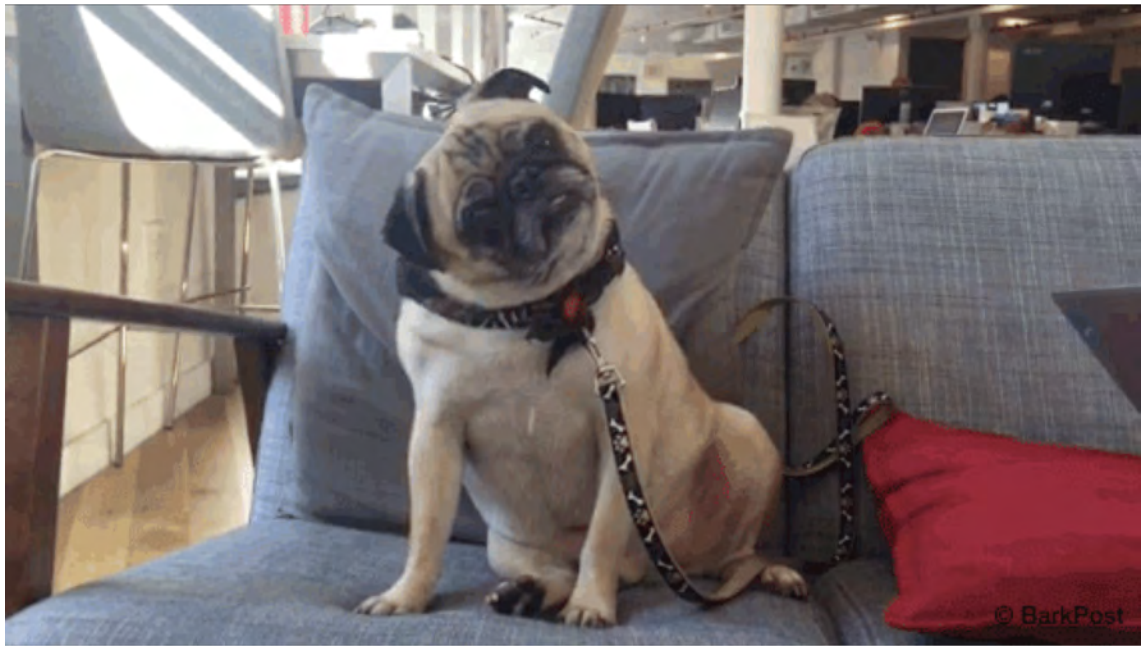
Poorly Aligned

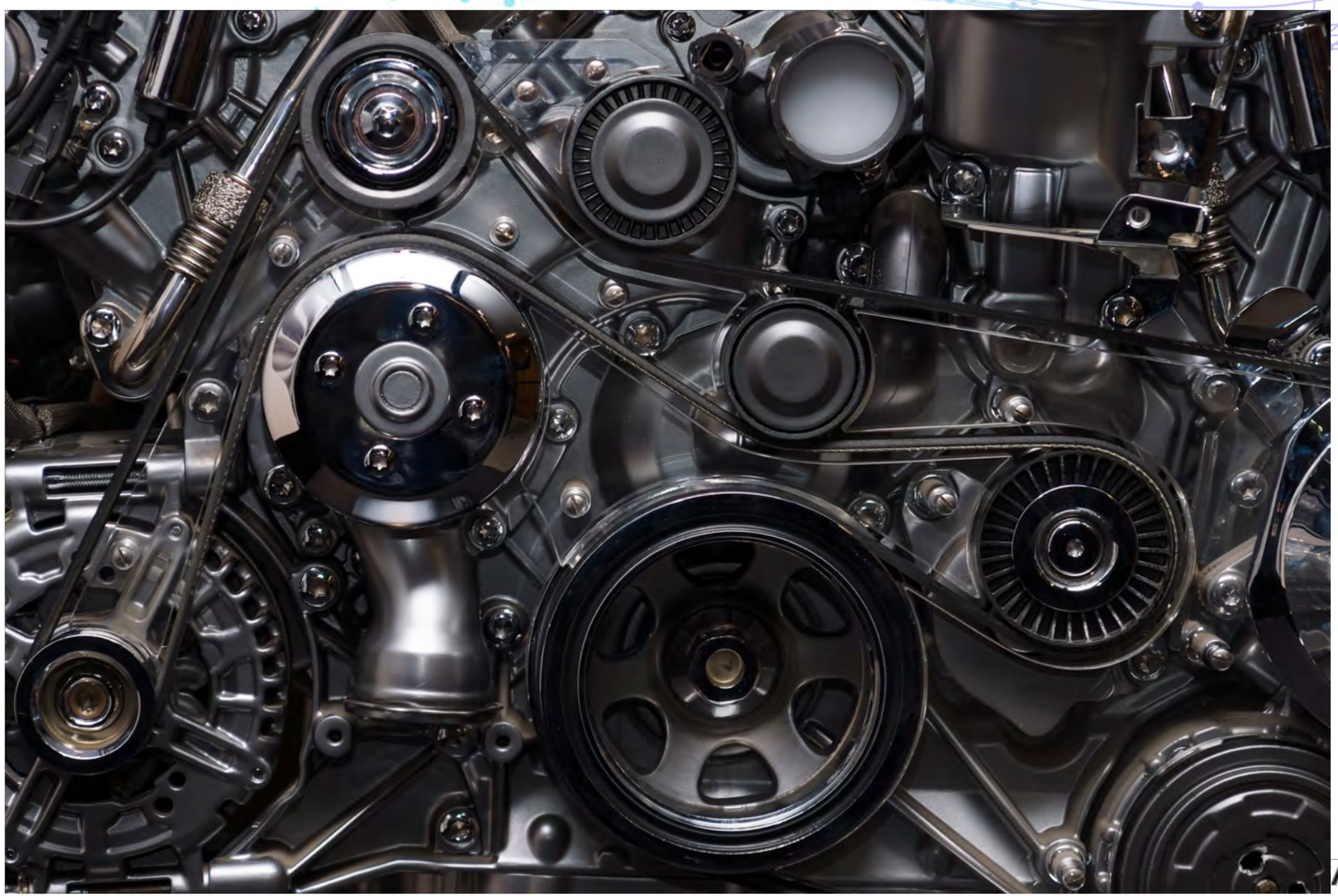
Defense in Depth

Requires Domain Knowledge

DevSecOps not widely adopted

Simplicity?





Complexity

How well do you really understand how your system works?

Stella Report

STELLA

Report from the SNAFUCatchers Workshop on Coping With Complexity

Brooklyn NY, March 14-16, 2017

Winter storm STELLA

Woods' Theorem: *As the complexity of a system increases, the accuracy of any single agent's own model of that system decreases.*

© 2017 DD Woods

http://stella.report

Year-long project

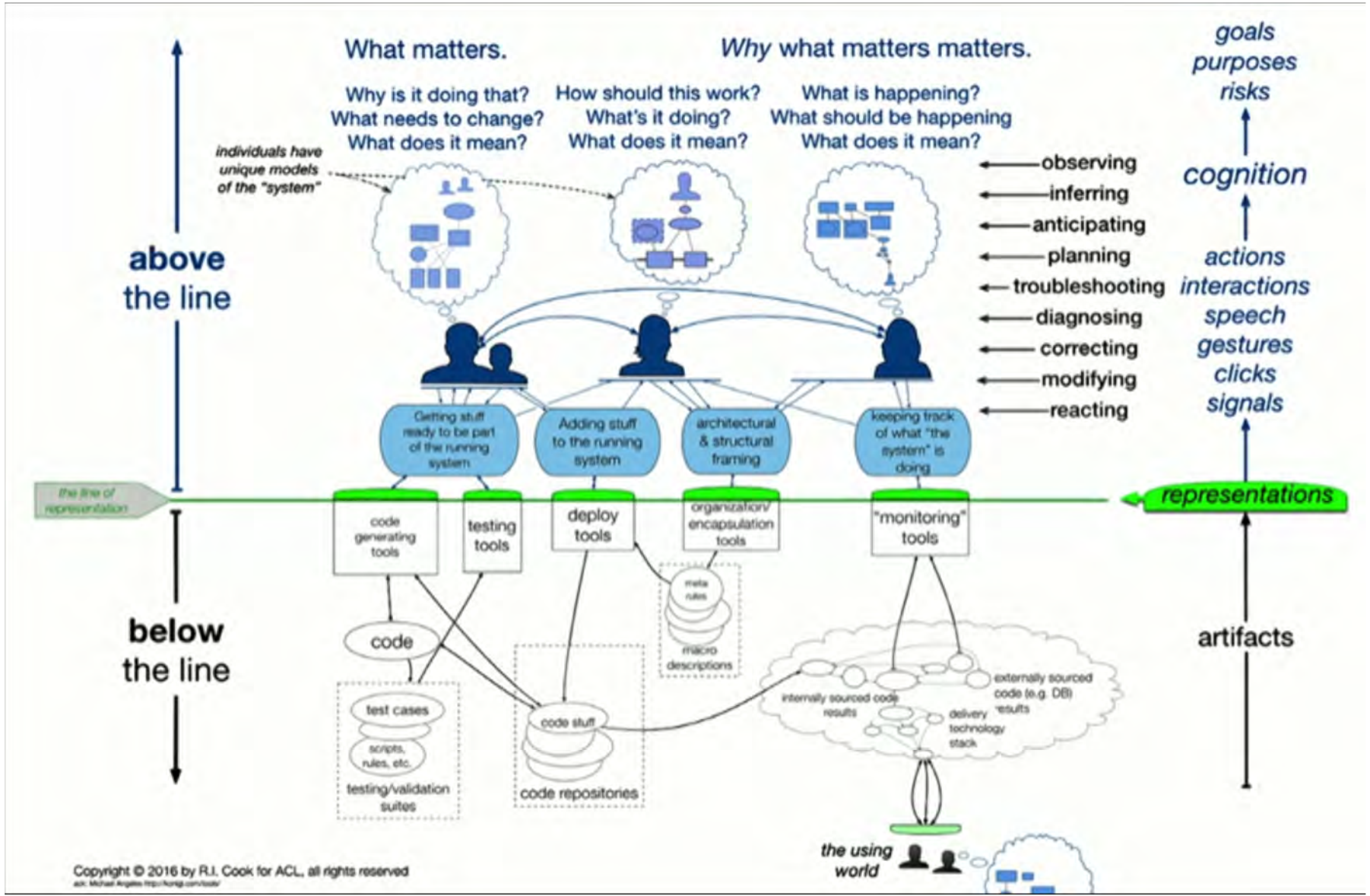
Researchers analyzed 3 incidents, at:

IBM iex Etsy

Six Themes

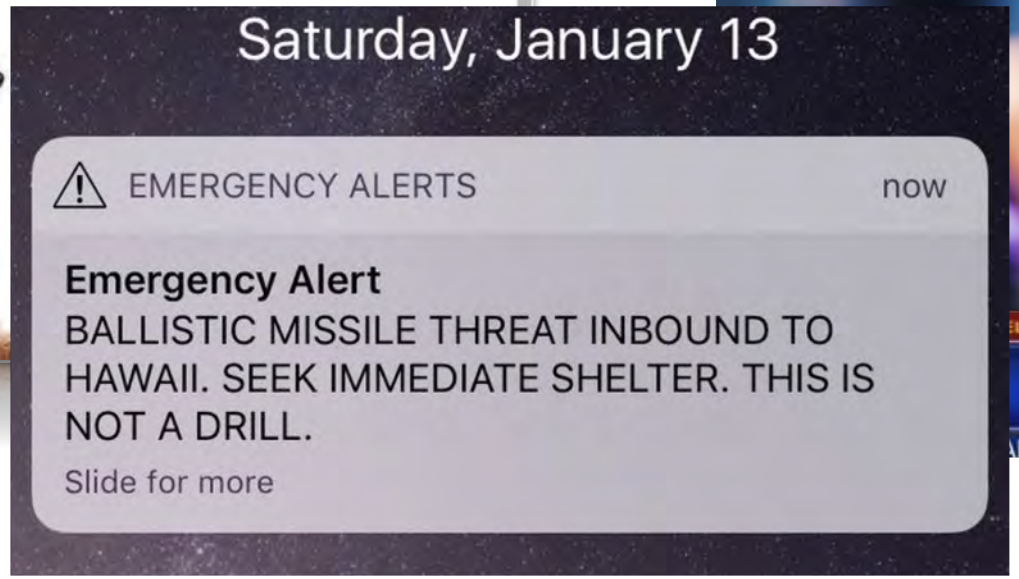
- Postmortems as re-calibration
- Blameless v. sanctionless after action actions
- Controlling the costs of coordination
- Visualizations during anomaly management
- Strange Loops
- Dark Debt

System Mental Model



So what does all of this have to do with Security?

Failure Happens.



Security Incidents & System Outages are Costly

Security Incidents are **Subjective** in Nature

We really don't know **very much**

Where?

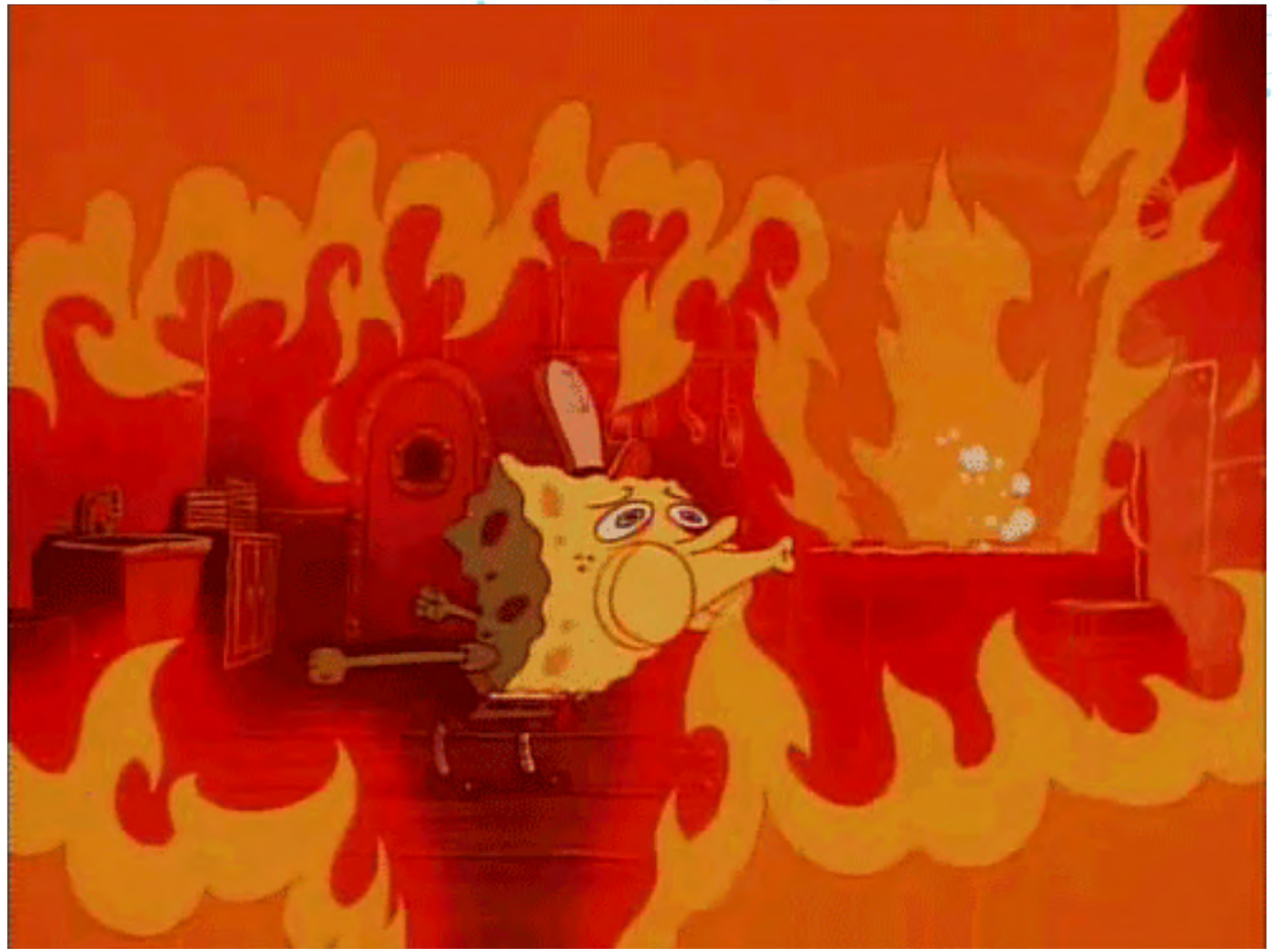
Why?

Who?

How?

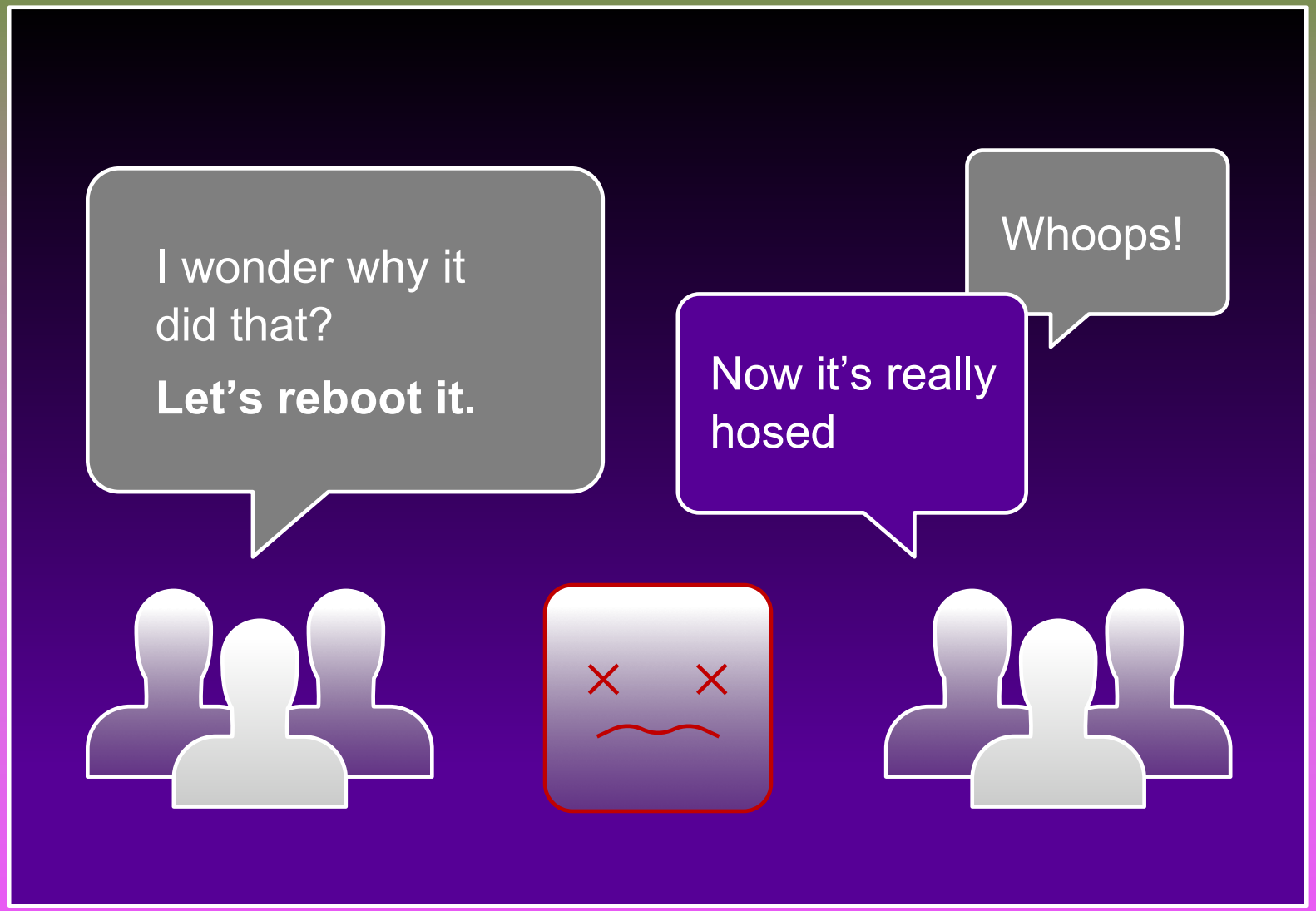
What?

Let's face it, when outages happen…..



Teams spend **too much time reacting to outages** instead of **building more resilient systems**.

Unexpected application behavior often causes people to intervene and make the situation worse



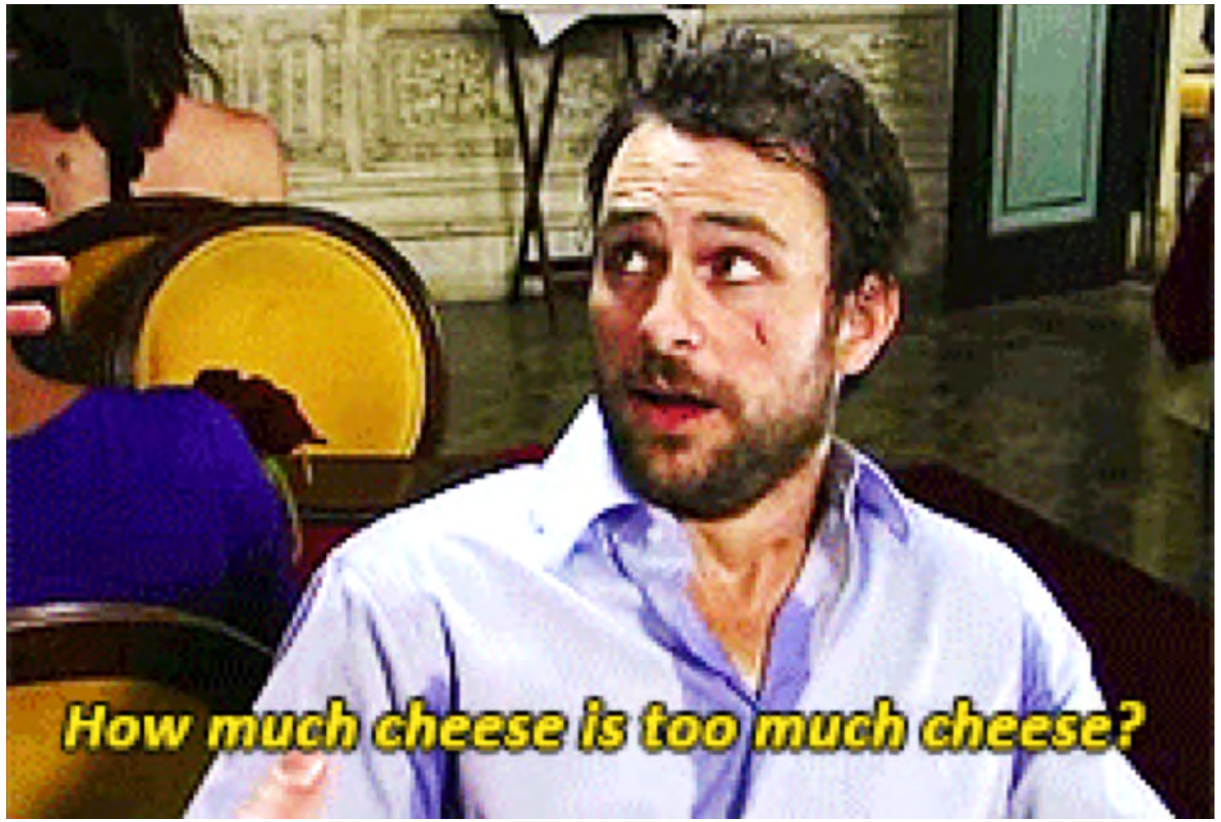
"Response" is the problem with Incident Response

Ring Ring!

Conditioned?

War Rooms

True Cost



IR Success


Preparation
Detection & Analysis
Containment Eradication & Recovery
Post-Incident Activity


Blame
Solution

"Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's ability to withstand turbulent conditions"

NETFLIX

NETFLIX

Who is doing Chaos?

PRINCIPLES OF CHAOS ENGINEERING

Last Update: 2017 April

Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system's capability to withstand turbulent conditions in production.

O'REILLY®

Compliments of NETFLIX

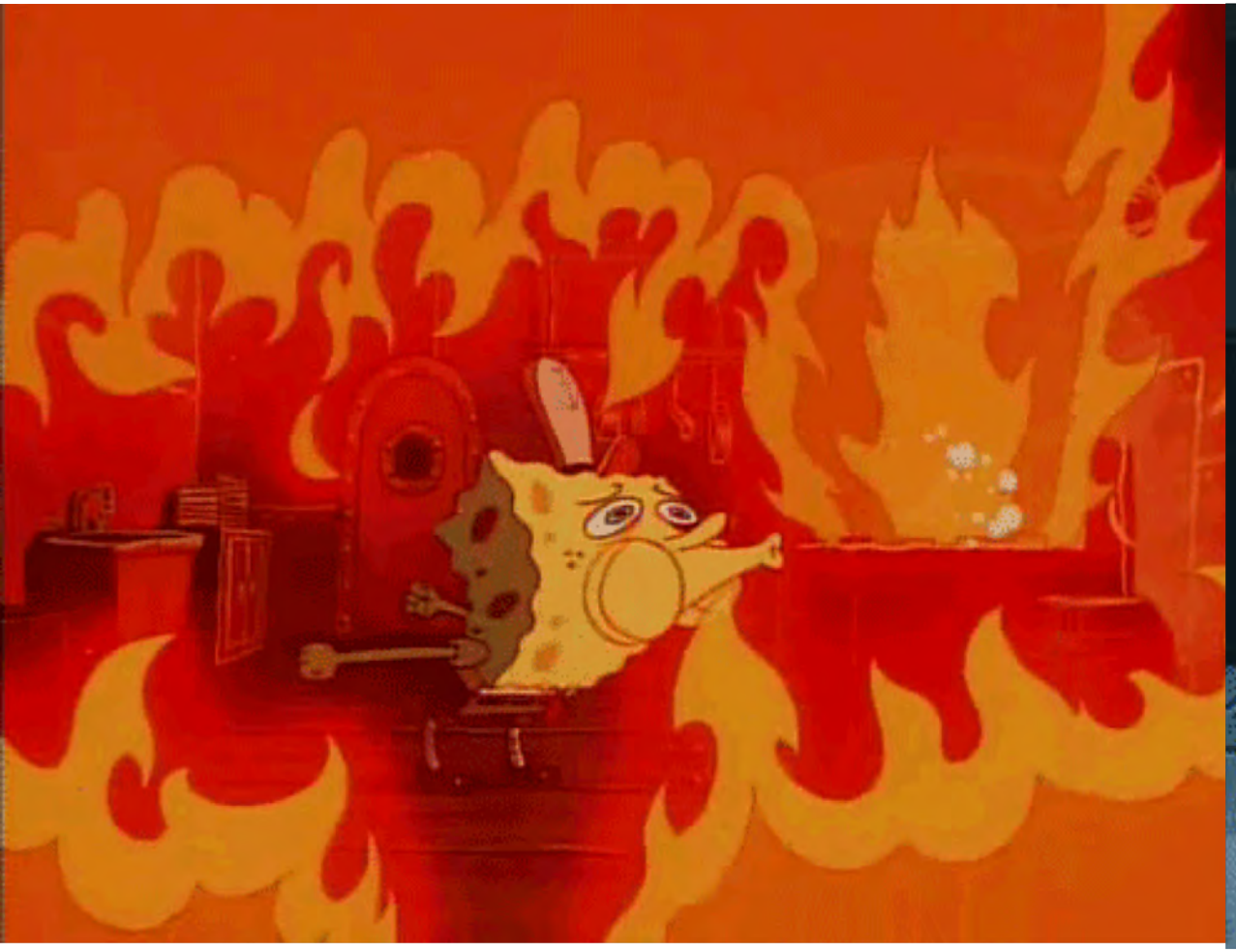
Chaos Engineering

Building Confidence in System Behavior through Experiments

Security

Engineering

People Operate Differently when they expect things to fail




NFL

The Normal Condition of a Human & Systems they Build is to

Fail

We need failure to learn & Grow

Let's Flip the Model

Post Mortem = Preparation

Create Order through Chaos

Use
Chaos Engineering
to initiate
Objective Feedback Loops
about Security
Effectiveness

Proactively Manage & Measure

Validate Runbooks
Measure Team Skills
Determine Control Effectiveness
Learn new insights into system behavior
Transfer knowledge
Build a learning culture

Testing vs. Experimentation

Security Crayon Differences

Noisy distributed system behavior

Not geared for Cascading Events

Point-in-time even if Automated

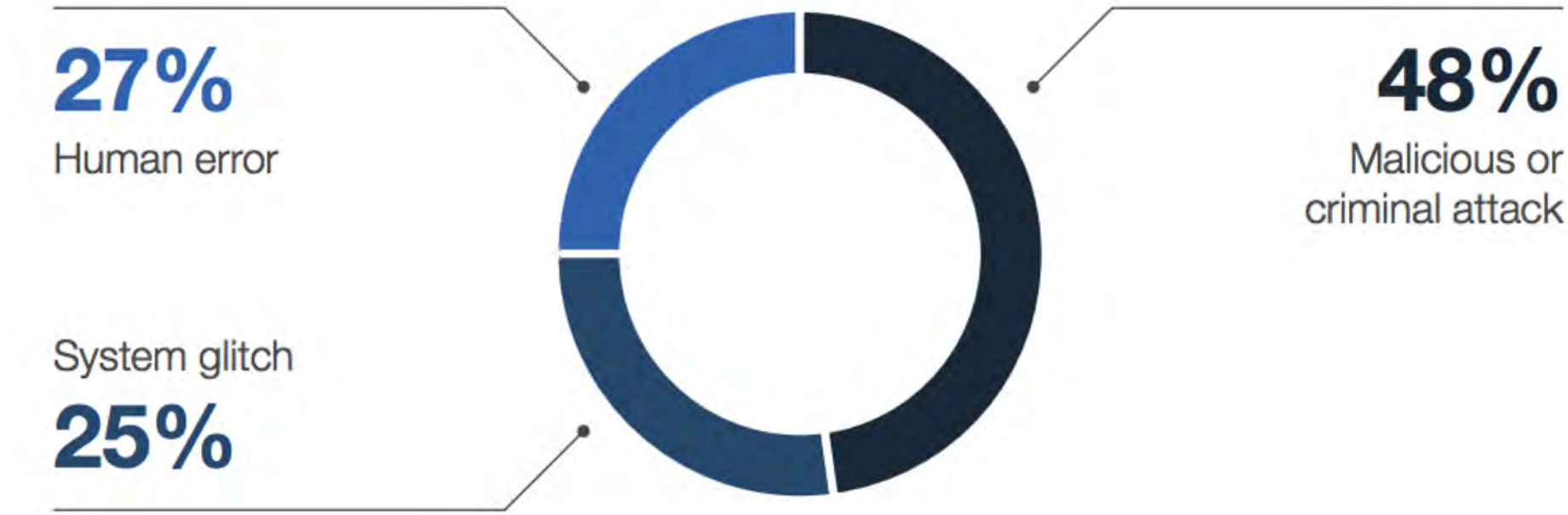
Performed by Security Teams with Specialized skill sets

Security Chaos Differences

Distributed Systems Focus
Goal: Experimentation
Human Factors focused
Small Isolated Scope
Focus on Cascading Events
Performed by Mixed Engineering Teams in Gameday
During business hours

2018 Causes of Data Breaches

Ponemon INSTITUTE



Proactively Manage & Measure

Incidents vs. *Chaos*

Uncontrolled	Controlled
Unpredictable	Scheduled
Time to Detect: Minutes	0 Time to Detect
Time to Resolve: ????	Time to Resolve: seconds
Analysis Time: ????	Root Cause Analysis: Intentional

Continuous SECURITY Validation

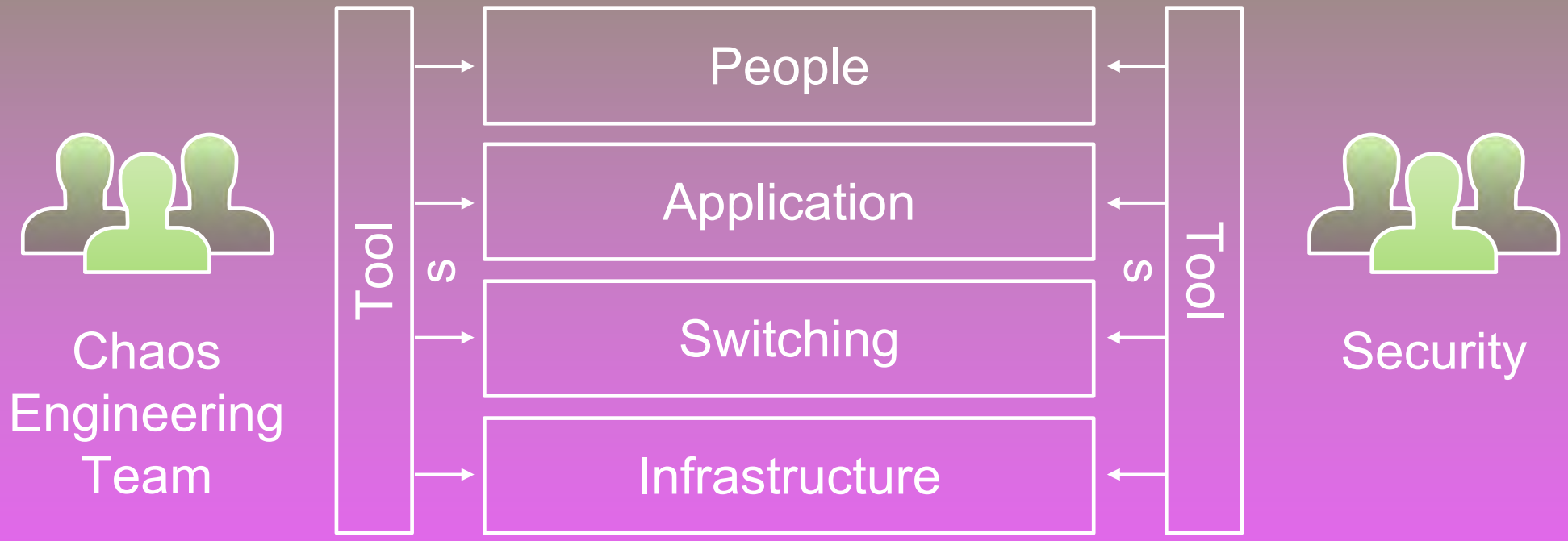
Build Confidence in What Actually Works

So how does it work?

How it Works

How it Works


Develop &
Evaluate
Chaos
Experiment
Plan &
Organize
GameDay
Exercise


Chaos Engineering Team
Tool
s
People
Application
Switching
Infrastructure
s
Tool
Security

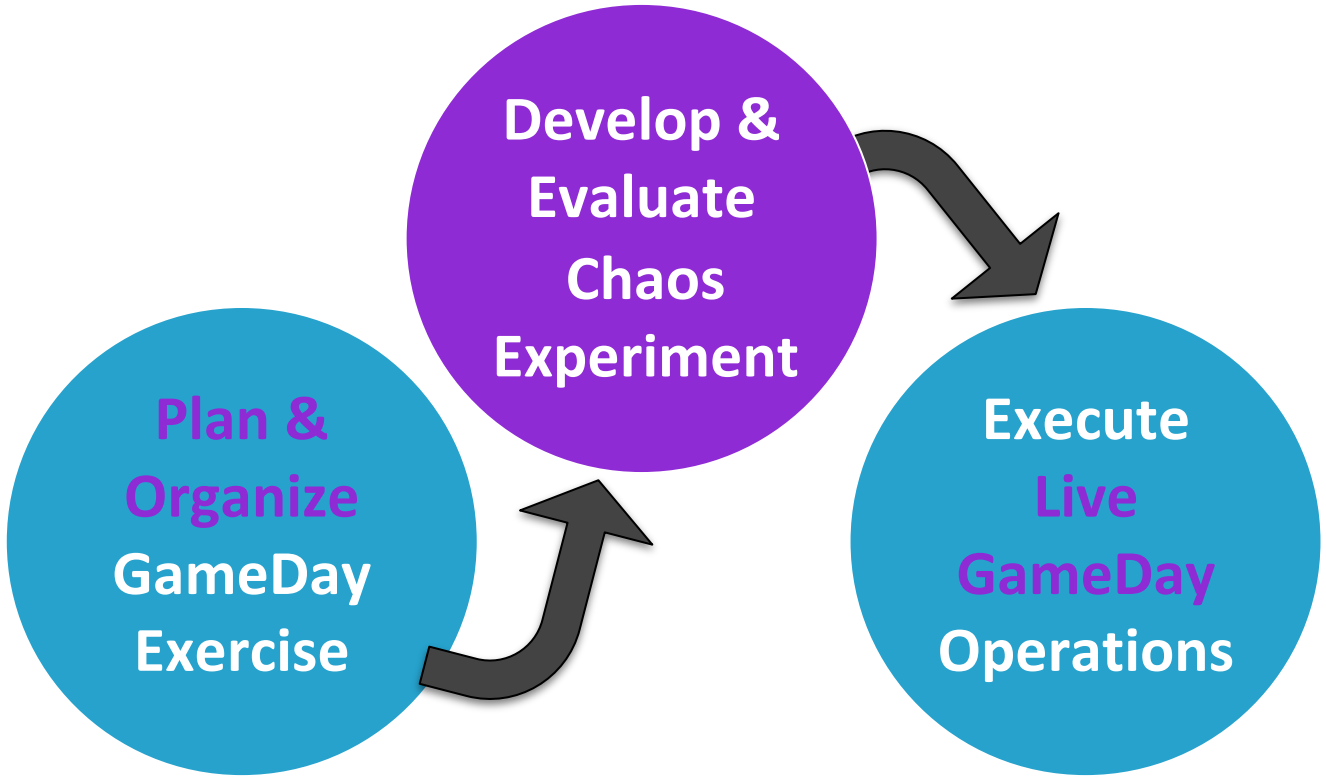
What could go wrong?

What has gone wrong before?

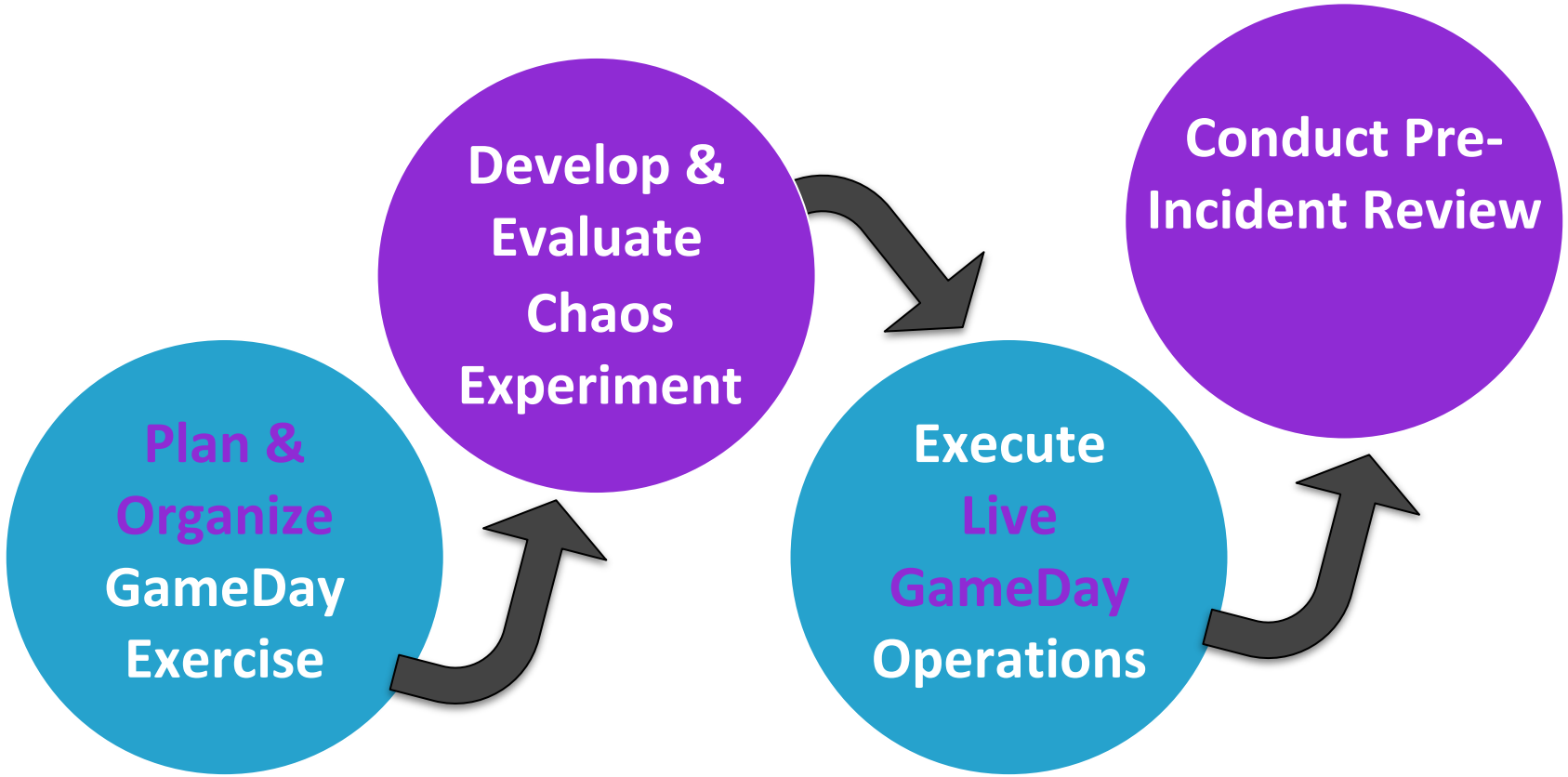
How does My Security Really Work?

What evidence do I have to prove it?

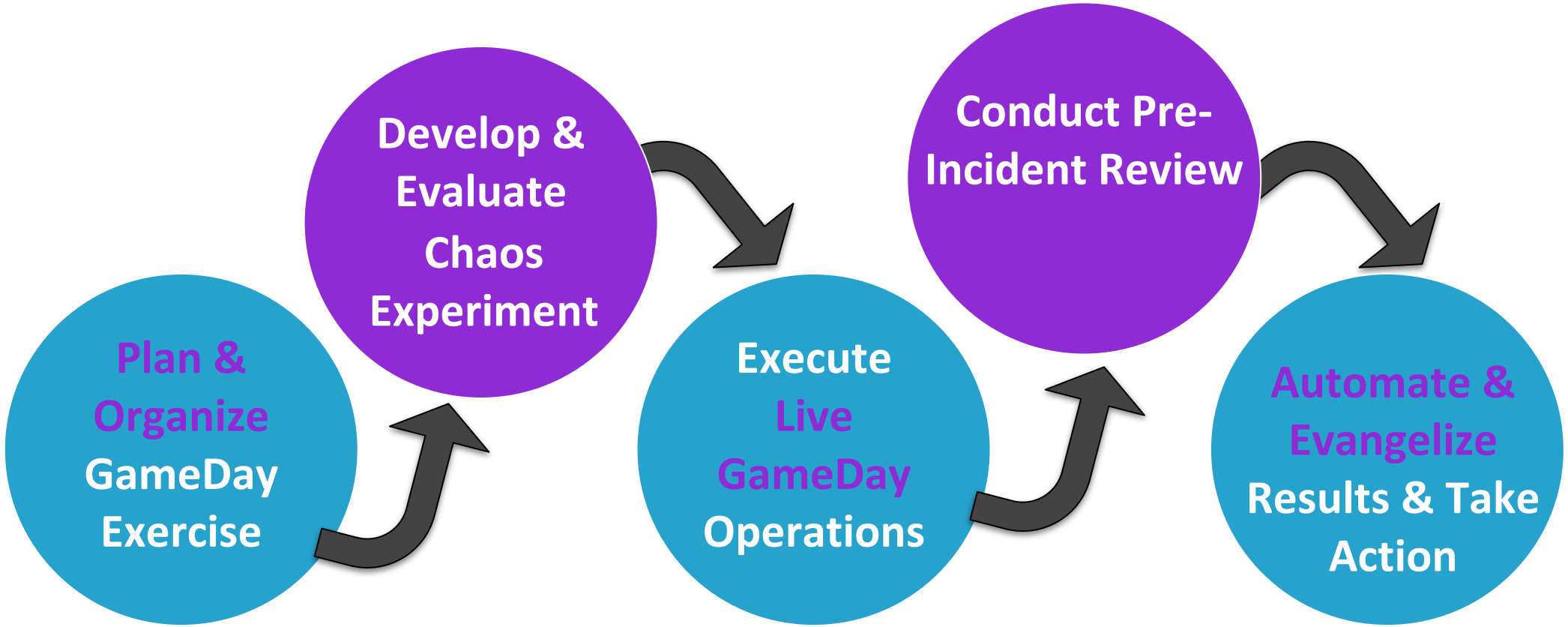
How it Works

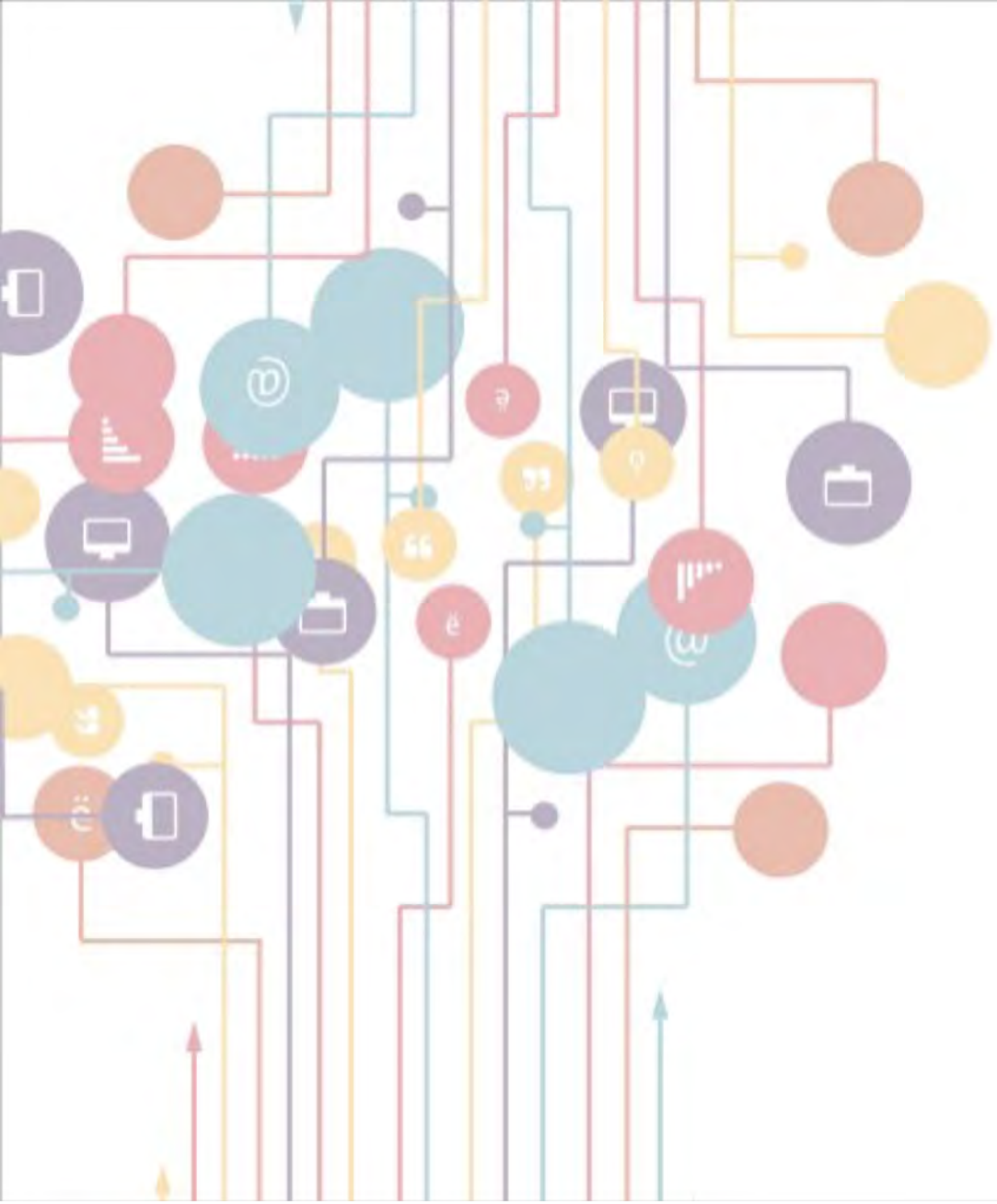

Develop & Evaluate Chaos Experiment
Plan & Organize GameDay Exercise
Execute Live GameDay Operations

How it Works


Plan & Organize GameDay Exercise
Develop & Evaluate Chaos Experiment
Execute Live GameDay Operations
Conduct Pre-Incident Review

How it Works


Plan & Organize GameDay Exercise
Develop & Evaluate Chaos Experiment
Execute Live GameDay Operations
Conduct Pre-Incident Review
Automate & Evangelize Results & Take Action



Example

Hypothesis: If someone accidentally or maliciously introduced a misconfigured port then we would immediately detect, block, and alert on the event.

Result: Hypothesis disproved. Firewall did not detect or block the change on all instances. Standard Port AAA security policy out of sync on the Portal Team instances. Port change did not trigger an alert and log data indicated successful change audit. However we unexpectedly learned the configuration mgmt tool caught change and alerted the SoC.

More Experiment Examples

- *Software Secret Clear Text Disclosure*
- *Permission collision in Shared IAM Role Policy*
- *Disabled Service Event Logging*
- *Introduce Latency on Security Controls*
- *API Gateway Shutdown*

- *Internet exposed Kubernetes API*
- *Unauthorized Bad Container Repo*
- *Unencrypted S3 Bucket*
- *Disable MFA*
- *Bad AWS Automated Block Rule*

ChaoSlingr

An Open Source Tool

ChaoSlingr Product Features

- ChatOps Integration
- Configuration-as-Code
- Example Code & Open Framework

- Serverless App in AWS
- 100% Native AWS
- Configurable Operational Mode & Frequency
- Opt-In | Opt-Out Model




O'REILLY®
Compliments of
NETFLIX
Chaos Engineering
Building Confidence in System Behavior through Experiments

Apply What You Have Learned Today

- Next week you should:
  - Identify critical database(s) within your organization
- In the first three months following this presentation you should:
  - Understand who is accessing the database(s), from where and why
  - Define appropriate controls for the database
- Within six months you should:
  - Select a security system which allows proactive policy to be set according to your organization's needs
  - Drive an implementation project to protect all critical databases

RSAConference2019

Q&A

@aaronrinehart
aaron@verica.io

@ericksonky
ericksonky@gmail.com

RSAConference2019

Thank you RSA!!!!

@aaronrinehart
aaron@verica.io

@ericksonky
ericksonky@gmail.com