Security Precognition: Crafting Secure & Resilient Systems using Chaos Engineering

Name: Aaron Rinehart
Title: CTO, Founder
Organization: Verica
Twitter: aaronrinehart
Email: aaron@verica.io
Aaron Rinehart, CTO, Founder

- Former Chief Security Architect @UnitedHealth
- Former DoD, NASA Safety & Reliability Engineering
- Frequent speaker and author on Chaos Engineering & Security
- O’Reilly Author: Chaos Engineering, Security Chaos Engineering Books
- Pioneer behind Security Chaos Engineering
- Led ChaoSlingr team at UnitedHealth

@aaronrinehart @verica_io #chaosengineering
In this Session we will cover
Incidents, Outages, & Breaches are Costly

Be right back.

We’re making updates to the Apple Store. Check back soon.
Facebook’s image outage reveals how the company’s AI tags your photos

An Obvious Problem
Why do they seem to be happening more often?
Combating Complexity in Software

@aaronrinehart  @verica_io #chaosengineering
“The growth of complexity in society has got ahead of our understanding of how complex systems work and fail”

-Sydney Dekker
Our systems have evolved beyond human ability to mentally model their behavior.
Our systems have evolved beyond human ability to mentally model their behavior.
Security?

Mostly Monolithic
- Prevention focused
- Requires Domain Knowledge
- Poorly Aligned

Defense in Depth
- Expert Systems

Stateful in nature
- Adversary Focused
- DevSecOps not widely adopted
Simplify?
Software has officially taken over
**Software Only Increases in Complexity**

- More Abstract
  - Scripting / interpreted languages
    - Perl, Python, Shell, Java
  - High / middle level languages
    - C, C++
  - Assembly language
    - Intel X86, etc (first layer of human-readable code)
  - Binary code
    - Binary code read by hardware - not human-readable
Software Complexity

Accidental  Essential
Woods Theorem:

“As the complexity of a system increases, the accuracy of any single agent’s own model of that system decreases”

- Dr. David Woods
What does this have to do with my systems?
Question - How well do you really understand how your system works?
In Reality......

Systems Engineering is Messy
In the beginning...we think it looks like
After a few months...

- Hard Coded Passwords
- New Security Tool
- Identity Conflicts
- Lead Software Engineering finds a new job at Google
- 300 Microservices $\rightarrow$ 850 Microservices
- Network is Unreliable
- Autoscaling Keeps Breaking
- Refactor Pricing
- Cloud Provider API Outage
- DNS Resolution Errors
- Code Freeze
- Regulatory Audit
- Expired Certificate
- Scalability Issues
- WAF Outage $\rightarrow$ Disabled
- Large Customer Outage
- DNS Resolution Errors
- Expired Certificate
- Regulatory Audit
- Rolling Sev1 Outage on Portal
- Code Freeze
- Delayed Features
Our systems become more complex and messy than we remember them.
Difficult to Mentally Model
So what does all of this $&%* have to do with Security?
Failure Happens A lot
The Normal Condition is to FAIL
We need failure to Learn & Grow
“things that have never happened before happen all the time”

-Scott Sagan “The Limits of Safety”
How do we typically discover when our security measures fail?
Security Incidents

Typically we don't find out our security is failing until there is a security incident.
Vanishing Traces

Logs, Stack Traces, Alerts

All we typically ever see is the Footsteps in the Sand
- Allspaw
Security incidents are not effective measures of detection because at that point it's already too late.
No System is inherently Secure by Default, its Humans that make them that way.
People Operate Differently when they expect things to fail
Awesome!
Chaos Engineering

@aaronrinehart  @verica_io #chaosengineering
Chaos Engineering

“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”
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.
Use Chaos to Establish Order
Testing vs. Experimentation

THIS IS A TEST.
This station is conducting a test of the Emergency Broadcast System.
THIS IS ONLY A TEST.
Chaos Monkey Story

During Business Hours
Born out of Netflix Cloud Transformation
Put well defined problems in front of engineers.
Terminate VMs on Random VPC Instances
Chaos Pitfalls: Breaking things on Purpose

The purpose of Chaos Engineering is **NOT** to “Break Things on Purpose”. If anything we are trying to “Fix them on Purpose”!

“*I’m pretty sure I won’t have a job very long if I break things on purpose all day.*”

-Casey Rosenthal

Reference: Nora Jones 8 Traps of Chaos Engineering
Security Chaos Engineering

@aaronrinehart  @verica_io #chaosengineering
Continuous Security Verification
Proactively Manage & Measure
Reduce Uncertainty by Building Confidence in how the system actually functions
Security Chaos Engineering

Use Cases

@aaronrinhart  @verica_io #chaosengineering
Use Cases

- Incident Response
- Solutions Architecture
- Security Control Validation
- Security Observability
- Continuous Verification
- Compliance Monitoring
Incident Response
Security Incidents are Subjective in Nature
We really don't know very much

“Response” is the problem with Incident Response
Lets face it, when outages happen.....

Teams spend too much time reacting to outages instead of building more resilient systems.
Let's Flip the Model

Post Mortem = Preparation
OMG!

What are your robot serial numbers?
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
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.
Stop looking for better answers and start asking better questions.

- John Allspaw
Free copy mailed to you
complements of Verica

cutt.ly/verica-book
THANK YOU!

Meet me in the Network Chat Lounge for questions
Agenda

- Combating Complexity in Software
- Chaos Engineering
- Resilience Engineering & Security
- Security Chaos Engineering