



CONTROLLED CHAOS

The Inevitable Marriage of DevOps & Security

Kelly Shortridge (@swagitda_)

Velocity Berlin 2019



Hi, I'm Kelly

CAPSULE 8



“Chaos isn’t a pit. Chaos is a ladder.”
— Petyr Baelish, *Game of Thrones*

Infosec has a choice: marry DevOps
or be rendered impotent & irrelevant





Infosec won't survive in a silo. It must be embedded in software delivery.

A surreal landscape with a dark, starry sky. The foreground is dominated by large, curved, layered rock formations in shades of purple and blue. In the center, a small, dark rectangular doorway or opening is visible, with a small, silhouetted figure standing within it. The lighting is dramatic, with a bright light source behind the doorway, creating a strong contrast and casting long shadows. The overall atmosphere is mysterious and futuristic.

DevOps can learn to carve its own
path to secure software delivery

A hand holding a glowing purple heart-shaped lightbulb against a black background with a large purple heart outline.

How can controlling chaos create a marriage of infosec and DevOps?

1. Chaos Theory
2. Time to D.I.E.
3. A Phoenix Rises



Chaos Theory

Chaos engineering = continual
experimentation to test resilience

“Things will fail” naturally extends
into “things will be pwned”



HURT ME

Security failure is when security controls don't operate as intended

A dramatic night scene of a volcanic eruption. A volcano is shown with a bright red and orange lava flow cascading down its right side. A powerful lightning bolt strikes the peak of the volcano, illuminating the dark, billowing smoke and ash that rises into the starry night sky.

What are the principles of chaotic security engineering?

1. Expect that security controls will fail & prepare accordingly

2. Don't try to avoid incidents – hone your ability to respond to them

Game days: like planned fire drills



Prioritize security game days based on potential business impacts

A futuristic cityscape at night, viewed through the dark silhouette of a large tree in the foreground. The city is illuminated with various colors, including purple, blue, and yellow, suggesting a sunset or sunrise. The buildings are tall and modern, with many windows lit up. The overall atmosphere is mysterious and high-tech.

Decision trees: start at target asset,
work back to easiest attacker paths

Determine the attacker's least-cost path (hint: it doesn't involve 0day)



Your goal is to raise the cost of
attack, ideally beginning at design



Time to D.I.E.

We need a model promoting qualities
that make systems more secure

Enter the **D.I.E. model** by Sounil Yu:
Distributed, Immutable, Ephemeral



Distributed: multiple systems
supporting the same overarching goal

Distributed infrastructure reduces risk of DoS attacks by design

A service mesh is like an on-demand VPN at the application level

A 3D rendered scene of a dark blue hallway. In the center, there is a doorway that is brightly lit from within, casting a strong red glow. Inside the doorway, a set of stairs is visible, also illuminated by the red light. The walls and floor of the hallway are dark blue, creating a high-contrast environment. The text "Attackers are forced to escalate privileges to access the iptables layer" is overlaid in white, sans-serif font across the middle of the image.

Attackers are forced to escalate
privileges to access the iptables layer



Immutable: infrastructure that
doesn't change after it's deployed



Immutable infra is more secure by design – ban shell access entirely

Patching is no longer a nightmare
with version-controlled images



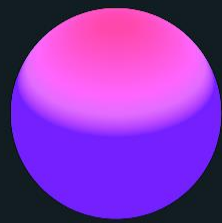
Ephemeral: infrastructure with a very short lifespan (dies after a task)

Ephemerality creates uncertainty for attackers (persistence = nightmare)

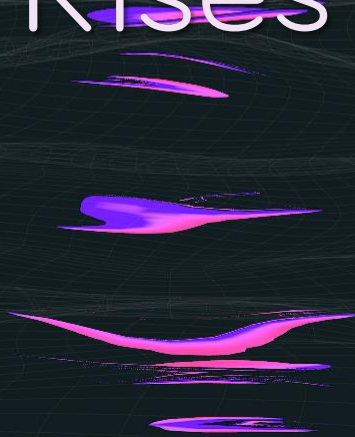


Installing a rootkit on a resource that dies in minutes is a waste of effort

Optimizing for D.I.E. reduces risk by
design & supports resilience



A Phoenix Rises



Begin with “dumb” testing before moving to “fancy” testing

D.I.E.ing is an art, like everything else



The background features a complex network graph with numerous nodes and edges. The nodes are represented by small, glowing purple spheres, and the edges are thin, light purple lines connecting these nodes. The overall aesthetic is futuristic and technical, set against a dark blue gradient background.

Controlling Chaos: Distributed



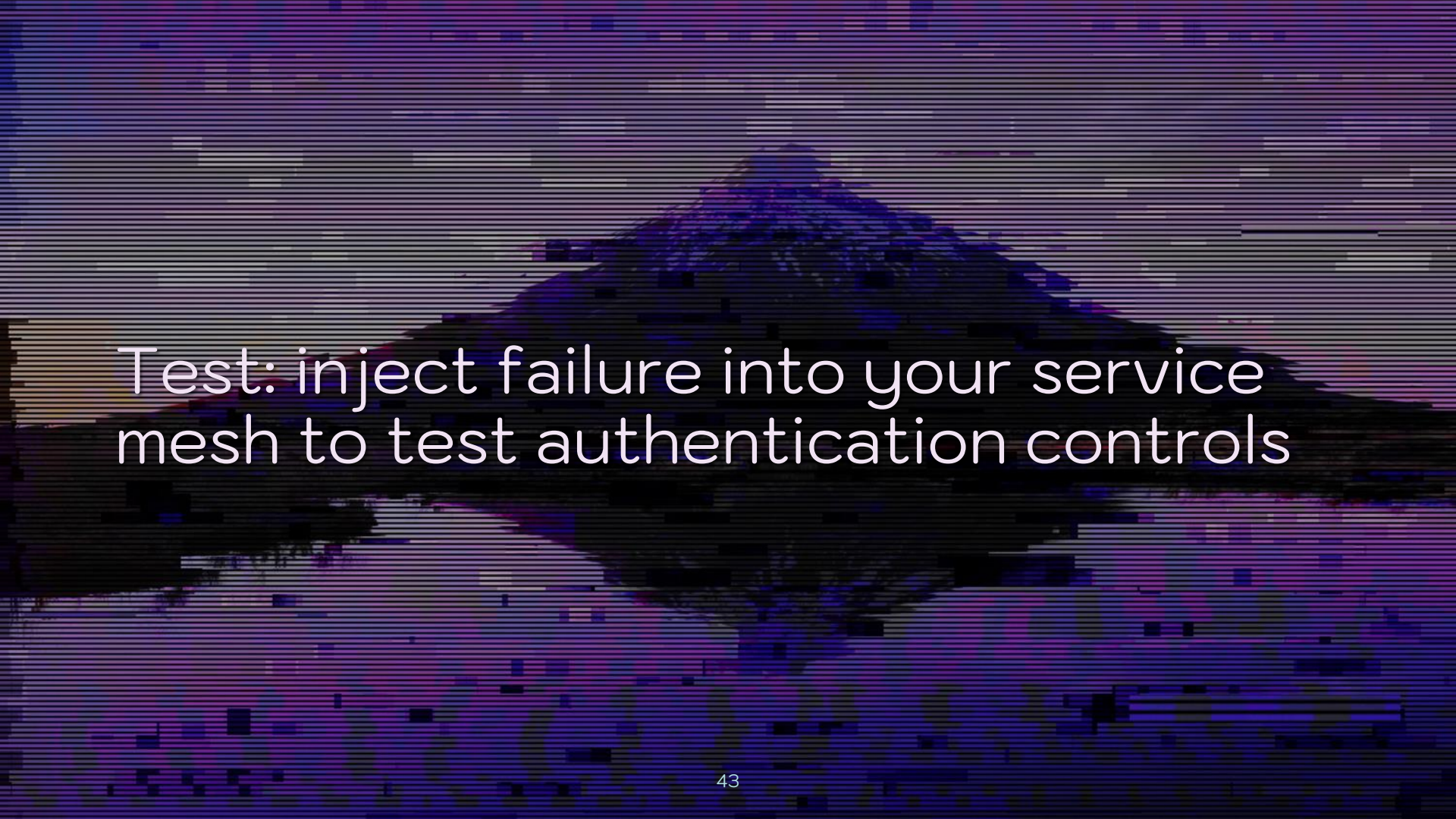
Distributed is mostly covered by the
existing repertoire of chaos eng tools

Repurpose these tools, but make
attackers the source of failure

A futuristic, glowing tunnel with a person standing in the center. The tunnel is illuminated with vibrant purple and blue light, creating a sense of depth and mystery. The person is standing on a circular platform in the middle of the tunnel, looking towards the viewer. The overall atmosphere is high-tech and cybernetic.

Multi-region services present a fun
opportunity to mess with attackers

Shuffle IP blocks regularly to change
attackers' lateral movement game



Test: inject failure into your service mesh to test authentication controls

A glowing pink square frame is centered in a dark, snowy landscape at night. The frame is illuminated from within, casting a soft pink glow. The background shows silhouettes of trees and a snow-covered ground. The text "Controlling Chaos: Immutable" is overlaid in white, sans-serif font across the center of the image.

Controlling Chaos: Immutable

Immutable infra is like a phoenix – it disappears & comes back a lot



Volatile environments with continually moving parts raise the cost of attack

Create rules like, “If there’s ever a write to disk, crash the node”



Attackers must stay in-memory,
which hopefully makes them cry

Bonus: disallowing all local IO
improves service reliability

Metasploit Meterpreter + webshell:
Touch passwords.txt & kaboom



Build your Docker images with a
garbage-filled “bamboozle layer”

Mark garbage files as “unreadable” to craft enticing bait for attackers

A futuristic, dark environment with glowing blue and purple lines and a central glowing rectangular frame. The scene is filled with complex, layered structures and glowing elements, creating a sense of depth and mystery. The text is overlaid in the center in a white, sans-serif font.

A potential goal: architect
immutability turtles all the way down

Test: inject attempts at writing to disk to ensure detection & reversion



Treat changes to disk by adversaries
similarly to failing disks: mercy kill



Controlling Chaos: Ephemeral

Most infosec bugs are stated-related
– get rid of state, get rid of bugs



Reverse uptime: longer host uptime
adds greater security risk

Test: change API tokens & test if services still accept old tokens

A statue of a man in a purple tunic and sandals, standing on a grid floor against a purple background with mountains. The statue is the central focus, with a black bar obscuring its face. The background features a grid floor and purple mountains under a purple sky.

Test: retrograde libraries, containers,
other resources in CI/CD pipelines

Test: inject hashes of old pieces of data to ensure no data persistence

Leverage lessons from toll fraud –
cloud billing becomes security signal



Test: exfil TBs or run a cryptominer
to inform billing spike detection



Conclusion



Chaos/resilience are natural homes
for infosec & represent its future.



The future of infosec involves unified responsibility & accountability.

Security can be innovative and fuel
the engine of business as well.





“You must have chaos within you to
give birth to a dancing star.”

— Friedrich Nietzsche



@swagitda_



/in/kellyshortridge



kelly@greywire.net