

FROM CATASTROPHE TO CHAOS IN PRODUCTION

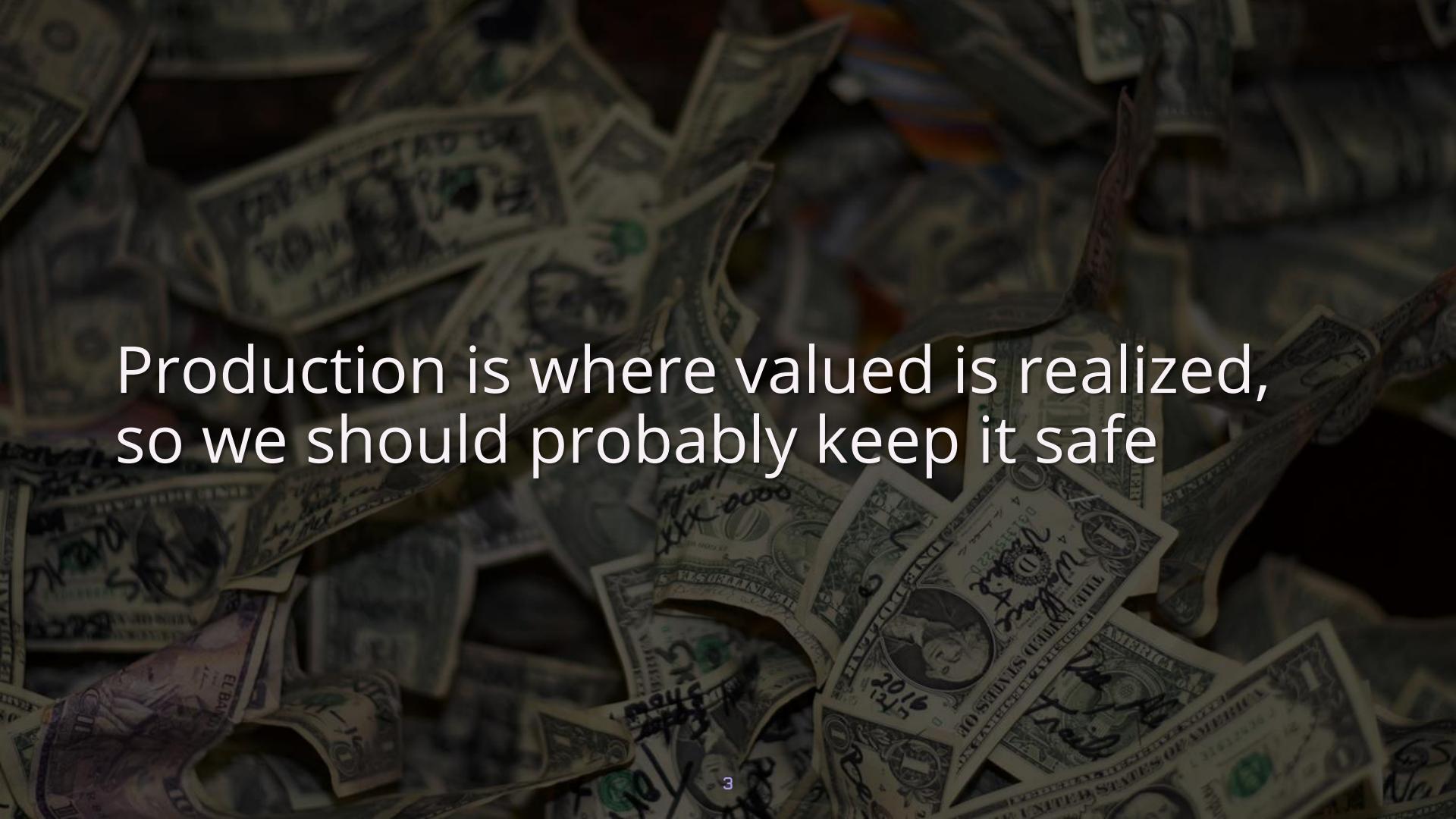
Kelly Shortridge (@swagitda_)

GOTOpia Chaos Engineering Day 2021



Hi, I'm Kelly

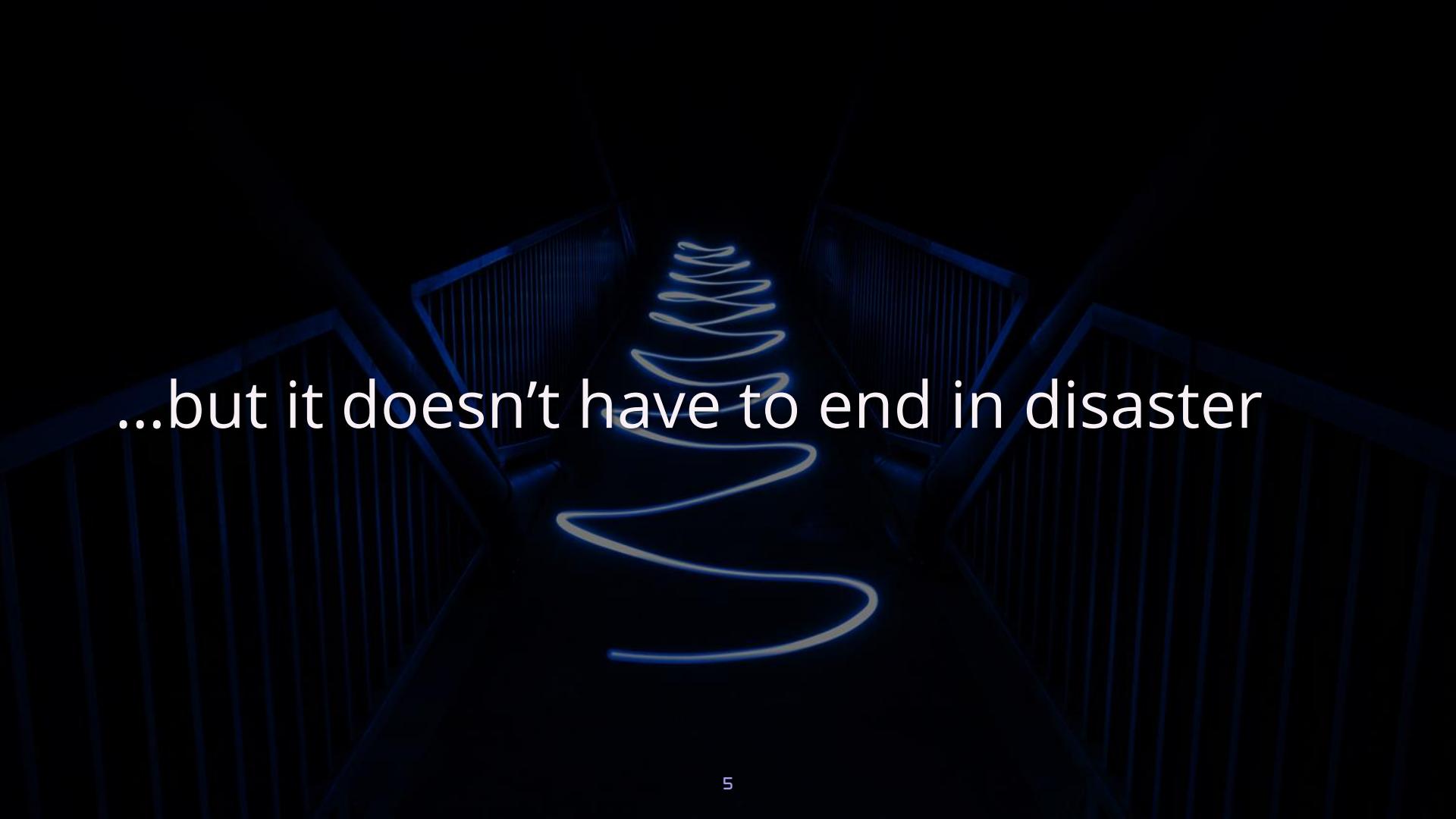
CAPSULE8

The background of the slide is a dark, out-of-focus pile of US dollar bills, creating a sense of wealth or capital.

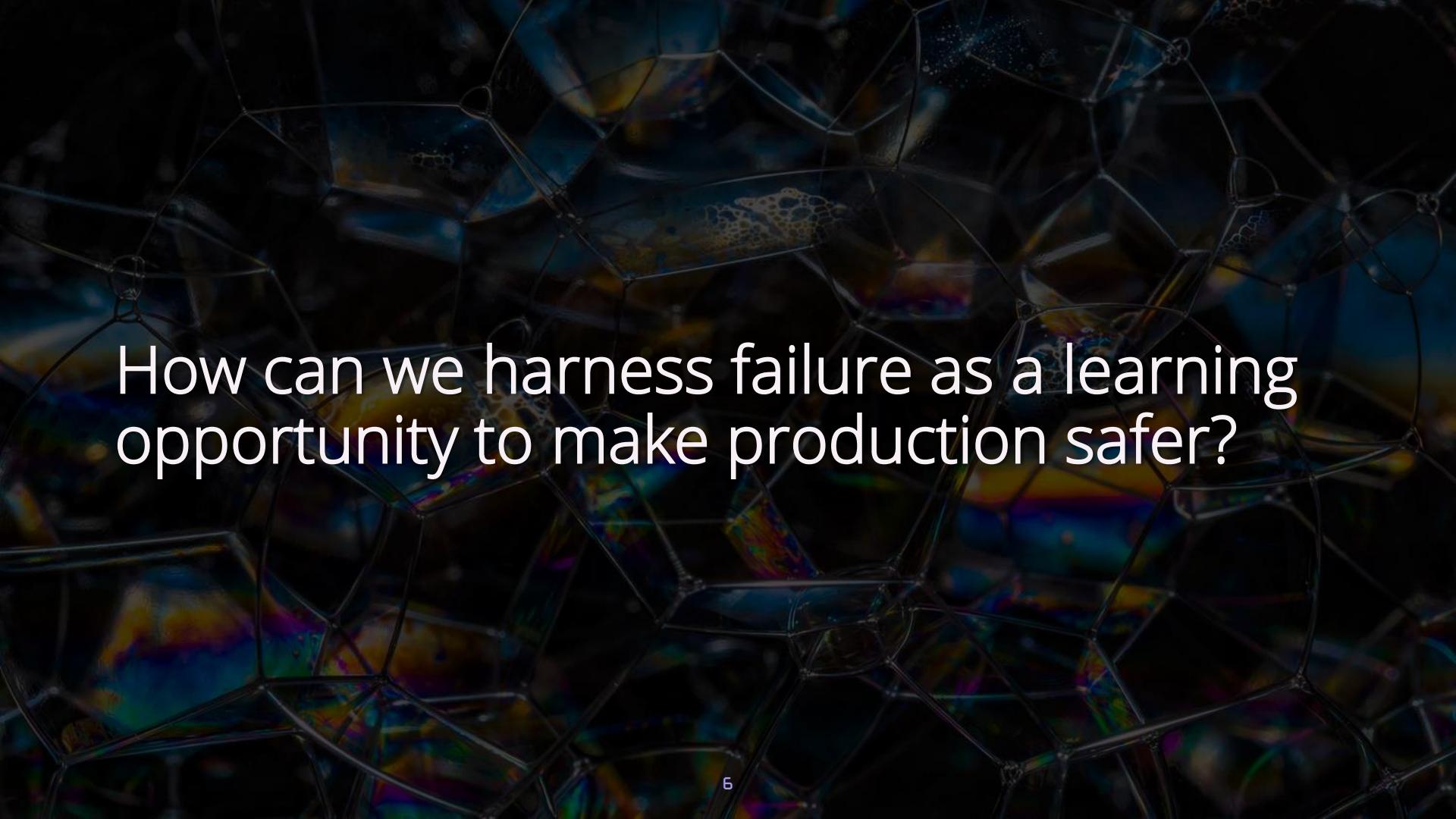
Production is where value is realized,
so we should probably keep it safe



Failure in production feels frightening

A glowing blue spiral light effect, resembling a Christmas tree or a DNA helix, is centered on a dark background. The spiral is composed of several concentric, slightly blurred loops of light, creating a sense of motion and depth.

...but it doesn't have to end in disaster



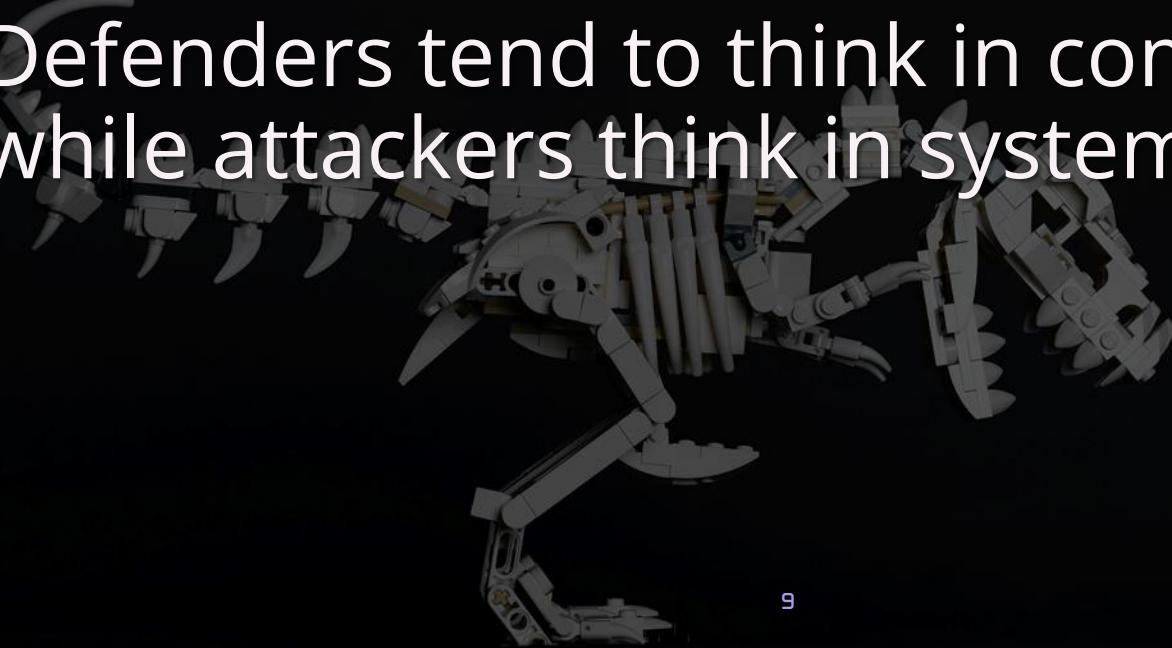
How can we harness failure as a learning opportunity to make production safer?

I. Failure in Production

II. Security Chaos
Engineering in Production

I. Failure in Production

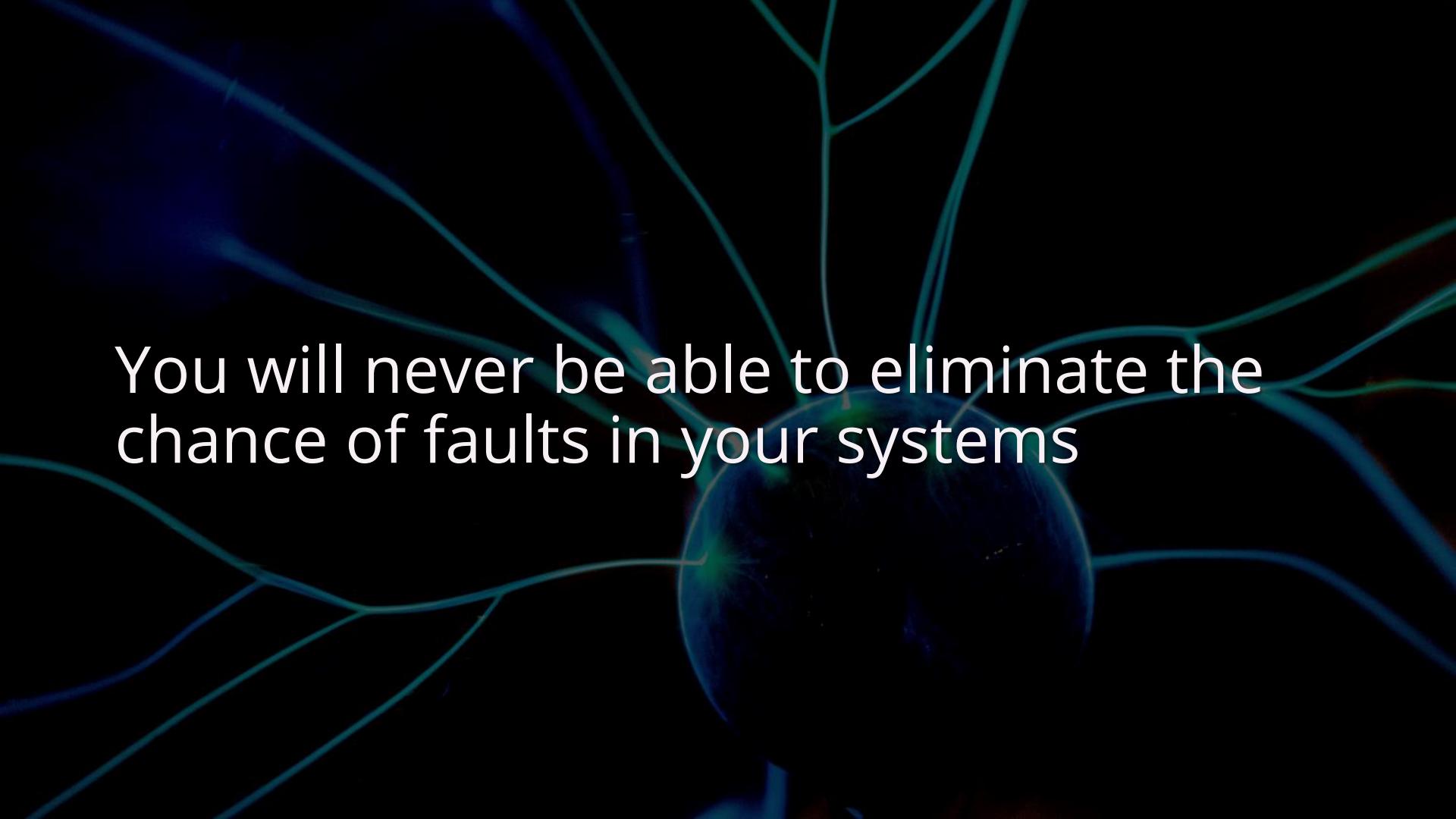
Defenders tend to think in components
while attackers think in systems



Component-level vs. system-level –
faults are different than failures

Faults: “one component of the system deviating from its spec”

Failure: “the system as a whole stops providing the required service to users”

The background of the slide features a dark, abstract design. It consists of several glowing blue lines of varying thicknesses that curve and intersect across the frame. Interspersed among these lines are several translucent, glowing blue spheres of different sizes, some appearing to emit light rays. The overall effect is futuristic and suggests a complex network or a microscopic view of particles.

You will never be able to eliminate the
chance of faults in your systems

Prevention only goes so far; too many variables are out of your control

A perfectly patched container can still be pwned if there's anon access in K8s

Scan all the code for vulns... then
attackers compromise the code scanner

Yubikeys for GitHub... then attackers
abuse Jenkin's anon script console



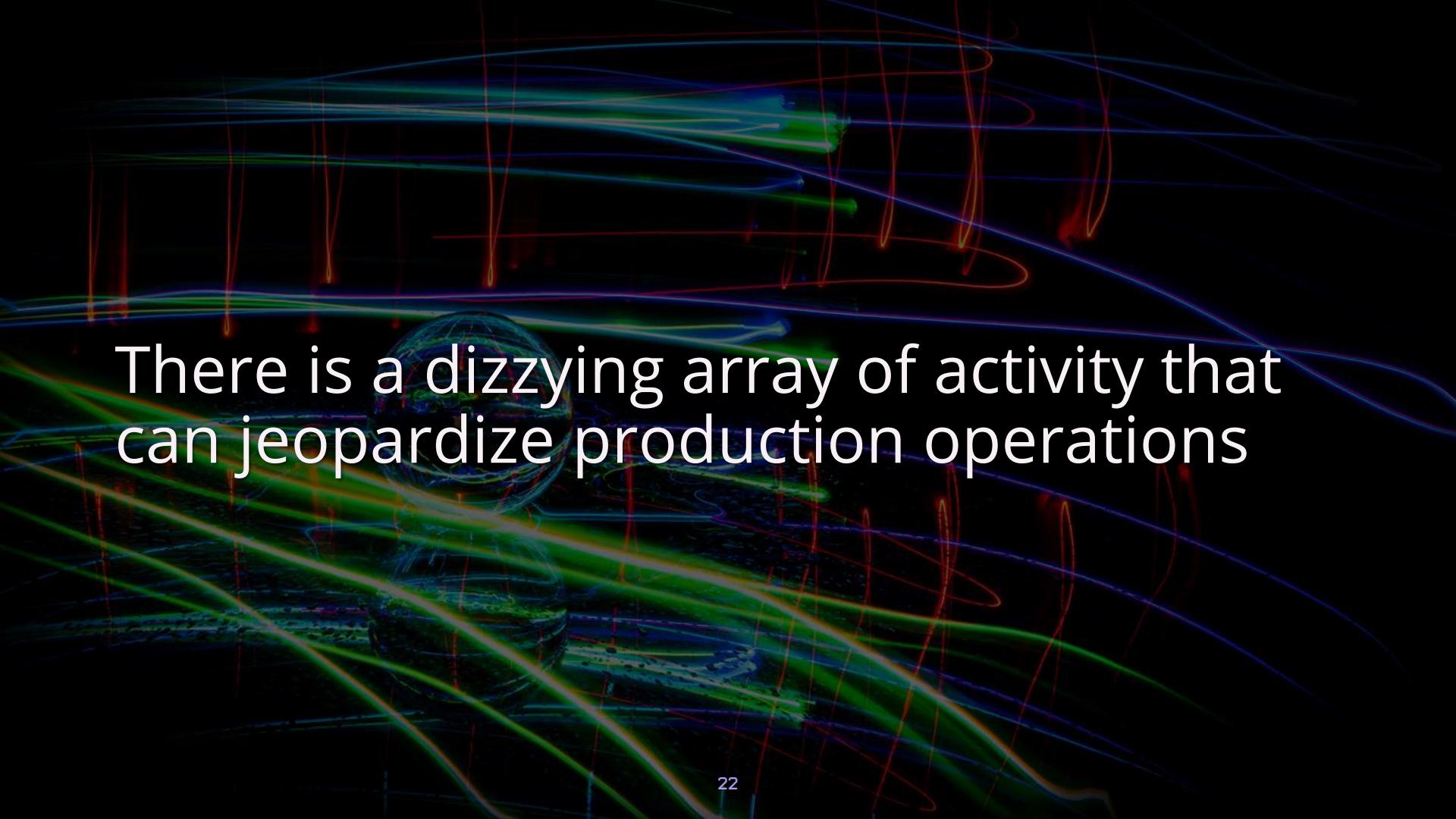
Failure in production manifests in a
mess of multiplicitous manners

Private / public clouds, VPS, VMs,
containers, serverless, computerless...

The background of the slide features a dark, abstract design composed of glowing, translucent spheres and flowing, wavy lines in various colors like green, yellow, red, blue, and purple. These light trails create a sense of motion and complexity, symbolizing the interconnected components of a production environment.

Production environments are complex systems full of interrelated components

Failure is like a tapestry of interwoven strands that can spread fire to the rest



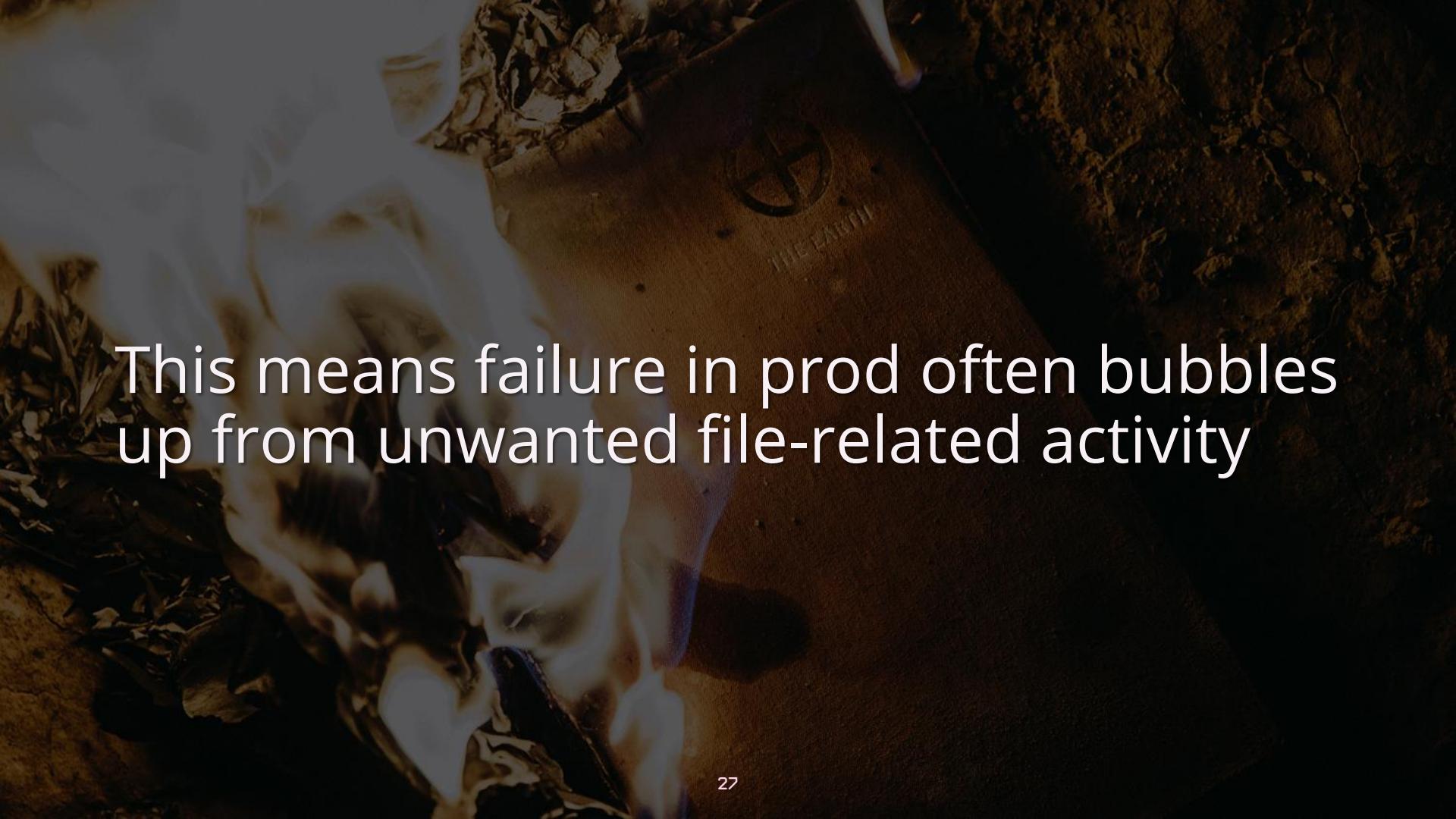
There is a dizzying array of activity that can jeopardize production operations

Two key types: deliberately malicious
(attackers) + accidentally careless (devs)

Sometimes they overlap! Like attaching
a debugger to a prod system

Attackers with privileged creds &
“insider threats” are basically the same

Most prod infrastructure runs on Linux,
where everything is a file



This means failure in prod often bubbles up from unwanted file-related activity

Example 1: Log files are deleted or tampered - your ops is likely screwed

Example 2: Changes to boot files, root cert stores, or SSH keys - stability snafus

Example 3: Resource limits are disabled
- highly sus and doubtless disastrous

Confronted with such complexity, how can we constructively cope?

II. Security Chaos Engineering in Prod

Our goal is to prevent faults from causing failures as much as we can

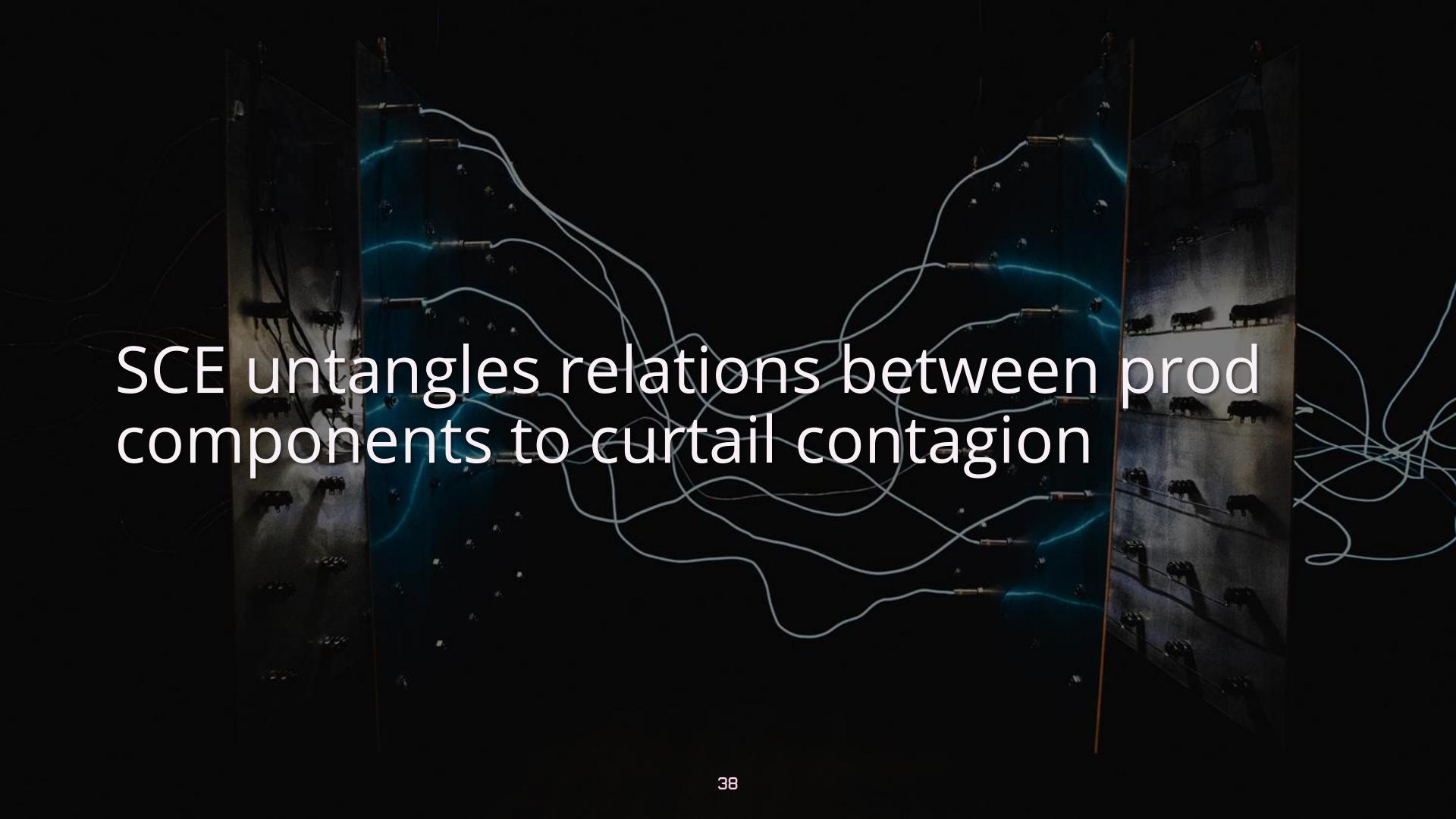


Purposefully triggering faults lets you
realize and test your success towards it

Security Chaos Engineering: Let's harness failure to build knowledge

Conducting experiments generates evidence & builds muscle memory

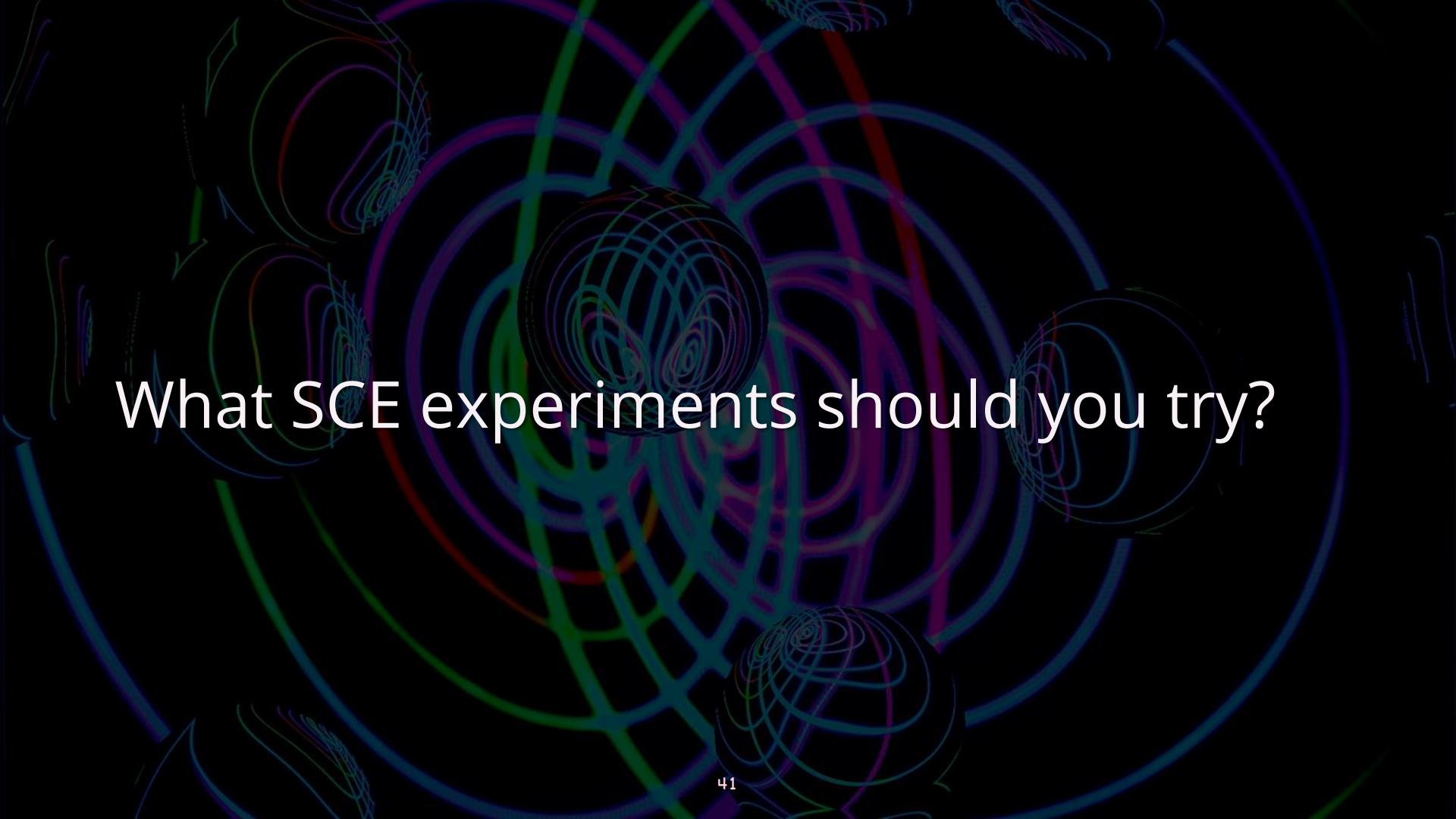
Make incident response boring because
it feels routine after repeated practice



SCE untangles relations between prod
components to curtail contagion

Learning how your systems respond to failure requires testing in prod itself

...but you can start in staging to build confidence before migrating to prod

The background of the slide features a complex, abstract pattern of colored lines on a black surface. The lines are primarily in shades of blue, green, and purple, creating a sense of depth and motion. They form various shapes, including spirals and loops, that overlap and intersect across the frame.

What SCE experiments should you try?

Example prioritization matrix of attack goals

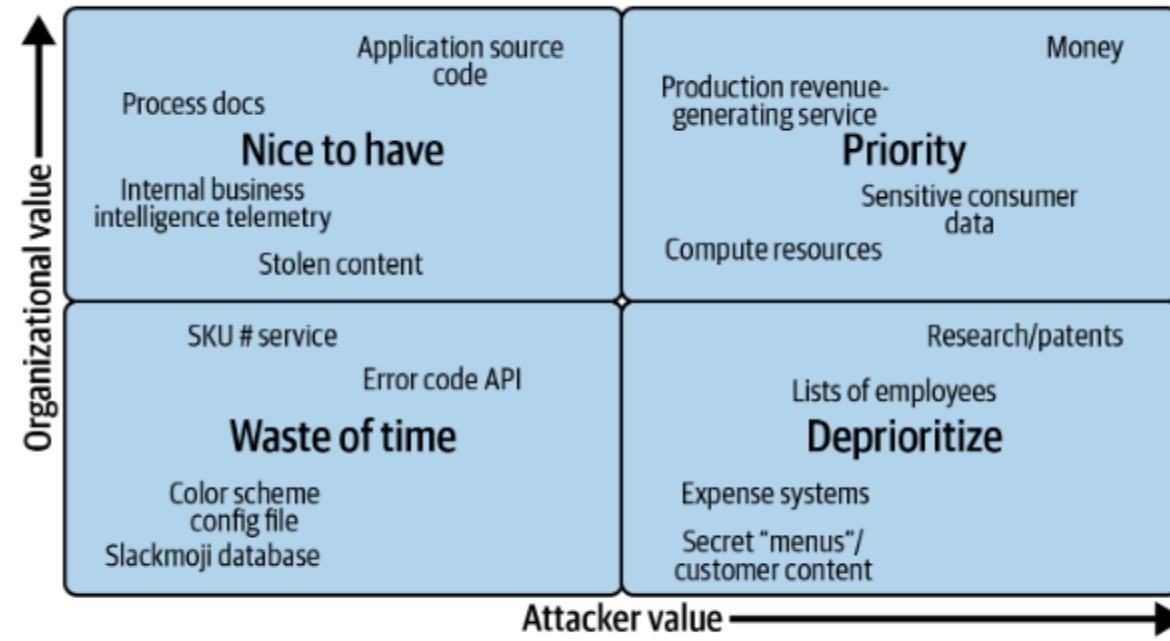


Figure 2-5. Example prioritization matrix of assets relative to attacker value and organizational value.

Let's explore some examples...



Example 1: Create & execute a new file
in a container

How does your container respond to
new file exec? Does it affect the cluster?



Example 2: Inject program crashes

Does your node restart itself? How quickly can you redeploy post-crash?

Example 3: Disable resource limits (CPU, file descriptors, memory, restarts, etc.)

Can an infinite script take up resources?
Do slower response times propagate?

Example 4: Disable access to DNS

How reliant are your systems on
external DNS? Do you have a fallback?



Example 5: Time travel on a host

How do systems handle expired certs?
Do time-related issues bork services?

The background of the image is a dark, textured surface, possibly black or dark blue, covered with numerous small, glowing orange and yellow specks that resemble stars or dust. Overlaid on this background is a thick, swirling stream of bright orange paint. The paint starts from the bottom left, curves upwards and to the right, then loops back down towards the bottom center. It has a glossy, liquid appearance with some darker, more viscous areas where it has settled.

In Conclusion

- Failure in production is inevitable, so you must learn from it early and often

The background of the slide features a complex, abstract pattern of thin, light-colored lines against a dark, almost black, background. These lines are densely packed and intersect at various angles, creating a sense of depth and complexity. Some lines are straight, while others are slightly curved or wavy. The overall effect is reminiscent of a microscopic view of a neural network or a complex web.

Conducting experiments uncovers new
knowledge & builds muscle memory



Security chaos engineering builds
confidence in the safety of prod systems

“Our real discoveries come from chaos,
from going to the place that looks wrong
and stupid and foolish.”

- Chuck Palahniuk

Download for free:
<https://www.verica.io/sce-book/>

Security Chaos Engineering

Gaining Confidence in Resilience
and Safety at Speed and Scale

Aaron Rinehart & Kelly Shortridge

REPORT



@swagitda_



/in/kellyshortridge



kelly@shortridge.io