

# Sysadministrivia

## Linux, Lagers, and Late Nights

---

### S5E9: "Chaotic Good"

Posted 2020-06-21 23:59

Modified 2020-06-21 03:16

Comments 0

#### Navigation

| Previous Episode           | Next Episode                                  |
|----------------------------|---|
| S5E8: "Feeling Vulnerable" | S5E10: "Beer in the Back, Party in the Front" |

#### Log

| Recorded (UTC)      | Aired (UTC)         | Editor  |
|---------------------|---------------------|---------|
| 2020-06-11 21:18:34 | 2020-06-21 05:53:53 | "Edita" |

#### Verification

| Format | SHA256   | GPG   | Audio File |
|--------|--|-------|------------|
| MP3    | aa79ff381229a7671774418c5d4b62325d9ae0186cd907ffb3be6ebc96186f61 | click | click      |
| OGG    | 9f31119d43ef4bbdeac97f66243fc8817b59cea9d3260761ef1d5a8be23bd0da | click | click      |

Quicklisten:

We have a guest on to talk about chaos engineering. Fighting "Schrödinger's failover"!

- Just the Tip
- Notes
- 15 Clams
- Errata
- Music

## Just the Tip

- Paden talks about the woes, trials, and tribulations of libboost.
  - Jthan and I reminisce our woes about compiling boost on Gentoo.

## Notes

Starts at **08m24s**.

I was drinking Moosehead Lager. Paden was drinking water and Diet Dr. Pepper. Jthan was drinking water. Miko was drinking a Merlot.

- We have Mikołaj Pawlikowski from Bloomberg on to talk about chaos engineering!
  - Miko is releasing a book on Chaos Engineering. It's titled Chaos Engineering: Crash Test Your Applications and it's being published by Manning Publications. (They seem pretty similar to NoStarch in that they're very friendly to ebook formats; give them a look!)
  - Chaos engineering can be summed up as *purposefully breaking infrastructure to test resilience in the context in an unexpected breakage*. It **very** strongly follows the scientific method.
    - It's a perpetual thing, not a one-off thing like benchmarking or stresstesting. Taking InfoSec as an analogy, it's not a pentest or audit, it's a standards and best-practices compliance (only instead of a certification, your standard is the SLA).
  - Chaos engineering is good for finding problems you haven't accounted for yet (by simulating real-world hazards that would typically be faced by infrastructure).
  - "Chaos", by its nature, can't really be engineered per se. But a lot of it is based on testing hypotheses to gain knowledge of your infrastructure's behaviour based on experimentation. (See? Basically the scientific method!)
  - Paden asks if the operations team(s) know when the "chaos" is being introduced. Miko says it's usually unannounced (because then you'd lose the unexpected quality of chaos, which is the entire point of chaos engineering arguably).
  - He also hopes chaos engineering will be more widely deployed and taken more seriously. (I'm with him; it's a useful tool and more or less the only reliable method for testing and verifying SLA.)
    - It's also a very effective way of combatting the paradox of the "Invisible Operations". (See S4E1 for an extended discussion on this paradox!)
- A good place to start is [principlesofchaos.org](http://principlesofchaos.org).
  - Miko recommends two books by Casey Rosenthal, appropriately titled Chaos Engineering (2017) and Chaos Engineering (2020) from O'Reilly (yes, they're different books!) for a technical/theory approach. His book focuses more on a practical approach (since so much of chaos engineering is based on designing experiments anyways).
  - If you're interested in furthering your chaos engineering knowledge, Miko has a newsletter about it that we highly recommend you subscribe to!
  - Miko also really likes (and uses) [powerfulseal...](#)
  - and highly recommends the resource dump [Awesome Chaos Engineering](#).

# 15 Clams

In this segment, Jthan shares with you a little slice of life. The title is a reference to this video. (2m16s in)

Starts at **44m33s**.

Jthan ponders why more distros don't do slotted installs. I point out that a distro kind of has to be designed ground-up to support it to do it well, and pretty much just Gentoo (and its derivatives) is designed for that.

Some ideas we come up with (aside from maintaining separate branches in your SCM e.g. git):

- Virtual machines
- Containerizing
- Chroots
  - And namespacing/cgroups in systemd's machined (or just the vanilla kernel)
- Flatpaks (and Snaps)...
  - But this might be encouraging poor development practices.
  
- Jthan's also still mad about the AddTrust CA SNAFU.

## Errata

- None so far! But my favourite is when Jthan asks Miko if he thinks containers are secure and he just laughs in response.

## Music

### Music Credits

| Track | Title                         | Artist        | Link                  | Copyright/License |
|-------|-------------------------------|---------------|-----------------------|-------------------|
| Intro | Do Not Pretend To Be The Same | Floating Mind | <a href="#">click</a> | CC-BY-ND 4.0      |
| Outro | Crawl                         | Dub Bred      | <a href="#">click</a> | CC-BY-NC-ND 4.0   |

(All music is royalty-free, properly licensed for use, used under fair use, or public domain.)

**Author** r00t^2

**Categories** Season Five

## Comments

There are currently no comments on this article.