

Chaos Engineering

Lukas Szerszen & Emil Gedda

Agenda

- Introduction
- Theoretical concepts
- Methodology
- Gremlin with example
- Take-home message

Vaccine for software

We inject harm to build immunity^[1]

Chaos Engineering

Chaos Engineering is the discipline of experimenting on a system in order to build confidence in the system's capability to withstand turbulent conditions in production. [2]

Chaos Engineering^{[1][2]}

- Inject faults into the system to assess resilience
- Controlled and planned
- Done in production environment
- Build confidence in error handling and fault tolerance
- Proactive

Chaos Engineering Core Concepts^[3]

- Metrics
- Perturbation
- Hypothesis
- Experiment
- Blast Radius

Methodology^[3]

- Define a steady state; should be expressed in terms of metric value ranges
- Perturbations; what faults to inject
- Control Phase vs Experimental Phase
- Hypothesize about the outcome
- Carry out chaos experiment
- Validate or Falsify hypothesis

Why Chaos Engineering?^[5]

- Shift to the cloud and microservices
 - +73% of companies already have an application in the cloud^[4]
- Some failure scenarios out of direct control
- Too complex to predict all failure scenarios
- Avoid costly outages by being prepared

Gremlin

- Hosted Chaos Engineering Framework
 - “Failure as a Service”
- Web based controller UI
- Deploy through Docker or Kubernetes
- Application and/or Infrastructure targeting



Matthew
Fornaciari



Kolton
Andrus



Practical example

(Setup)

```
$ export GREMLIN_TEAM_ID="..."
$ export GREMLIN_TEAM_SECRET="..."
$ sudo docker run -d \
    --net=host --pid=host \
    --cap-add=NET_ADMIN --cap-add=SYS_BOOT \
    --cap-add=SYS_TIME --cap-add=KILL \
    -e GREMLIN_TEAM_ID \
    -e GREMLIN_TEAM_SECRET \
    -v /var/run/docker.sock \
    -v /var/log/gremlin \
    -v /var/lib/gremlin \
    gremlin/gremlin daemon
```

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND
c48db9cab315	gremlin/gremlin	"/entrypoint.sh daem..." [...]

1.



Choose The Targets

Specify the coverage and details for impact.

Hosts

Containers

find hosts by tags...



endor

endor.localdomain

2.10.0

endor.localdomain

2.



Choose a Gremlin

Select the type of attack to unleash.

60 4

Category



Resource

Impact cores, workers, and memory



State

Process killer, shutdown and time travel.



Network

Blackhole, latency, packet loss and DNS

Attacks



CPU

Consumes CPU resources



Disk

Consumes disk space



IO

Consumes targeted file system devices resources



Memory

Consumes memory

3.

Practical example

(Execution)

74,4	Id,	0,1	wa,	0,0	hi,	0,0	si,	0,0	st
free,	1156,8	used,	1906,1	buff/cache					
free,	0,0	used.	6306,3	avail Mem					
S	SHR	S	%CPU	%MEM	TIME+	COMMAND			
0	6176	S	99,7	0,1	0:17.56	gremlin			
0	97384	S	0,3	3,0	1:32.20	gnome-she+			
4	95612	S	0,3	2,6	12:57.93	gnome-she+			
0	4148	S	0,0	0,1	0:14.32	systemd			
0	0	S	0,0	0,0	0:00.00	httplib			

Take-home message

- New and emergent practice
- Empirical methodology
- Provides new information about the system
- Researched at KTH

References

1. Gremlin. Introduction to Chaos Engineering with Gremlin's ceo & co-founder, Kolton Andrus (Video).. https://www.youtube.com/watch?v=F26_uBAyOM
February 2019 [Accessed Apr 29 2019]
2. Aaron Blohowiak Nora Jones Ali Basiri Casey Rosenthal, Lorin Hochstein. Chaos Engineering. <https://www.oreilly.com/ideas/chaos-engineering> [Accessed Apr 29 2019], September 2017
3. Long Zhang, Brice Morin, Philipp Haller, Benoit Baudry, and Martin Monperrus. A Chaos Engineering System for Live Analysis and Falsification of Exception-handling in the JVM. arXiv preprint arXiv:1805.05246, 2018.
4. 2018 IDG Cloud Computing survey
5. Gremlin Inc. Chaos engineering: Breaking your systems for fun and profit. <https://www.gremlin.com/uploads/20171210%20%E2%80%93%20Chaos%20Engineering%20White%20Paper.pdf> [Accessed: Apr 29 2019], December 2017.

Thank you for listening