Scaling Security Operations

Securing DevOps and Automating SecOps

CAPSULE8

Cybersecurity Skills Shortage

2 million

global shortage of cybersecurity professionals by 2019

84%

of organizations believe that half or fewer of applicants for open security jobs are qualified

53%

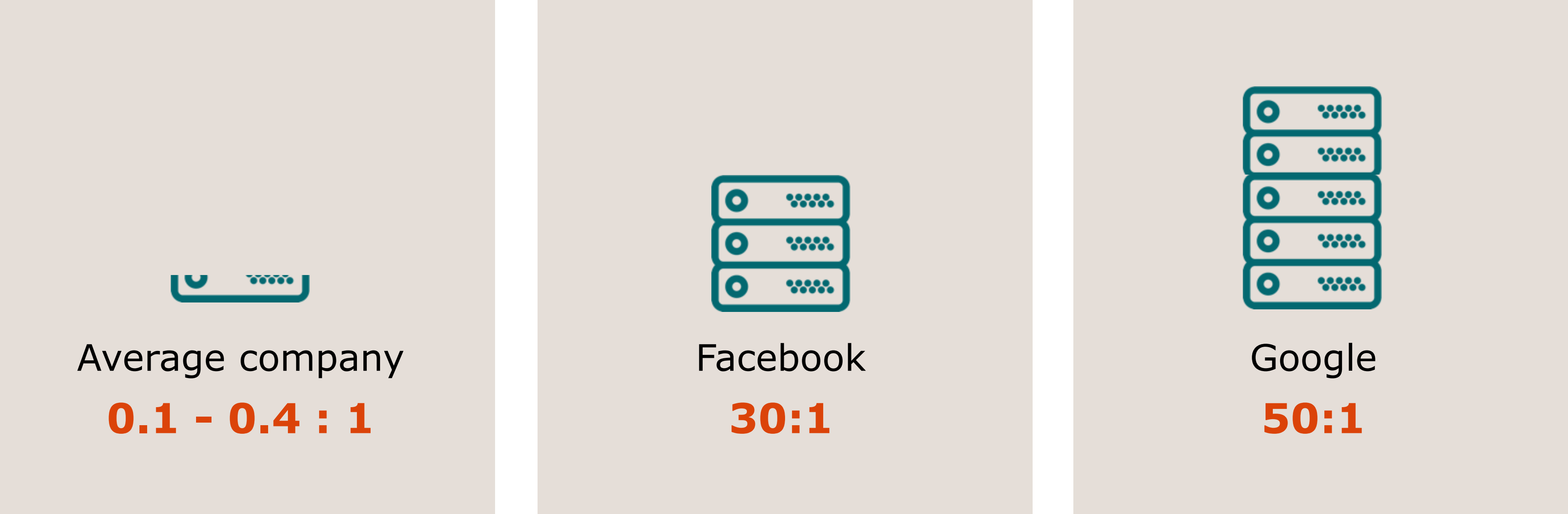
of organizations experience delays as long as 6 months to find qualified security candidates

3x rate

of cybersecurity job growth vs. IT jobs overall 2010-2014

Operations: Scaling with bodies vs. software

Servers Per Employee (Timothy Chou, 2013)



Average company

0.1 - 0.4 : 1

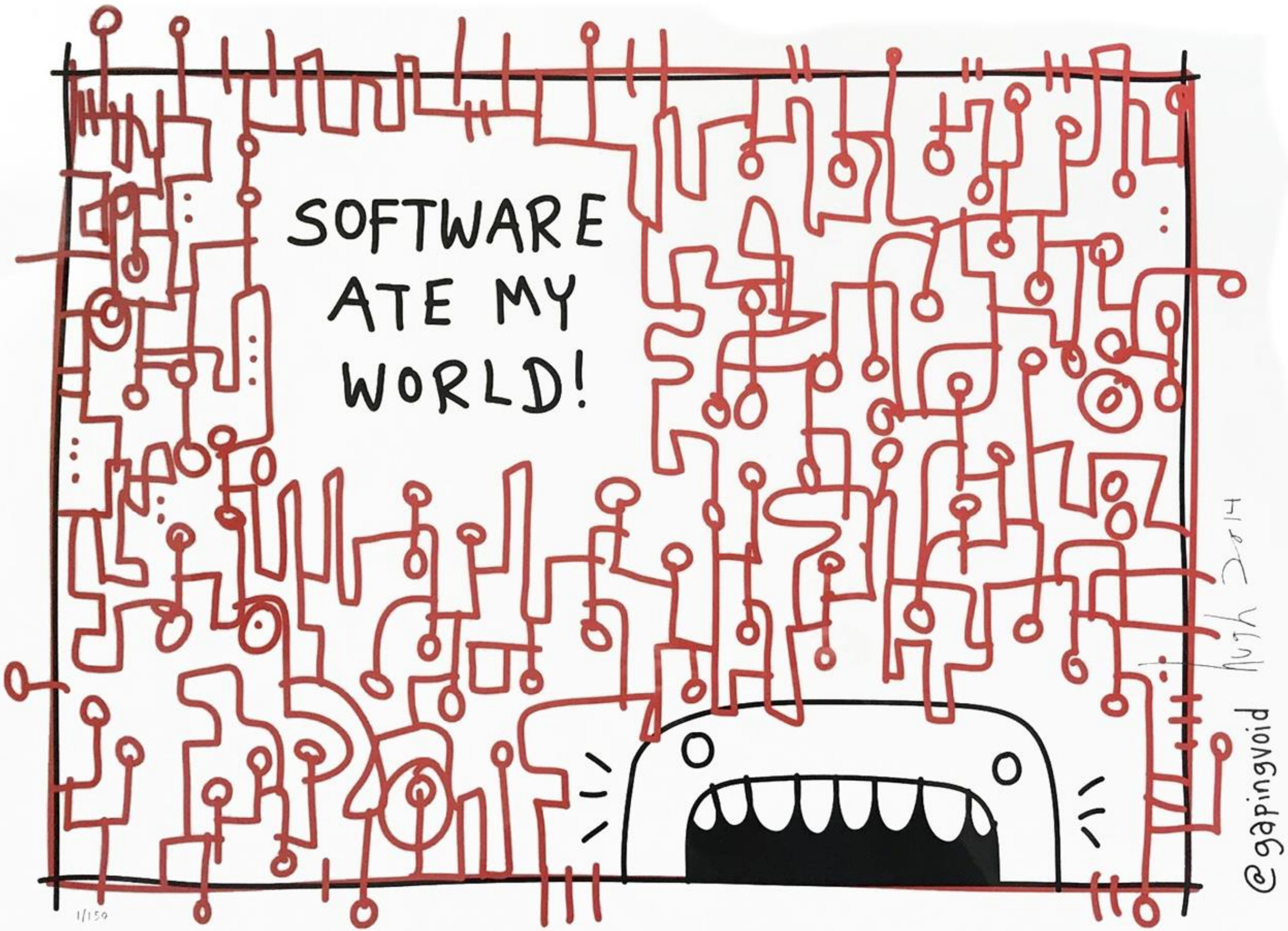
Facebook

30:1

Google

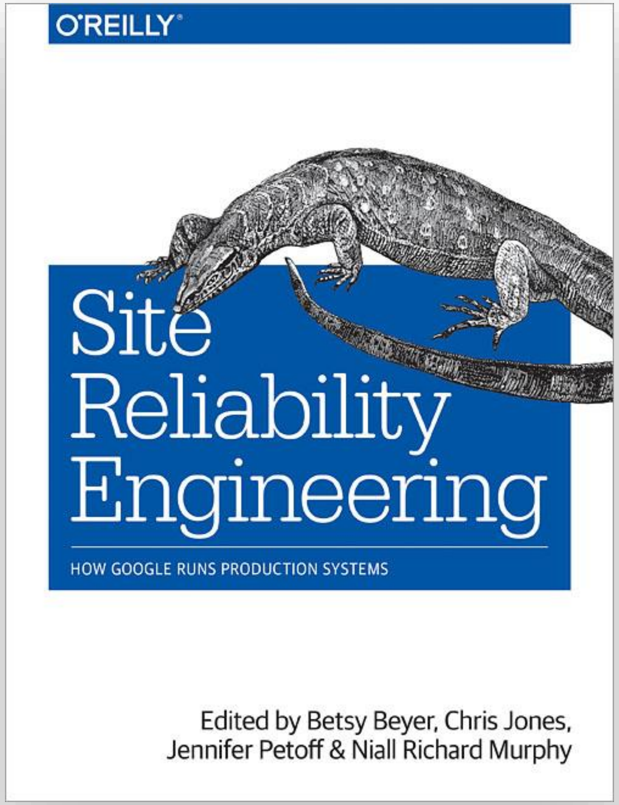
50:1

How did they achieve this scale? By treating operations like a software problem.


SOFTWARE
ATE MY
WORLD!
@gapingvoid
Hugh 2014
1/150

What Makes SRE, SRE (Ben Treynor 2014)

1. *Hire only coders*
2. Have an SLA for your service
3. Measure and report performance against your SLA
4. Use Error Budgets and gate launches on them
5. Common staffing pool for SRE and DEV
6. Excess Ops work overflows to DEV team
7. Cap SRE operational load at 50%
8. Share 5% of Ops work with DEV team
9. On-call teams at least 8 people, or 6x2
10. Maximum of 2 events per on-call shift
11. Postmortem for every event
12. Postmortems are blameless and focus on process and technology, not people


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Richard Murphy

Translating SRE Principles to Security

1. *Hire only coders*
2. **Have an SLA for your service**
3. **Measure and report performance against your SLA**
4. **Use Error Budgets and gate launches on them**
5. Common staffing pool for SRE and DEV
6. Excess Ops work overflows to DEV team
7. Cap SRE operational load at 50%
8. Share 5% of Ops work with DEV team
9. On-call teams at least 8 people, or 6x2
10. Maximum of 2 events per on-call shift
11. Postmortem for every event
12. Postmortems are blameless and focus on process and technology, not people

SecDevOps?

DevSecOps?

DevOpsSec?

Continuous Security.

Bringing the pain forward

> “If it hurts, do it more frequently, and bring the pain forward.”

Jez Humble

Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation

Deployment velocity as a fitness

CEO:

“I’ve been asked to spend 15 minutes with President Obama to provide recommendations on how he should fix HealthCare.gov. What would you say?”

CTO:

“Suggest to President Obama that the team should deploy to production every day.”

“Tools as a catalyst for culture change” https://medium.com BillHiggins/tools-as-a-catalyst-for-culture-change-f012b2c0b527

Fitness Functions for Security Operations

What hurts the most in security operations?

Vulnerability Management

Host Security

Incident Response

Good FFs roll up many other key performance indicators into one metric that is objective, comparable, clear, and can be automated

Good example from Ops: end-to-end service latency

Vulnerability Management (Prevent Exploitation)

Fitness function:

vulnerability lifetime in production

Identify the deployment of the vulnerable code or component, not just when vulnerability was discovered or announced

Behavior that it drives:

- Eliminating vulnerabilities before they end up in production
- Minimizing time of exposure between deployment and elimination
- Automating vulnerability identification and elimination

Host Security (Prevent Persistence)

Fitness function:

time-since-software-refresh

When was the last time that OS was installed from known good onto host?

Behavior that it drives:

- Decoupling of application from underlying OS (e.g. containerization)
- Resilience of applications to underlying host outages (e.g. orchestration)
- Automation of host software upgrades

Incident Response (Prevent Actions on Objectives)

Fitness function:

time-to-evict-from-initial-exploitation

When was the last time that OS was installed from known good onto host?

Behavior that it drives:

- Custom detection strategies and logic
- Shorter feedback loops between detection logic, responders, and infrastructure teams
- Automation of detection and response

Embarking on journey to Continuous Security

- ✓ **Embrace automation by building security teams with people who can code**
- ✓ **Automate gathering and reporting of fitness functions before anything else**
    - ○ Measure value of initiatives by their impact on your FFs
- ✓ **Shorten feedback loops between security and app development teams as well as infrastructure operations teams**
- ✓ **Require APIs from security products and services that facilitate and encourage automation**
- ✓ **Automate the repetitive**
    - ○ If you hire coders on your security team, they’ll do that without even being asked

Thank You!

@dinodaizovi
https://capsule8.com

CAPSULE8