**Mosquito Capital** **@MosquitoCapital** *Fri Nov 18 09:46:38 +0000 2022*

I've seen a lot of people asking "why does everyone think Twitter is doomed?"

As an SRE and sysadmin with 10+ years of industry experience, I wanted to write up a few scenarios that are real threats to the integrity of the bird site over the coming weeks.

For context, I have seen some variant of every one of these problems pose a serious threat to a billion-user application. I've even caused a couple of the more technical ones. I've been involved with triaging or fixing even more.

1) Random hard drive fills up. You have no idea how common it is for a single hosed box to cause cascading failures across systems, even well-engineered fault-tolerant ones with active maintenance. Where's the box? What's filling it up? Who will figure that out?

2) Physical issue with the network takes down a DC. I gather Twitter is primarily on-prem, and I've seen what happens when a tree knocks out a critical fiber line during a big news event.

3) Bad code push takes the site down. Preventing this was my day job, and I can tell you that it's one of the scariest scenarios for any SRE team, much less a completely understaffed and burnt-out one.

4) Bad code push takes the site down *in a way that also fucks up the ability to push new code*. This is the absolute nightmare scenario for teams like mine. When something like this happens, it's all hands on deck. Without deep systems understanding, you might never get it back.

5) Mystery SEV. Suddenly, the site goes dark. The dashboard is red. Everything seems fucked. There's no indication why. You need to call in the big guns. Teams with names that end in Foundation. Who are they? How do you call them?

6) Database is fucked. It's a big one. Everything is on fire. Who's the expert for this one?

7) Someone, say, entirely hypothetically, **@wongmjane**, finds a critical security vulnerability in your prod iOS app. You need to fast-track a fix, *stat*. You have a team of experts who know how to navigate Apple's Kafkaesque bureaucracy for app updates, right? I sure hope you do.

8) Someone notices that it's possible to read anyone else's DMs by loading up a particular URL. This is a SEV1, massive, all-hands-on-deck, critical issue. You need people who understand deeply how your privacy abstractions work, and how to fix them.

9) The site goes dark at 4am. The oncalls have no idea what's wrong. You *need* an IMOC (Incident Manager On Call) who knows who to wake up, why, and how. Someone who understands your systems, can synthesize information at lightning speed, and coordinate a recovery effort.

10) The system you use to *find other systems* internally goes down. None of your systems can talk to each other. The site, and all your tools, immediately fail. The tools you need to revert the breaking change are all FUCKED. Can you figure this one out with a skeleton team?

11) It's 5pm on a Friday. The dashboards all go red at once. The web fleet is seeing cascading reboots. The disks have been filling up since Wednesday. There were hundreds of code changes across multiple interlocking systems on Wednesday. Revert any of them at your own risk...

12) Oh shit. You reverted one of them. Now every locked account's tweets are visible to everyone. People might literally get murdered with machetes over their posts. That's not a hypothetical. It's now 9pm. The site is fucked. Who are you going to call?

13) The system that ensures server changes are safe to push to prod is failing. You have, say, 30000 tests that *must* run to ensure privacy/security/compliance/reliability. One of the tests is causing the failures. Can you find it? Also it's the

World Cup. Also the site is down.

14) A user in the Phillipines is about to post CEI to the platform. You *cannot* leave that content up. Do you have your employees with relationships with PH law enforcement? Do you have your content moderation systems working? Do you have your moderators?

15) The FBI wants to inspect the contents of the DMs of someone they think is about to commit 9/11 2: Atomic Boogaloo. Do you have a system to grant them access? Do you refuse them access? How do you know it's really them?

16) You grant them access. Now someone from a country known for horrific human rights violations is knocking. They have an official-looking subpoena. Do you let them see a dissident's DMs? Can you articulate why? You might need to, in a very official court somewhere in Europe.

17) Another country is telling you that they want all of your data on their users stored on servers in their country. Do you have policy experts for that country? Do you have a lot of *very* motivated lawyers? Do you have an infra eng who knows how to partition your data just so?

18) GDPR. You're found in violation. It took a team of 100s of engineers, lawyers, policy experts, designers, and managers months of "hardcore engineering" to be in compliance in the first place. Can you get back? I assure you, not doing so will cost more than an org's headcount.

19) Once a day, every day, at 12:13am, a specific service in your data pipeline slows to an absolute crawl. It doesn't seem to be causing any issues, but you're a bit concerned as it seems to be getting worse. Do you assign an SRE to take a look? Do you have any left?

20) The service you use to discover other services is working fine, but one of your best engineers does some calculations and realizes it won't scale to more users and more services, and (hypothetically) you want to build a super-app called X. Do you rewrite? What do?

21) You decide to rewrite. 8 months later (lol) your new system is ready to take on its first users. Who's coordinating the migration? Do they *really* understand complex systems? Are they good with people? Can they execute? Do they have the domain knowledge they need?

22) You just hired a great-seeming engineering director from Microsoft for a core org. Slowly, their org's productivity slows, and attrition climbs *way* up. The director swears everything is fine. If you fire the director, one of your VPs suddenly has like 18 reports. What do?

23) An engineer just kicked off a command to reboot the fleet. Oops, they didn't use --slow. Now all of your caches are empty. All of them. Every request goes straight to DB. The DBs all get overloaded instantly, some start to OOM and reboot loop... How do you refill the cache?

24) World Cup. It is *the* defining event. We used to have watch parties for the traffic charts. The amount of traffic your site gets in one week is mind-blowing. It's in huge bursts. It tests *every* system you have to its limits. If one breaks, hope it doesn't cascade. It will.

25) New Year's Eve, USA East Coast. Every year. I remember sitting outside the office, fireworks exploding in the distance, frantically calling the video oncall. Everyone posts videos of their fireworks. *Everyone*. It will fill up disks and test your bandwidth to the very limit.

26) I've said it before, but... CEI. If you mishandle it, if your policy people and lawyers are not top of the fucking line, you *will* get yanked in front of Congress, in front of judges, into the evening news, places you don't want to be if you're running a social media company

NSFL - For those asking, CEI is the acronym for child sexual/abuse imagery :(