Netflix – Written evidence (RSK0097)

1. We are grateful for the opportunity to submit written evidence to the Risk Assessment and Risk Planning Committee as part of its ongoing inquiry into the UK's resilience and approach to risk management. We were invited to provide this submission following reference by a witness in oral evidence to Netflix's use of Chaos Monkey to stress-test our streaming system. Chaos Monkey is one of several open source tools originally built by Netflix and freely available for others to use.[1]

2. The Netflix video streaming system is composed of many interacting services. In such a large system, failures in individual services are not uncommon, and indeed inevitable. With so many interacting components, the number of things that can go wrong in a distributed system is enormous.

3. There is a level of complexity in modern distributed systems that is chaotic. As more companies move toward microservices and other distributed technologies, the complexity of these systems increases. For this reason at Netflix we practise Chaos Engineering,[2] a disciplined approach to identifying failures before they become outages. By proactively testing how a system responds to failure conditions, we can identify and fix failures before they become public facing outages. Experimenting on a system in this way enables us to build confidence in the system's capability to withstand turbulent conditions in production.

4. Years ago, we decided to improve the resiliency of our microservice architecture. At our scale it is guaranteed that servers on our cloud platform will sometimes suddenly fail or disappear without warning. If we don't have proper redundancy and automation, these disappearing servers could cause service problems.

5. We needed an automated solution for ensuring the system is resilient to failures in non-critical services. While it is never possible to prevent all possible failure modes, many of the weaknesses in a system can be identified before they are triggered by events. Our objective is to prevent system-level failures. In particular, our goal is to reduce the likelihood of an outage, when Netflix customers are not able to stream videos.

[1] https://github.com/netflix/chaosmonkey

[2] https://principlesofchaos.org/

6. Created in 2010, Chaos Monkey is one example of Chaos Engineering in practice at Netflix. We created Chaos Monkey as a resilience tool, designed to randomly choose servers in our production environment and turn them off during business hours. Chaos Monkey enables engineering teams to run Chaos Engineering experiments on live traffic in production in order to build confidence that our service will be able to degrade gracefully in response to any failure of non-critical downstream services. Our ultimate goal is to be able to detect automatically whether a service is resilient to failure rather than relying on a human looking at dashboards and making a judgement.

7. This in turn helped ensure that our engineers implement their services to be resilient to instance failures. Without such a tool, we couldn't depend on the otherwise relatively infrequent occurrence to influence behaviour, but knowing that this would happen on a frequent basis created strong alignment among our engineers to build in the redundancy and automation to survive this type of incident without any impact to the many millions of Netflix members around the world.

8. In 2011, Netflix announced the evolution of Chaos Monkey with a series of additional tools known as The Simian Army, a suite of failure-inducing tools designed to add more capabilities beyond Chaos Monkey, each serving a specific purpose aimed at bolstering a system's failure resilience.[3] In summary, while we can't remove the complexity of modern distributed systems, through Chaos Monkey we can discover vulnerabilities and prevent outages before they impact our members. We value Chaos Monkey as a highly effective tool for improving the quality of our service.

20 May 2021

[3] https://netflixtechblog.com/the-netflix-simian-army-16e57fbab116