RESEARCH ARTICLE OPEN ACCESS

Chaos Engineering for Fintech Infrastructure Resilience and Fraud Prevention

Anirudh Mustyala
Software Engineering, Plano - Texas
Email: anirudhmusthyala@gmail.com

--************************----------------------------------

Abstract

Although conventional software testing methods contribute significantly to the security of fintech systems, these methods have various shortcomings that limit their ability to ensure the safety of fintech platforms. These methods tend to be more reactive than proactive, only test known vulnerabilities, and are suitable for less complex systems. Chaos engineering model is a novel security management paradigm that proactively mitigates vulnerabilities, is suited to complex distributed networks such as fintech systems, and provides a background for researching and mitigating unknown vulnerabilities. This writing discusses chaos engineering and how it can be leveraged to mitigate fraud in fintech entities.

Keywords: Fintech,Fraud,Platform, Data.

--************************----------------------------------

I. Introduction

In the 1960s, Edward Lorenz, a meteorologist at the Massachusetts Institute of Technology in Cambridge, noted that a computer could predict very different weather patterns from almost similar data inputs. He discovered that very minor differences in input data led to very diverse outcomes. He explained this phenomenon as the 'butterfly effect.' According to the concept, insect flaps in South America could set up conditions that would cause a tornado in North America (Ghys, 2015). This effect was later described as chaos theory. Chaos theory is the study of ostensibly arbitrary or unpredictable behavior in systems caused by deterministic laws.

Chaos engineering is based on chaos theory concepts. It involves testing distributed computer systems to ensure they can withstand unexpected disruptions. The objective of chaos engineering is to detect weaknesses in systems through controlled experiments that introduce random and unpredictable behavior in systems (Basiri et al., 2016). Fintech companies tend to have complex and elaborate systems. While traditional methods like unit testing, system testing, and acceptance testing play an integral role in ensuring the safety of these systems, they are not sufficient to guarantee security. Chaos engineering helps engineers to unearth system vulnerabilities that cannot be predicted easily. This writing discusses chaos engineering and how it can be leveraged to mitigate fraud in fintech entities.

II. BODY

What is chaos engineering?

As aforementioned, chaos engineering is the practice of introducing faults and failure scenarios in a system with the intention of testing its resilience in the face of random disruptions. The principal objective of chaos engineering is to determine potential failure points and remediate them before attackers can take advantage of them. Chaos engineering is based on the concept that minor disruptions can cause applications to respond unpredictably and cause monumental adverse impacts on the system. By injecting faults in the applications, engineers can gauge how the systems respond to these scenarios and optimize them accordingly. Chaos engineering is suitable for testing natural, technical, and malicious crises. For example, engineers may test how the system responds to earthquakes affecting the availability of data centers or cyber attackers injecting malware into system applications. Although chaos engineering can be applied in testing any type of application or system, it is primarily used in distributed systems.

Why chaos engineering

The software development life cycle (SDLC) leverages different application testing methods to enhance the security of cyber systems. While these conventional testing methods are instrumental in ensuring the safety of software systems, they are susceptible to various weaknesses that limit their effectiveness. These weaknesses include;

- **Only suitable for known risks**: Traditional testing methods, such as unit testing, are designed to focus on the known properties of the system (Tucker at al., 2018). This implies that emerging vulnerabilities are likely to go undetected by traditional methods.
- **Reactive testing**: These methods also tend to be reactive. Engineers must first detect a threat, learn about it, implement a solution, and then test the efficacy of the solution. Reactive testing is not suitable for systems that hold sensitive data and resources like fintech systems.
- **Not suitable for complex systems**: Conventional testing methods are ideal for simple and medium systems. However, testing complex integrated

systems using traditional methods is almost unviable. Conventional testing methods may be used to test individual components of complex integrated systems but may not provide comprehensive security insights for distributed systems.

Chaos engineering addresses most of the weaknesses of traditional testing methods. Rather than testing known risks, chaos engineering is based on experimentation. The model proposes hypotheses, which are then experimented through controlled simulations. The model reveals how systems cope under different situations, disclosing unknown information about applications. No traditional software testing methods can generate insights comparable to chaos engineering. Experimentation generates information that cannot be revealed by typical testing. Chaos engineering addresses systems vulnerabilities proactively. The model requires engineers to create hypotheses for potential vulnerabilities and test them. If the premises are proven, developers reconfigure the system to deal with such scenarios in the future. Chaos engineering is particularly meant for large-scale distributed systems. The model is well suited for testing systems with complex dependencies and evolving components. Complex systems also tend to have multiple failure points. Chaos engineering is well-specialized to test systems with numerous failure points.

Typically, systems used by established fintech companies are relatively complex and distributed. Chaos engineering provides a novel approach to effectively test how these applications perform under different strains. Most of the data stored by fintech companies is related to customers' financial information. Such information is sensitive and should never be exposed to unauthorized persons. Chaos engineering proactively mitigates vulnerabilities, protecting customers' data all the time. In general, chaos engineering is a unique testing model that specifically meets the testing needs of fintech applications. It is ideal for distributed systems and proactively mitigates issues before fraudsters can leverage them.

Principles of chaos engineering

For engineers to conduct compelling chaos engineering experiments, they must follow a set of guiding principles. These principles define how engineers can identify scenarios that are not tested by traditional methods, how to plan for the experiments, manage the simulation process, and what to do with the results. According to IBM (2023), the four main principles that guide chaos engineering are;

Experiment planning

The first principle that guides chaos engineering is planning the experiments. Prior to planning, engineers must have comprehensive knowledge about the system's normal behavior and what constitutes abnormal functioning. The planning stage must start with the formulation of hypotheses. A basic hypothesis must describe a possible vulnerability and how it can affect the overall functioning of the system. It is noble to also define metrics that will be used to measure the level of system normalcy. Such metrics can include latency and error rates.

Real-world events

Chaos engineering should experiment real-world events likely to undermine proper functioning of systems. Real-world events should be centered around hardware, servers, and other external events likely to cause system outages or malfunctioning, such as surges in traffic and cyberattacks. Focus on real-world events prevents engineers from paying attention to events less likely to happen.

Run experiments

After formulating hypotheses, defining system's normal and abnormal behavior, and deciding on performance metrics to measure, the next phase is carrying out experiment to collect actual results. It is recommended the experiment is conducted in real production environments to get more accurate results. However, it is a rule of thumb to minimize the blast radius when running experiments in production environments. This ensures adverse impacts are kept minimal in cases when the system does not cope well with the experiment. If the system seems resilient, the blast radius can be gradually increased until the entire system is tested. It is also advisable to automate and run experiments continuously. Running chaos engineering experiments manually can be labor-intensive and unsustainable.

Monitor results

The primary goal of chaos engineering is to collect results that can be leveraged to understand the resilience of a system. The experiments should collect both control and experimental results. Control results are vital for helping teams understand normal system behavior at any particular time. Any deviation of experimental results from control results can be traced to specific experimental actions.

Best practices

Chaos engineering is an intricate practice that can lead to unintended outcomes when not conducted properly. To meet the intended goals, engineers must adhere to various best practices. Some of the best practices for chaos engineering include;

- **Focus on critical parts**: During hypothesis creation, it is vital to prioritize the most important aspects of the system. In Fintech, servers and communication networks are the most critical components of the system.
- **Gradually scale-up experiments**: As aforementioned, it is clever to carry out chaos engineering experiments in confined environments. This helps in

minimizing the impact of the experiment on the entire system if the implications are dire. Only scale up if the effects are non-threatening. This can be done by introducing a minor disruption in a smaller component of the system and then increasing the blast radius and complexity of the fault in subsequent experiments.

- **Have a rollback plan**: Chaos engineering experiments can be unpredictable. Even with measures such as a limited blast radius, a simple experiment can easily bring the entire system down. It is essential engineers have a rollback plan when executing experiments. This allows faults to be reverted quickly, allowing safe abortion of experiments and return to normalcy (Splunk, 2023).
- **Measure impact**: Apart from measuring system resilience and observing how the experiment affects system performance, it is recommended engineers measure how the experiment affects customer success. This may include tracking metrics such as stream starts per second and orders per minute. These metrics are vital for determining when to stop the experiment. For example, when orders per minute or stream starts per second start slipping, it may mean the experiment is harming user experience or even limiting access to the platform. Chaos engineering experiments should never affect the usability or accessibility of the platforms being tested. If this happens, the experiment should be halted immediately.
- **Incorporate lessons in c-suit decision-making**: Chaos engineering is not only meant for engineers and low-level IT teams. The board of directors can also use insights

retrieved from chaos engineering experiments to make crucial decisions, such as key changes in technology stack and IT budgets. Information collected from experiments can be summarized and shared in the right format with fintech leaders.

III. Conclusion

Although conventional software testing methods contribute significantly to the security of fintech systems, these methods have various shortcomings that limit their ability to ensure the safety of fintech platforms. These methods tend to be more reactive than proactive, only test known vulnerabilities, and are suitable for less complex systems. Chaos engineering model is a novel security management paradigm that proactively mitigates vulnerabilities, is suited to complex distributed networks such as fintech systems, and provides a background for researching and mitigating unknown vulnerabilities. Chaos engineering can help fintech companies enhance the reliability and resilience of their IT systems, enhance user experience, proactively curb online revenue losses, and enhance confidence in systems. Fintech firms can take their cybersecurity game a notch higher by assimilating chaos engineering as part of their system management routines.

IV. REFERENCES

Basiri, A., Behnam, N., De Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., & Rosenthal, C. (2016). Chaos engineering. IEEE Software, 33(3), 35-41.

Ghys, É. (2015). The butterfly effect. In The Proceedings of the 12th International Congress on Mathematical Education: Intellectual and attitudinal challenges (pp. 19-39). Springer International Publishing.

IBM (2023), IBM's principles of chaos engineering. Retrieved From: https://www.ibm.com/cloud/architecture/architecture/practices/chaos-engineering-principles/

Splunk (2023), Chaos Engineering: Benefits, Best Practices, and Challenges. Retrieved From: https://www.splunk.com/en_us/blog/learn/chaos-engineering.html

Tucker, H., Hochstein, L., Jones, N., Basiri, A., & Rosenthal, C. (2018). The business case for chaos engineering. IEEE Cloud Computing, 5(3), 45-54.