VIEW POINT

SECURITY CHAOS ENGINEERING FOR IMPROVING CLOUD CYBER RESILIENCY

Infosys®
Navigate your next

OVERVIEW

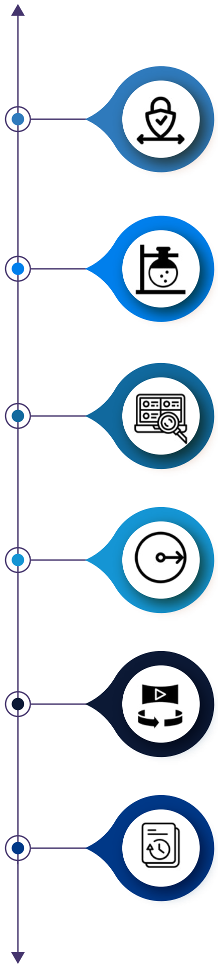
The world is changing at a pace that is faster than our imagination. Technology is driving disruptions across industries. All facets of enterprises are being challenged with diverse risks due to exponential growth in servers, devices, accounts, and networks. Many organizations have been shifting their workloads to the cloud to improve efficiencies and reduce operating costs. While cloud does offer good features, it also attributes to the overall system complexity. Most of the times due to the incomprehension of vulnerabilities, many challenges can be imposed on the overall security. Hence, a proactive approach to unearth the weak components and mitigate them is critical.

Security Chaos Engineering is one such proactive approach that can be considered as a next generation evolution of penetration testing. Penetration testing focuses on specific events whereas chaos engineering covers the entire cloud security posture. The key objective of chaos experiments is to check cyber resiliency through controlled but random experiments and identify potential failures before they turn into outages.

The Chaos engineering idea originated in Netflix in 2010 when the company decided to move to cloud based environments - AWS. Most of the architects were not confident about this transformation as they had limited control on the infrastructure. That was when they came up with the concept of chaos engineering. In simple terms, it is a way to learn systems by breaking things and checking if systems can recover gracefully.

Aaron Rinehart is a pioneer in security chaos engineering area. He was the first one to develop a tool called ChaoSlingr. This tool was used to simulate an experiment to open or close a port and check if firewall can detect the change and respond to it. He along with Kelly Shortridge have recently published a comprehensive guide on this topic on O'Reilly and it serves as a good reference.

PRINCIPLES OF CHAOS ENGINEERING



Define Baseline

Define baselines for each experiment to decide if the outcome of the experiment is a failure or success. It can vary based on the technology type and functionality. This baseline will be considered as steady state during entire experiment lifecycle.

Retractable Experiments

It is essential to identify experiments that are retractable, else, the failure could adversely impact applications, infrastructure, and environments.

Robust Observability

Without robust monitoring across layers, it would be difficult to trace sequence of events and decide remediation post chaos experiments. Observability robustness is decided by 3 powerful components – logs, metrics, and traces.

Minimize Blast Radius

Chaos engineering experiments can be highly disruptive. Hence limiting blast radius will ensure that systems are not experiencing downtime, while the damage is minimized, and other teams' work is not hampered. This can be achieved by running experiments in a controlled environment.

360 Degree Study

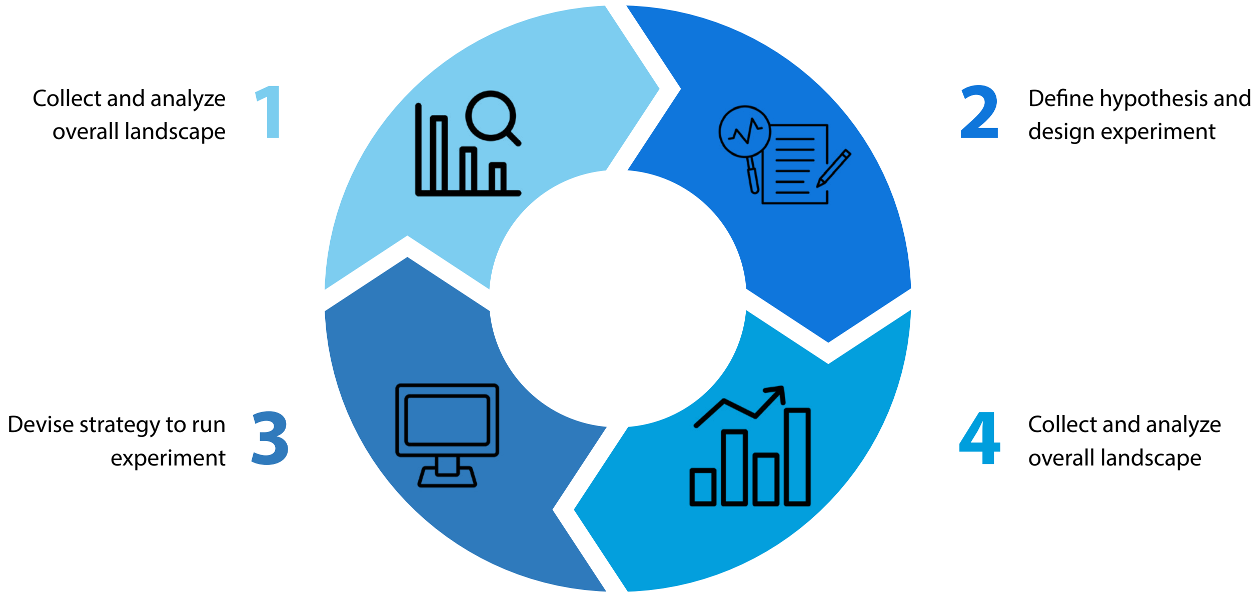
Having 360-degree study of cloud systems in scope, helps devise remediations or strategies to reduce unanticipated conditions during chaos engineering exercise.

What-if Scenarios

What-if analysis is a structured exercise to determine how changing parameters/conditions will impact interfacing systems/components/environments. This exercise will give informal insights on how the given situation can be handled.

Chaos engineering is disruptive in nature and hence the approach to security chaos engineering is iterative. It also needs a mindset change as majority of the testing teams focus on positive testing scenarios.

The security chaos engineering exercise comprises of 4 phases:



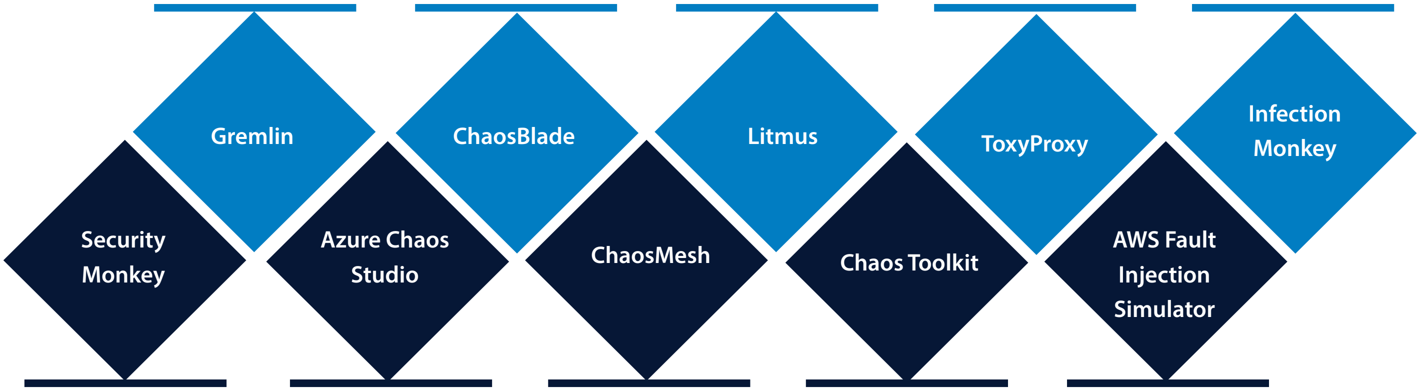
1 Collect and analyze overall landscape

2 Define hypothesis and design experiment

3 Devise strategy to run experiment

4 Collect and analyze overall landscape

There are multiple tools available in the market for carrying out chaos experiments and the same tools can be extended for security chaos use cases as well. The key ones are:



Security Monkey

An open-source tool that monitors and alerts AWS/GCP account anomalies and policy changes.

Gremlin

A commercial tool that supports multiple out of the box attacks like packet loss, network delays, CPU and Memory exhaustion.

Azure Chaos Studio

This is offered with subscription-based pricing with multiple out of the box experiments such as DNS failure, network disconnection with firewall rule, kernel faults apart from standard experiments related to CPU, memory, VMs and IO

ChaosBlade

– An open-source tool that supports application-level fault injection for Java, C++, and Node.js.

ChaosMesh

A Kubernetes native tool that supports multiple attacks including latencies, system time manipulation and kernel panics.

Litmus

Another Kubernetes native tool facilitating experiments such as terminate Amazon EC2 instances, simulate a disk failure in a Kafka cluster etc.

Chaos Toolkit

This is an open-source tool quite similar to Ansible. It supports multiple platforms and network failure simulations.

ToxyProxy

An open-source network failure injection tool that works as a proxy service.

AWS Fault Injection Simulator

This tool is offered as pay per use and supports multiple attacks like draining an ECS cluster or rebooting an RDS instance.

Infection Monkey

An open-source tool to test data center resiliency from Guardicore. It can infect a random machine and discover security risks in automated way. It supports multiple scenarios like credential theft and compromised infrastructure.

While most of these tools support various experiments, Infection Monkey is a new age tool from security standpoint as it gives actionable recommendations as well. It also provides visual map of network from attacker's perspective. So, it can be combined with other tools to get complete Security Chaos Engineering coverage.

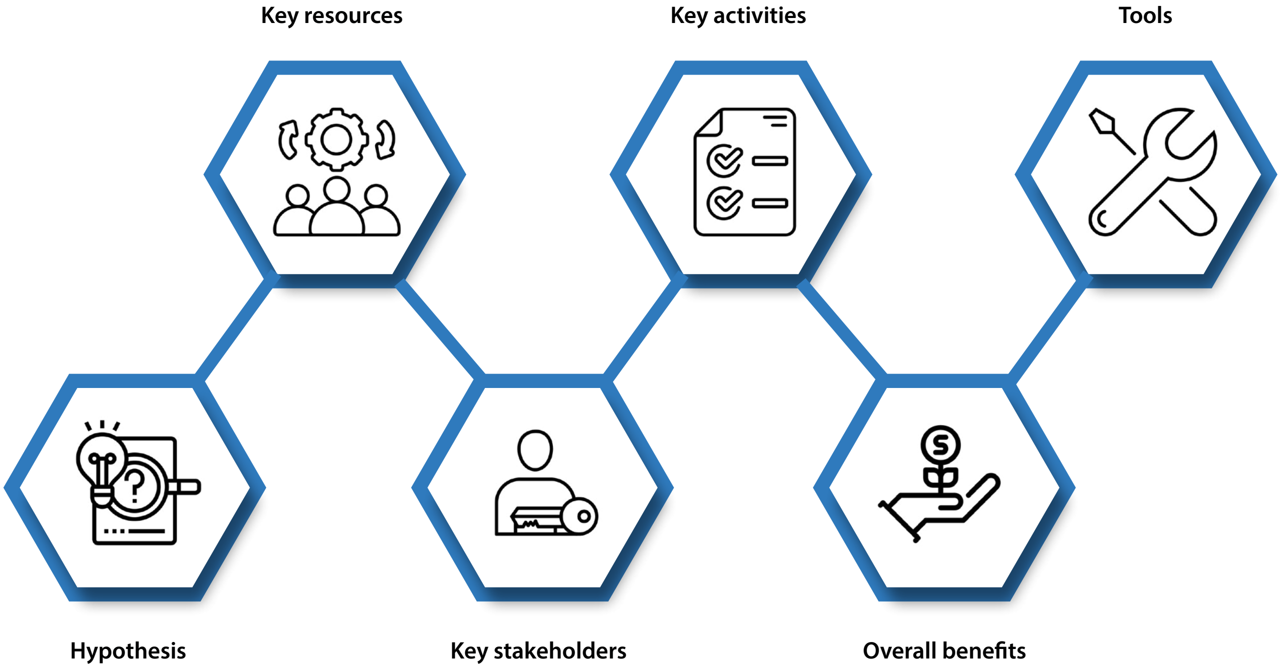
While there are tons of use cases that can be designed for security chaos engineering, the examples below are the sample experiments.

Sample Security Chaos Engineering scenarios

Area	Experiment	Steady State	Hypothesis	Tool
Identity and Access Management	Change critical policy	Depending on the change, access would be revoked or granted to individuals	Priority alerts are generated and concerned stakeholders are notified	Azure Chaos Studio or AWS FIS Injector or Gremlin or any other tool listed above
Identity and Access Management	Any server failure/CPU or Memory Exhaustion of any server	All access management requests will be processed	If a single server fails, rest of the servers should be able to handle the request. There will be no impact on the access requests. While the new server is becoming active, there could be increase in response time for the requests	Azure Chaos Studio or AWS FIS Injector or Gremlin or any other tool listed above
Identity and Access Management	Network delay in IAM processes	All access management requests will be processed	Failed requests are included in logs and available in audit trail reports	Azure Chaos Studio or AWS FIS Injector, Gremlin or any other tool listed above
Network Firewall	Change firewall rule with network delay	Depending on the change, communication will be affected between servers or ports	Failed requests are included in logs and available in audit trail reports	Azure Chaos Studio or AWS FIS Injector, Gremlin or any other tool listed above
Network Firewalls	Randomly disable firewall rules in controlled environment	Depending on the rule, communication between server and port will be prevented, or entire traffic will be blocked	Incident is logged, alerts are generated and concerned stakeholders are notified.	Python Script or any other relevant tool
Kubernetes cluster	Manipulation of rules in Azure Network security group with network delay	Rule changes are reflected in ASG	Priority Alerts are generated and concerned stakeholders are notified	Azure Chaos Studio
Unpatched Software	CPU/Memory exhaustion of patch management servers	All software updates are notified	If a single server fails, rest of the servers should be able to release patch notifications. There could be increase in response time for the requests.	Azure Chaos Studio or AWS FIS Injector, Gremlin or any other tool listed above
Credentials Leak	Expose credentials in network	Credentials are protected	Sensitive data is identified, and priority notification is sent	Infection Monkey
AWS – S3 bucket policy changes	Changes to user transport and encryption status	Encryption should be enforced all the time	Unencrypted access is denied, and alert is generated	AWS FIS Injector
Compromised Machines	Simulate network breach using brute force and similar safe exploits	Infected or compromised machine is isolated	High priority alerts are generated for infected or compromised machines and notifications are issues	Infection Monkey

SECURITY CHAOS EXPERIMENT CANVAS – TOOL TO MODEL EXPERIMENT OUTCOMES

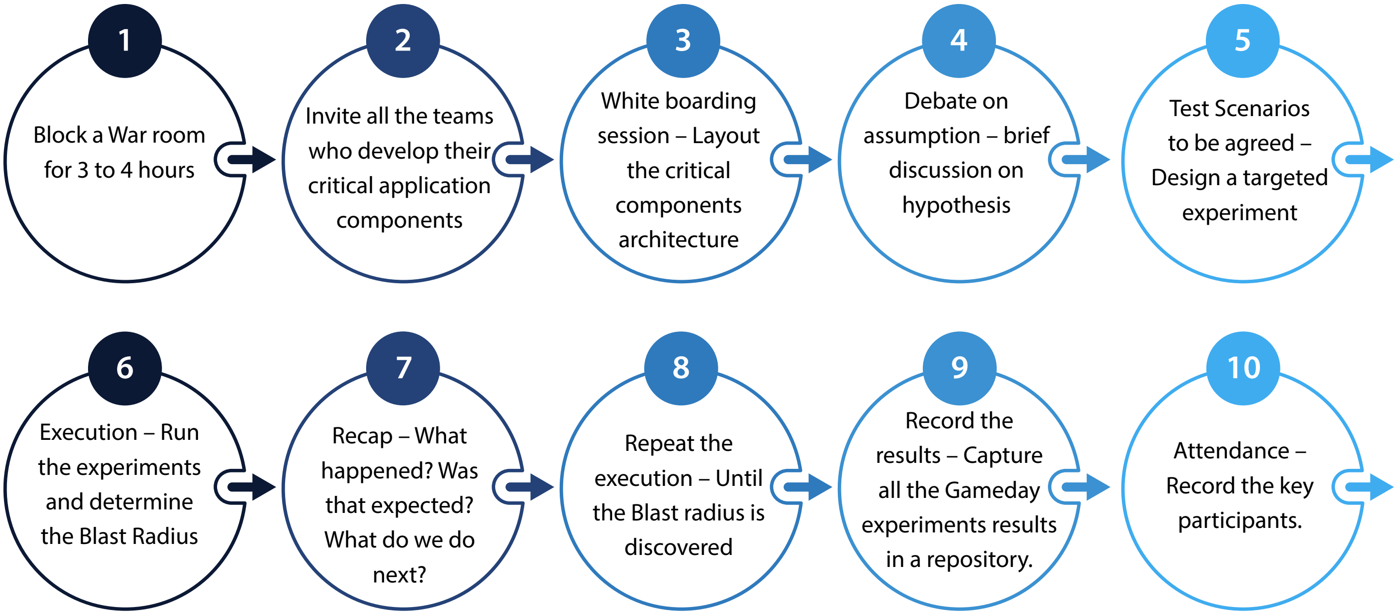
Chaos engineering is a complex activity and needs a disciplined approach to realize benefits. The Security Chaos Experiment Canvas is a tool that helps visualize each experiment dependencies, activities, and outcomes.



GAME DAYS – AN EFFECTIVE WAY OF RUNNING SECURITY CHAOS EXPERIMENTS

While running security chaos experiments, it is essential to ensure that the entire process is well coordinated. Game day is a systematic approach to run well-coordinated experiments and can be correlated to fire-drills. The success of game day depends on extensive planning, involving SOC and other relevant teams, and capturing the outcomes.

The below diagram depicts a typical Game Day.


1
Block a War room for 3 to 4 hours
2
Invite all the teams who develop their critical application components
3
White boarding session – Layout the critical components architecture
4
Debate on assumption – brief discussion on hypothesis
5
Test Scenarios to be agreed – Design a targeted experiment
6
Execution – Run the experiments and determine the Blast Radius
7
Recap – What happened? Was that expected? What do we do next?
8
Repeat the execution – Until the Blast radius is discovered
9
Record the results – Capture all the Gameday experiments results in a repository.
10
Attendance – Record the key participants.

CONCLUSION

It is impossible to have failure-free systems and hence understanding failures, simulating failures, and gracefully recovering from failures is extremely important. Organizations can get very good insights into cyber resilience through proactive security chaos experiments. Security chaos engineering will help in multiple aspects:

1. Security teams will get more experience in handling any new security control failure effectively.
2. Management can decide budget allocations based on the risk assessments.
3. It will help improve overall efficiency of telemetry and observability controls in place.

When these security chaos experiments are augmented with new age technologies like AI/ML and automation, more unknowns and anomalies in security functions can be discovered.

REFERENCES:

- PRINCIPLES OF CHAOS ENGINEERING - Principles of chaos engineering
- Breach and Attack Simulation | Infection Monkey (guardicore.com)
- Chaos Studio fault and action library | Microsoft Docs
- https://elib.uni-stuttgart.de/bitstream/11682/10918/1/bachelor_thesis__1_ %283%29.pdf
- SecOps with Security Monkey. Cloud Security | by Nag Medida | Medium
- Test S3 bucket policy using IAM simulator - k9 Security
- GitHub - dastergon/awesome-chaos-engineering: A curated list of Chaos Engineering resources.
- Building Trust & Confidence with Security Chaos Engineering (infoq.com)
- SCE Tools - Implementing Security Chaos Engineering 4/4 - Alice&Bob.Company (aliceandbob.company)
- GitHub - Optum/ChaoSlingr: ChaoSlingr: Introducing Security into Chaos Testing
- Security Chaos Engineering [Book] (oreilly.com)
- Security Chaos Engineering • Kelly Shortridge, Aaron Rinehart & Mark Miller • GOTO 2022 - YouTube

Author

Rohini Sathaye

Principal Consultant

With over 22 years of IT experience, Rohini has led performance engineering, quality engineering and SRE engagements for multiple customers across the globe. She is currently a part of Cybersecurity Automation team, where she focuses on delivering automation platforms that help accelerate incident response process.

For more information, contact askus@infosys.com

Infosys®
Navigate your next

Infosys.com | NYSE: INFY

Stay Connected