Chaos Engineering
By the Numbers
Hey there! 😊 Can I ask you a quick question?

Go for it!

Connect me to sales please!

I’m a Qualtrics user and need some help!

I’m interested in careers at Qualtrics

I’m just browsing

By using this chat service, I agree that Qualtrics may p...
qualtrics®

- 40+ teams
- < 3 months
- 500+ hrs saved
Mission

Build a more reliable internet.
Gremlin
Proactively improve reliability

Free stickers: gremlin.com/talk/SkillUp-Days
Chaos Engineering

Thoughtful, planned experiments designed to reveal the weakness in our systems.
Chaos Engineering

Thoughtful, planned experiments designed to reveal the weakness in our systems, improve our understanding of how systems work, so that we can improve them.
Chaos Engineering

1. There’s a process. It’s planned, not random.
2. Validate your mental models/understanding.
3. Use your knowledge to improve systems.
4. Build a practice and culture of learning
Blackholes

- 2nd most popular attack
- ~5700 attacks in Q1 2021
What percent of your incidents (SEV0&1) have been caused by:

<table>
<thead>
<tr>
<th>Incident Type</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad code deploy (e.g., a bug in code deployed to production leading to an incident)</td>
<td>39%</td>
</tr>
<tr>
<td>Internal dependency issues (non-DB) (e.g., a service operated by your company had an outage)</td>
<td>41%</td>
</tr>
<tr>
<td>Configuration error (e.g., wrong settings in your cloud infrastructure or container orchestrator causing an incident)</td>
<td>48%</td>
</tr>
<tr>
<td>Networking issues (e.g., ISP or DNS outage)</td>
<td>50%</td>
</tr>
<tr>
<td>3rd party dependency issues (non-DB) (e.g., lost connection to a payment processor)</td>
<td>48%</td>
</tr>
<tr>
<td>Managed service provider issues (e.g., cloud provider AZ outage)</td>
<td>61%</td>
</tr>
<tr>
<td>Machine/infrastructure failure (on-prem) (e.g., a power outage)</td>
<td>64%</td>
</tr>
<tr>
<td>Database, messaging, or cache issues (e.g., lost DB node leading to an incident)</td>
<td>58%</td>
</tr>
<tr>
<td>Unknown</td>
<td>66%</td>
</tr>
</tbody>
</table>
Let’s get into some numbers
State of Chaos Engineering

- 400+ respondents
- Large range of company sizes
- Broad range of company ages
- Mix of industries
- Varied maturity levels

gremlin.com/state-of-chaos-engineering
Who is running Chaos Engineering attacks?
(by size of org)
So how do you do it?
Scientific Process

1. Question
2. Hypothesis
3. Testing
4. Analysis
5. Share & repeat
Scientific Process

1. **Question:**

   Is my app / system reliable?
Scientific Process

1. **Question:**

   Is my app/system reliable?

   Does my app/system work the way I think it does?
Scientific Process

1. Question
2. Hypothesis:

I think given $CONDITION, my app / system will $REACTION
Scientific Process

1. Question
2. Hypothesis
3. **Testing:**
   
   Insert failure or replicate the condition
Scientific Process

1. Question
2. Hypothesis
3. Testing
4. Analysis:
   What did you learn?
   What can be improved?
Scientific Process

1. Question
2. Hypothesis
3. Testing
4. Analysis
5. **Share & repeat:**
   - Share your learnings!
   - Repeat to test improvements
Start Small & Increase the Blast Radius
Where are they running Chaos Engineering attacks?

- Dev/Test: 63%
- Staging: 50%
- Production: 34%
What Chaos Engineering attacks are they running?
More results & what’s next?
What results can you expect?

- Increased availability: 47%
- Reduced mean time to resolution (MTTR): 45%
- Reduced mean time to detection (MTTD): 41%
- Reduced # of bugs shipped to production: 38%
- Reduced # of outages: 37%
- Reduced # of pages: 25%
Teams who frequently run Chaos Engineering experiments have >99.9% availability.

23% of teams have a mean time to resolution (MTTR) of under 1 hour & 60% under 12 hours.
Thank you!
Questions?

Stickers: gremlin.com/talk/SkillUp-Days
Report: gremlin.com/state-of-chaos-engineering

Jason Yee
Director of Advocacy, Gremlin
jyee@gremlin.com
@gitbisect