Total Chaos
“How Experimentation Leads to Greater Control”
IN THIS SESSION WE WILL COVER
- 25 min Presentation
- 15 min Demo
- 5 min Q&A
Mike Zhou
Software engineer @ UHG
b
The Challenge: We are Large & Complex

- Fortune 5 Company
- 380+ Companies & Growing
- 28,000+ Developers
- 8,000+ Applications
- Highly Regulated
- Diverse Technology Portfolio
- Security Testing: Mostly Human Driven
- Cloud Journey: Mixed

- Largest Private HealthCare Company in World
- 1000+ Security Professionals
- Multinational Business
- Some DevOps
- Waterfall, Agile, & Others
2018 Causes of Data Breaches

- Human error: 27%
- System glitch: 25%
- Malicious or criminal attack: 48%
Security Incidents are Subjective in Nature
We don't know very much

Where?  Why?  Who?

How?  What?
How do we find out typically?
SECURITY INCIDENTS ARE NOT DETECTIVE MEASURES
IS NOT A STRATEGY
Teams spend too much time reacting to outages instead of building more resilient systems.
Build Confidence in What Actually Works
Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system’s ability to withstand turbulent conditions.

“Chaos Engineering is the discipline of experimenting on a distributed system in order to build confidence in the system’s ability to withstand turbulent conditions”
Netflix Chaos Monkey

Netflix

SMBIAN ARMY
Who is doing Chaos?
Don’t just test

....Experiment
Testing vs. Experimentation
Use Chaos Engineering to derive Objective Feedback about Security
Validate Security Incident Runbooks
Discover new insights about Security Toolchains
Build a Learning Culture around Security
Drive Security Transparency
Safely explore failures within system defenses
Drive out redundancy
Discover System Observability Gaps
Improve Quality of System Events
Fail Forward.

Focus is on the immediate remediation of the problem not the deeper understanding of how the system is behaving
Failure Happens.

Saturday, January 13

Emergency Alert
BALLISTIC MISSILE THREAT INBOUND TO HAWAII. SEEK IMMEDIATE SHELTER. THIS IS NOT A DRILL.

Slide for more
Build a Learning Culture around Failure
Humans & Systems need failure to learn & Grow
Am I Sure It Works That Way?
How would I know?
How it Works

Plan & Organize GameDay Exercise

Develop & Evaluate Chaos Experiment

Execute Live GameDay Operations

Conduct Pre-Incident Review

Automate & Evangelize Results & Take Action

Chaos Experiment

Plan & Organize

Develop & Evaluate

Execute

Conduct

Automate & Evangelize

Pre-Incident Review

Results & Take Action

GameDay Operations

Operations
AN OPEN SOURCE TOOL
ChaoSlingr Product Features

• ChatOps Integration
• Configuration-as-Code
• Example Code & Open Framework

• Serverless App in AWS
• 100% Native AWS
• Configurable Operational Mode & Frequency
• Opt-In | Opt-Out Model
Chaos Environment

https://docs.aws.amazon.com/waf/latest/developerguide/tutorials-4xx-blocking.html
Chaos Environment

https://docs.aws.amazon.com/waf/latest/developerguide/tutorials-4xx-blocking.html
Simple Config change to CDN

Experiment as Code built into Terraform or CloudFormation for documentation and automation engine to execute

```
Metadata:
Experiment: 
Hypothesis: |
if someone deploys cloudfront while using a different configuration for IPV6 enabled, our WAF should still be able to interpret the access logs generated by cloudfront and block 4XXX errors at a threshold.
StateChange: 
IPV6Enabled: true
```
Find the chaos

[merge_current_blocked_requesters] Start
[merge_current_blocked_requesters] Download current blocked IPs
[merge_current_blocked_requesters] Expire Block IP rules
[merge_current_blocked_requesters] End
[lambda_handler] Update new blocked requesters list to S3
[write_output] Start
[write_output] End
[lambda_handler] Update WAF IP Set
[get_ip_set_already_blocked] Start
[get_ip_set_already_blocked] End
[update_waf_ip_set] Start
[update_waf_ip_set] Truncate [if necessary] list to respect WAF limit
[update_waf_ip_set] Remove IPs that are not in current outstanding requesters list
[update_waf_ip_set] Block remaining outstanding requesters
[update_waf_ip_set] Commit changes in WAF IP set

An error occurred (ValidationException) when calling the UpdateIPSet operation: Update failed, invalid CIDR block: cannot parse 2601:281:8000:6f60:e416:c2fd:28e5:e2bb/32 as a CIDR
[update_waf_ip_set] Retrying in 1 seconds...

An error occurred (ValidationException) when calling the UpdateIPSet operation: Update failed, invalid CIDR block: cannot parse 2601:281:8000:6f60:e416:c2fd:28e5:e2bb/32 as a CIDR
[update_waf_ip_set] Retrying in 2 seconds...

An error occurred (ValidationException) when calling the UpdateIPSet operation: Update failed, invalid CIDR block: cannot parse 2601:281:8000:6f60:e416:c2fd:28e5:e2bb/32 as a CIDR
[update_waf_ip_set] Retrying in 4 seconds...
[update_waf_ip_set] Failed ALL attempts to call API
[update_waf_ip_set] End

END RequestId: 144ae83b-d555-11e8-84ad-f3550560a96a
REPORT RequestId: 144ae83b-d555-11e8-84ad-f3550560a96a Duration: 8507.83 ms Billed Duration: 8600 ms Memory Size: 512 MB Max Memory
Another experiment

Test if a bad deployment on the API gateway triggers WAF rule updates cascading to other services
THANK YOU @SANS
#SecDevOpsSummit!
QUESTIONS