**Opportunities**

- Create better (automated) workflow between incident response and RCA. Ex: based on labels (P1 S1) within an issue, generate an RCA from it when the issue closes.

- Engineering group has made some changes recently and now prioritizes keeping things up and running. In addition, Security is building out their own infrastructure and tooling.

- Consider more granular access controls

- Allow custom fields (which can be set as required and can’t be removed) to help generate certain metrics, e.g. mean time to acknowledge the page.

**Problems**

- Too reliant on Infrastructure team building out tools

- Transparency is a company value, but it can compromise privacy within the org. Often, sensitive info should not even be exposed internally (between teams).

- Can’t find data points between issues, which leads to inconsistent processes, and limited ability to classify by incident type which would help adjust the threat model and with project prioritization

- Cumbersome to edit description of timeline and often isn’t completed. Currently has to be done in a Google Doc

- Asynchronous culture and distributed team doesn’t lend to people (reporter or pager) sticking around to help collect information and begin triage

- RCAs required for S1 P1 to help prevent these problems from reoccurring, but team maturity doesn’t always lend to this (there’s been a lack of resources and new team members have to be trained). Tedious, manual process to copy and paste info from issue into the RCA.

**Actions**

- Create an Incident Response issue type with a timeline feature built in

- Reporters/ pagers should be available to help, even synchronously

- Page (PagerDuty) notifies SecOps and creates an issue (incident identification)

- Responders acknowledges page, reviews issue, engages with pager for clarifications

- Responders creates a Zoom bridge and invites relevant collaborators

- responder acknowledges page, reviews issue, engages with pager for clarifications

- responder creates a Zoom bridge and invites relevant collaborators

**Stages**

- Collect information

- Triage & Mitigation

- Documentation

**Incident Response Flow**

- Containment (contain the incident, limit the damage)

- Determine impact / risk

- Solve the issue & mitigate the risk to the org

- Ensure the vulnerability is fixed

- Try to replicate the problem and update on findings continuously

- Possibly create a runbook, documenting code snippets used

- Write an RCA (Root Cause Analysis)

- Fill out the incident issue based on the Google Doc, and close it

- Post-mortem (lessons learned)

- Ensure the vulnerability is fixed

- Solve the issue & mitigate the risk to the org

- Determine impact / risk

- Try to replicate the problem and update on findings continuously

- Possibly create a runbook, documenting code snippets used

- Write an RCA (Root Cause Analysis)

- Fill out the incident issue based on the Google Doc, and close it

- Post-mortem (lessons learned)