Plan For Final Project (post WK5 milestone)

Phase 1 ***(COMPLETED)***

1. Intake a select amount of data
2. Set up initial Lambdas for rule processing, adding context, and severity assignment
3. Transforming that data into a standard machine readable format e.g. JSON w/ schema
4. For demo purposes connecting that information to Slack for alerting but could be generalised into other systems such as a SIEM, PagerDuty etc.

Structure:

1. Stand Up at Beginning Or Tuesday And Thursday
2. Write The Things We Have Been Working On (Summary) By Sunday Night
3. Update The Trello board

Phase 1B (fixing things from phase 1)

1. Get tags as a part of the event OR query them using API (jas + talia)
2. Route events from S3 to user identity lambda (emma)
3. Context lambda - events from EventBridge don't have all the fields → just need to test that this works after EventBridge is fixed (gordon)
4. SH to slack (immanuel)
5. Review any feedback from AWS mentors and implement smaller changes (jas + talia)
6. [use case 1 - PII] Adding PII context from Macie findings - (jas + talia)
7. **add Custom data identifiers to macie console**
    a. **Passport**
    b. **License number**
    c. **Birth Certificate**
    d. **Citizenship**
    e. **(Energy bills) - extension**
    f. **Frequent flyer number → any type of account number**
    g. **Length of time that you have lived at address**
    h. **TFN**
    i. **Car rego**

Phase 2 (adding more sources/use cases)

1. Improving automatic severity escalation/de-escalation
    a. Calculate severity
        i. Take into account:
            - geo location being out of operating zone
                → easy to implement
            - The actual information that the user is accessing (i.e. if it has been tagged with PII) ⇒ ***Macie severity***
                → easy but with medium dependencies
            - How wide the bucket visibility is
                → easy s3.getBucketPolicy
                - Look at definition of 'public' on AWS
                - https://docs.aws.amazon.com/AmazonS3/latest/userguide/access-control-block-public-access.html
            - Level of encryption that the bucket has
                → easy
                - When encrypt in s3 on server side, it is transparent → if you have access to the bucket, they inherently get the decrypted data back.

- Only not true if we use a customer managed CMK ? Manage key policy
- Team of people responsible for managing s3, but dont want them to be able to read the s3
- Separate policy
- Access to buckets, but not the key
- Customer Managed CMK
- Option to do **client side encryption** → no key in AWS → only useful in this context → **would have to tag that?**
    - How do we detect this?
    - Put object api specify encryption type
    - NEED TO FOLLOW UP ON THIS
    - What happens if we do get object? wil l it allow us to pull the data without key? → if yes, then the use case is valuable

- Privileges of the user
  → easy
- **Other suspicious actions conducted recently (need to wait for Phase 2 (4))**
  → hard

b. Implement drop down for human to select severity → medium, but need to finish (1)(a)(i) first
- https://api.slack.com/reference/block-kit/blocks
- https://app.slack.com/block-kit-builder/T01N9HUT3CH
- More features/responses
- button that sends a reply with more info
- button that deeplinks to aws console
- button that opens a page with more info?

2. Creating a post incident report
Dashboard/Report = basic full display of all notes/related events chronologically e.g. static webpage
   a. Set up report using custom date range
   b. Use security hub findings to put into a report
   c. Filtering based on the concept of "incidents" where different events are grouped together (based on different tags)
   ~~d. Question for jas: how we can create graphs with the data we have and have them dynamically change as we push more data.~~
   e. Resolve: Report or dashboard
   f. https://www.chartjs.org/samples/latest/
   g. https://echarts.apache.org/en/index.html
   - **Potentially add findings to Security Hub instead of building a UI and don't have to add an additional tool**
   - **Can make insights with a custom view and query on the back end**
   - **Show alerts from 'product name' that are 'open' and 'unassigned'**
   - **Create insights within SH insights**
   - **Post incident report**
3. Storing incident metadata/state in a database so it can be updated on-the-fly
   a. Database with findings + raw events; **OR**
      - Difficult
   b. Pull security hub findings - can store custom key values
      - Medium but with less value

- Might be doing too much, less use cases, more depth.

- State might be too far
- if we do, use Dynamo DB

4. Creating logic statements with the improved contextual information we have acquired
    a. Including more descriptive analysis of what happened and what makes the finding so severe.
    b. Remove irrelevant information from alert but keep it in the finding (keep alert short and sweet) - include link to details
        - Easy/Medium to implement
        - Difficult to think through the logic
5. Ops teams can create their own basic rulesets
    a. Allow user to create custom logic defined in DSL (JSON?YAML?) to add custom context in order to escalate/de-escalate severity (an eval statement? Let them write python)
6. Option for employees to add more custom identifiers

Other/Questions:

-
- Order of step functions → should it be event first and then identity

TODO:

- Brainstorm step function:
    - Identity in parallel with context lambda, and bring together the results of both of them.
- Look into SNS - multiple lambdas pulled into one event

Questions from mentor

- We found a very similar code, can we use the same structure?
- Explain what we have decided to do
- Feedback from darran

Phase 3 *(Extensions)*

1. Adding times when there are planned changes - Change windows
2. SecOps can remove extraneous events from an incident, SecOps can add events they think are related
3. Remediation
    a. Overview of all active incidents, ability to “resolve” them and for them to re-open
    b. Automated actions to quickly lock down a resource while investigating
    c. Slack actions to make it easy for personnel to escalate or remediate and immediately respond to threats
4. SecOps can manually add notes or escalate/de-escalate events according to their own reasoning
5. SecOps can assign individuals to incidents so they can track remediation processes
    a. Alternatively teams can simply use Slack or a more advanced SIEM to manage that.

Feedback

Maintain state/sliding window for event management
- S3 storage
- DynamoDB for key/value store
    - Natively supports json
- Redis

Pre-existing state machine e.g. call S3/CloudTrail directly
or incorporate dynamoDB storage

Identity federation
- IAM role + attributes

Scope canary bucket to read only, anyone on the account

Presentation length could've been cut down