Google Cloud

Security Command Center から始めるクラウドセキュリティ運用

工藤 淳真
Classi株式会社 Python エンジニア

滑川 智也
Classi株式会社 データエンジニア

発表の概要

本発表では Classi で実施している Security Command Center (SCC) を用いたセキュリティ運用を 紹介します

- 話すこと
  - Security Command Center を用いたセキュリティ運用をどう進めたか
  - 運用する上で利用した Google Cloud のサービスと設定方法
  - 運用した上で学んだノウハウ
- 話さないこと
  - クラウドセキュリティとはなにか
  - 体系立てたベストプラクティス

目次

- はじめに
  - 会社紹介
  - Security Command Center の概要と導入経緯
- Security Command Center によるアプリケーション開発のセキュリティ対策
- Organization 全体での Security Command Center 運用
- おわりに

自己紹介

工藤淳真 (クドウジュンマ)

- Twitter: @irisuinwl
- github: @irisu-inwl

Classi 株式会社 2020年9月入社

Python エンジニア

学習チーム / Google Cloud Admin チーム

- アプリケーション開発
- Google Cloud 管理・運用
- クラウドセキュリティ

はじめに

教育プラットフォーム「Classi」

Classi は、4つの基本サービスを通じて先生の授業・生徒指導や、生徒の学び・成長をサポートする教育プラットフォームです。

Classi

学びの軌跡を保存し振り返る
ポートフォリオ
多面的・総合的評価への対応
- 日々の振り返りの蓄積
- 面談時の進路指導に
- 受験時の出願作成に

生徒一人ひとりに最適な学習
アダプティブラーニング
知識・技能の効率的な習得
- ベネッセテストに連動したレコメンドで自学自習
- 豊富な動画・問題コンテンツ
- 自動集計で採点不要

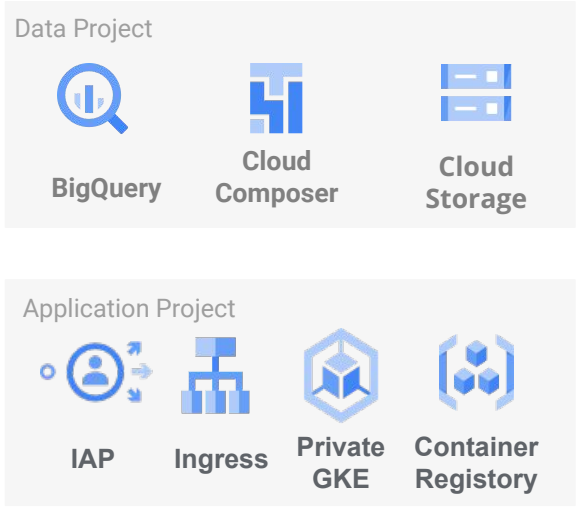
Classi ID 1つで様々なアプリを
プラットフォーム
様々な学校の課題をITで解決
- 探究学習用プログラム
- 英語4技能対応アプリ
- プログラミング教育
- いじめ検知

双方向の情報共有
コミュニケーション
先生・生徒・保護者の情報共有を円滑に
- 校内の会議をペーパーレス化
- 学校からの連絡もオンラインで

Classi での Google Cloud 活用

用途ごとにサービスを構築しており、40 程度の Project が存在している

- Classi のデータを活用するためのデータ基盤
- アプリケーション実行基盤として Kubernetes Engine
- 社内向けツール
- Firebase Analytics
- etc...



Security Command Center の概要と導入経緯

Security Command Center (SCC) の概要

Google Cloud 上のセキュリティ脅威の検知および脅威の可視化を行う
ダッシュボードを提供するサービス

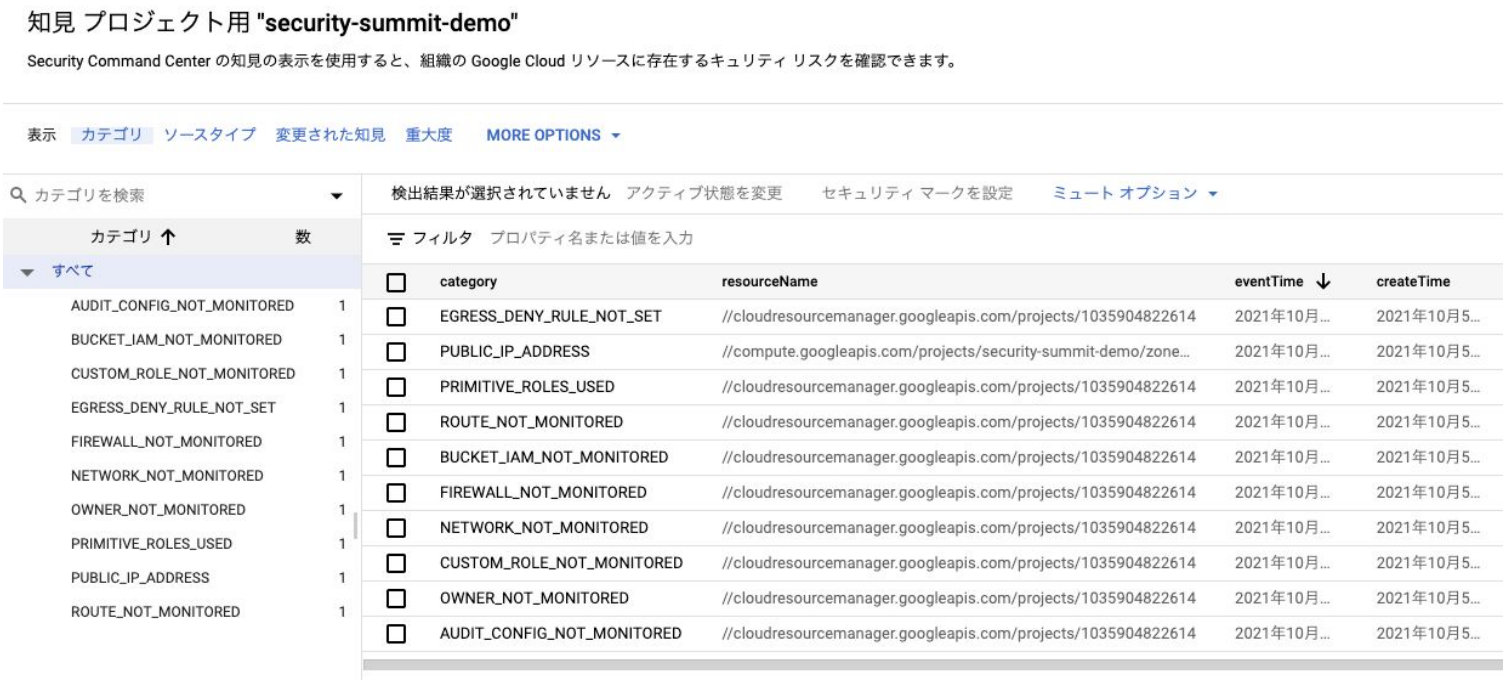
スタンダードティア (無料版) とプレミアムティア (有償版) がある

プレミアムティアで利用可能な機能

- **Event Threat Detection**: ログ内から脅威の痕跡を検知
- **Container Threat Detection**: コンテナ内の脅威を検知
- **Security Health Analytics の拡張**: 幅広い範囲のリスクをスキャン
- **Web Security Scanner**: アプリケーション一般の脆弱性スキャン
- **組織単位の Pub/Sub 自動エクスポート**

Security Command Center 検知例

Project 中の検知した項目を列挙する



Security Command Center 検知例

対処方法が明示されるため管理者のセキュリティスキルに依存しない

PUBLIC_IP_ADDRESS

この知見の重大度は high です。

概要

攻撃対象領域を小さくするには、VM にパブリック IP アドレスを割り当てないようにします。インスタンスは、停止している間もパブリック IP 検出の対象のままになることがあります。たとえば、起動時にエフェメラルなパブリック IP アドレスが割り当てられるようにネットワーク インターフェースが構成されている場合です。停止しているインスタンスのネットワーク構成に、外部アドレスが含まれていないことを確認してください。

改善

1. Compute インスタンスの詳細ページに移動します。
2. [編集] をクリックします。
3. [ネットワーク インターフェース] にあるインターフェースごとに、[外部 IP] を [なし] に設定します。
4. [完了] をクリックし、[保存] をクリックします。

影響を受けるアセット

instance-1

//compute.googleapis.com/projects/security-summit-demo/zones/asia-northeast1-b/instances/8989523388774065623

最終変更日時

2021/10/05

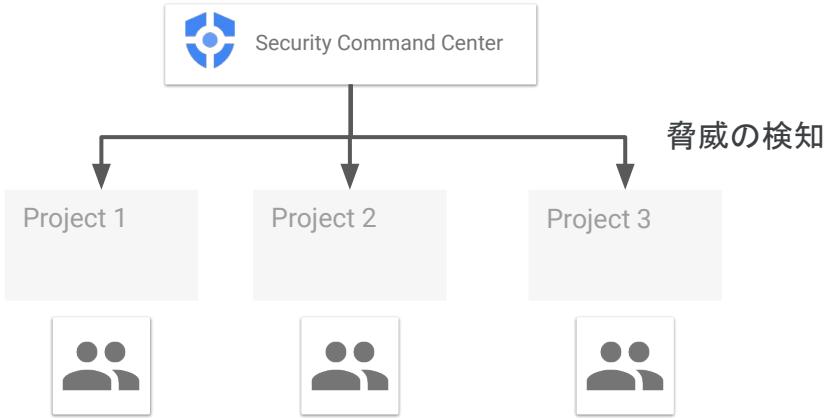
18:30 (4 分前)

Security Command Center の導入経緯

2020年のセキュリティインシデントを機に運用を見直し

それまで個々にセキュリティ対策を委譲していた体制の統合管理を期待し Security Command Center を導入

Classi では 2020 年 11 月からスタンダードティアで運用開始、2021 年 2 月にプレミアムティアに移行



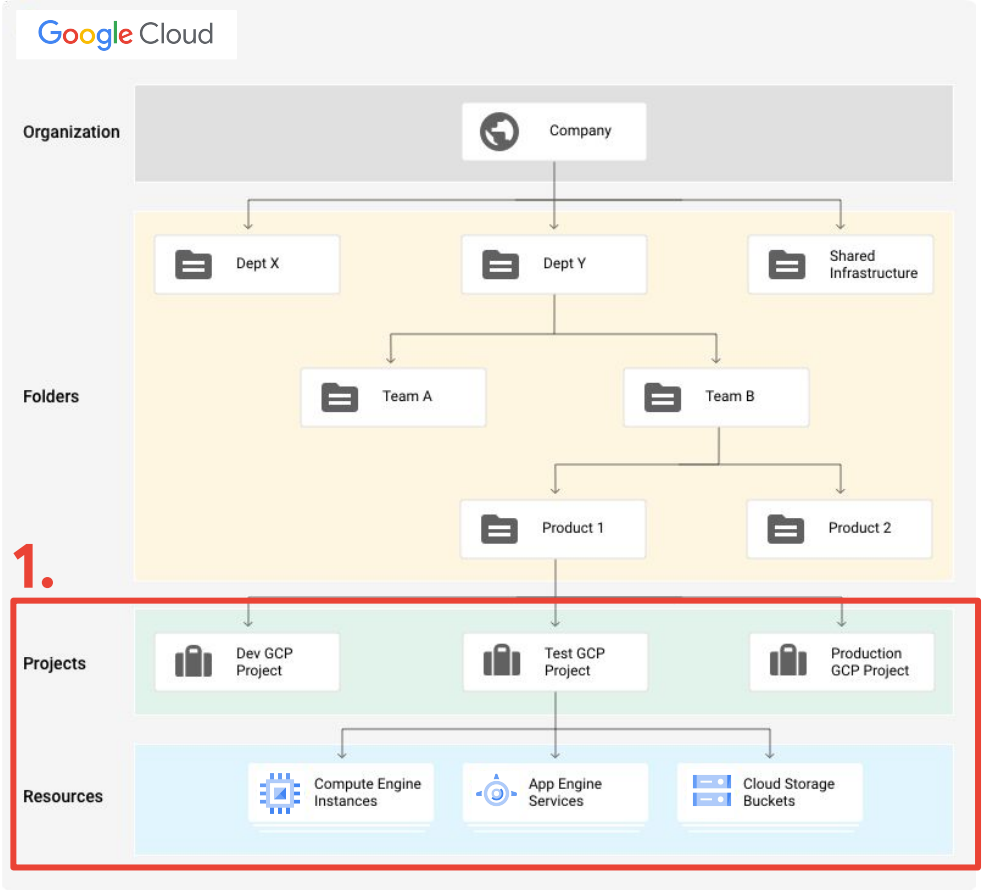
各オーナーは Security Command Center の検出項目を対応

参考) Classi株式会社：脆弱性を可視化し脅威検出も可能。対応策の提案機能を有効活用し、統合管理でセキュリティをさらに強化

発表の流れ

1. Project レベルでの Security Command Center 活用

2. Organization レベルでの Security Command Center を用いた管理・運用



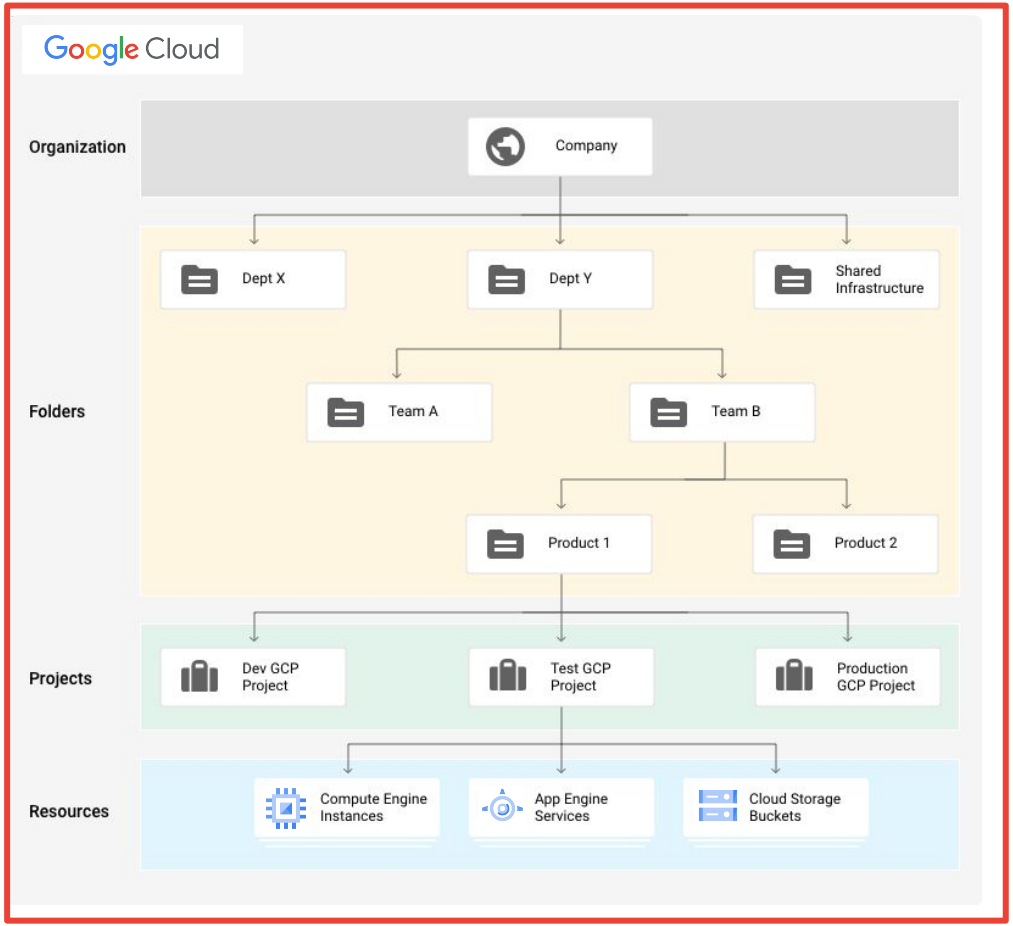
(https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy)

発表の流れ

1. Project レベルでの Security Command Center 活用

2. Organization レベルでの Security Command Center を用いた管理・運用

2.



(https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy)

Security Command Center によるアプリケーション開発のセキュリティ対策

Project の説明

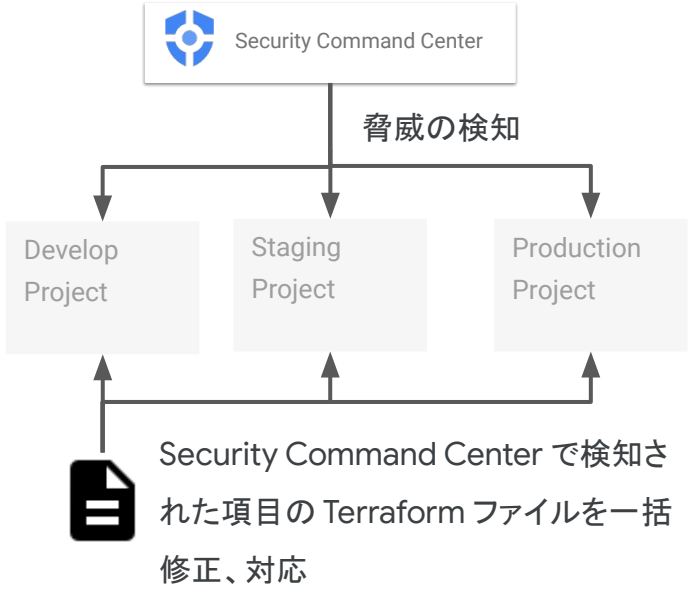
インフラ管理の前提

- アプリケーション開発プロジェクト
- 開発環境、ステージング環境、本番環境ごとに Project を分割
- インフラ管理には Terraform を利用し、workspace によって各環境の設定を管理

インフラストラクチャに対する対策

Security Command Center を用いて下記の手順で対策を実施

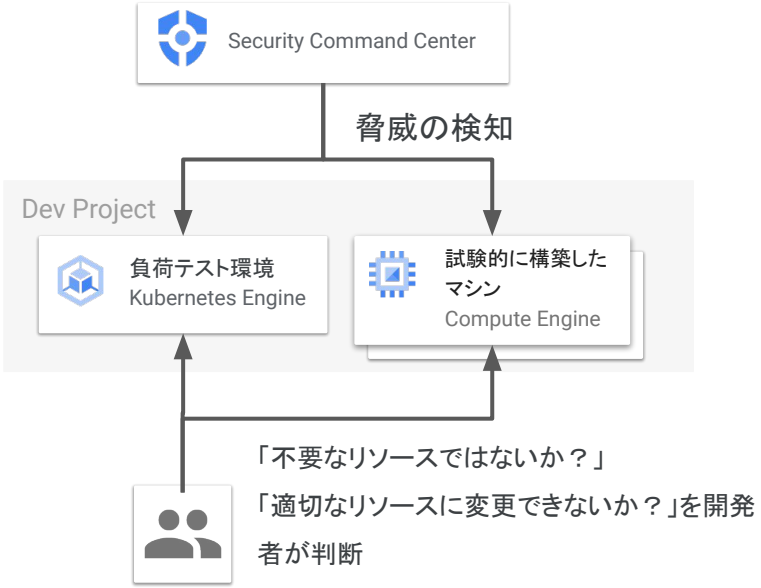
- Security Command Center により、各環境で共通の脅威を検知
- Terraform variable の共通項目を修正することで容易に対策することが可能



インフラストラクチャに対する対策

開発環境のように変更が多い環境では、Security Command Center の検知から適切なリソースへの変更ができた

- 「アプリケーションの負荷テスト用の GKE は Autopilot mode に出来ないか？」を検討・移行
- 試験的に作成した内部向けサービスを検知、現在必要かを検討、削除

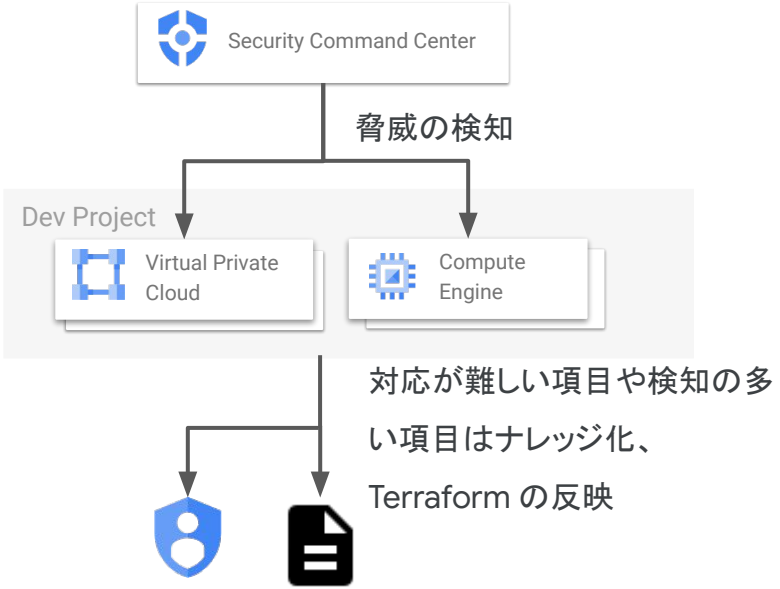


インフラストラクチャに対する対策

Security Command Center でよく検知される項目をまとめ、以降検知されないように対策

- インフラを構築する際に検知されないような設定をナレッジ化
- Terraform ファイルの設定

よく検知される項目で組織ポリシー制約で対策可能な項目は組織ポリシー制約で対策（組織ポリシー制約については後述）



Google Kubernetes Engine に対する対策

コンテナ利用のアプリケーションを開発する上で必要な GKE のセキュリティ項目を検知・対策

- 自動修復の有効化
- Pod Security Policy の有効化
- Workload Identity の有効化
- Private Cluster と Master Authorized Network の有効化

これまでは Security Command Center を
用いて対策できる内容を紹介したが
以降は Security Command Center では
カバーできない部分の対策を紹介

GKE に対する対策 (Security Command Center 範囲外)

Security Command Center で
対策できない部分

- Image Vulnerability Scanning
  - Container Registry 上のイメージの脆弱性スキャン
- Kubernetes Engine のバージョン戦略
  - 基本的にクラスタのバージョンは自動更新を設定
  - バージョンを手動更新するクラスタの場合は更新を通知、開発者が更新を確認、更新の実施を判断

Image Vulnerability Scanning の画面

フィルタ　脆弱性のフィルタリング

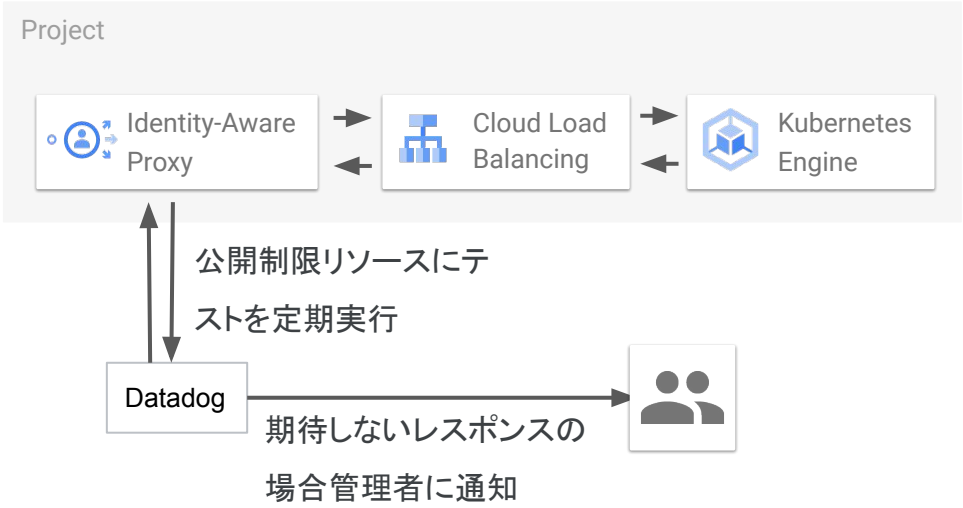
名前	有効な重大度	CVSS	使用可能な修正	パッケージ	
CVE-2019-19814	重大	9.3	–	linux	表示
CVE-2021-33574	高	7.5	–	glibc	表示
CVE-2021-38300	高	7.2	–	linux	表示
CVE-2021-3177	高	7.5	–	python2.7	表示

バージョン手動更新戦略

外部公開制限リソースの外形監視

外部公開制限すべきリソースに対して公開制限できているかの監視は Security Command Center では対応できない

- 内部ツールは Identity-Aware Proxy または Cloud Armor の IP 制限で外部公開制限
- Datadog を用いて外部公開制限すべきリソースにリクエスト、期待するレスポンスが返るか定期実行し監視

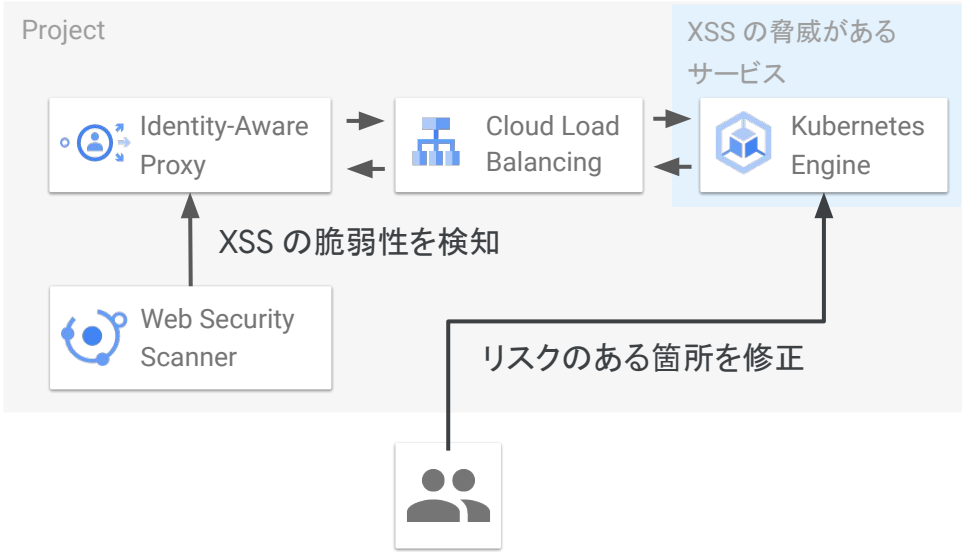


検知のテスティング

各種検知に対して、対応までのテストを実施

- Web Security Scanner:
  脆弱性をもつアプリケーションをデプロイ、検知から解決までのシナリオ実施
- Container Threat Detection:
  コンテナ脅威を検知、解消までをテスト
- Event Threat Detection:
  脅威を検知、解消までをテスト
- Security Health Analytics
  リソースのリスクのある設定を検知、解消までをテスト

(例) Web Security Scanner のテスト



Organization 全体での Security Command Center 運用

自己紹介

滑川 智也 (なめかわ ともや) @tomoyanamekawa

Classi 株式会社 (2019年5月入社)

データ AI 部 データエンジニア

データプラットフォームチーム / Google Cloud Admin チーム

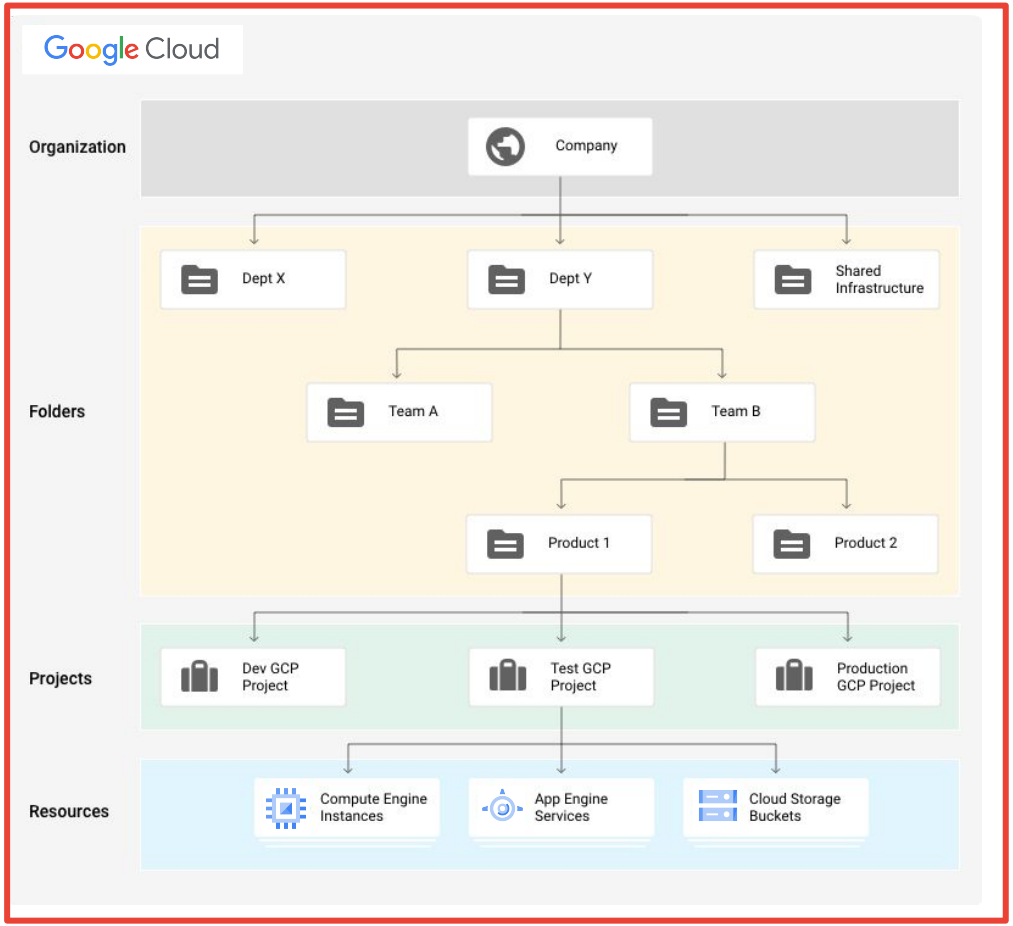
- データ基盤構築
- データマネジメント
- データ活用支援
- Google Cloud の Organization 管理・運用
- etc...

発表の流れ

1. Project レベルでの Security Command Center 活用

2. Organization レベルでの Security Command Center を用いた管理・運用

2.



(https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy)

Organization レベルでのセキュリティ対策の必要性

「Security Command Center 入れたから各 Project よしなにやってね」では対策が不十分

- 前半に紹介した対策も Project によって対応のレベルに差が出る
  - そもそも Project 管理者ごとに用途も意識も違う
    - 同じ問題意識を持てないので、対応へのモチベーションに差がある
    - Project の用途によって「どこまで対応するべきか」が変わってきて判断できない

- Organization 独自の項目や全体でまとめて対応した効率が良いものがある
  - 全体で統一の対応になっていることに意味がある

→ 当たり前だが、Organization レベルでのまとまったセキュリティ対策が必要

Project レベルでの管理との違い

Organization で管理するときに意識する必要があるポイント

管理対象数の違い

- Classi の Project は 40程度あるので単純に数が多い
- リソース数はその数百倍ある

Organization, Folder の概念

- Project の上位概念として Organization, Folder が加わる
- Organization, Folder 特有のリソースや機能もある

1つ1つの Project に詳しくない

- 限られた管理者が全ての Project について背景や実装を把握することは現実的でない

Security Command Center ドリブンでのセキュリティ対策

セキュリティ対策したほうがいい部分は思いつくが、全体像を描けず、何からどうやって進めていくのがいいかわからなかった
→「Security Command Center で検出されたものに対応すること」にスコープを狭めた

運用を考えた時の方針：

- わからないことが多いので、とりあえず早く始めることを優先
  - 使っていく中で、セキュリティへの知見を貯める
  - Google Cloud が社内で使われるようになるほど、変更の影響が大きくなり、試行錯誤しづらくなる
- 社内では AWS 側での対応が進んでいるので、そちらに足並みを揃えることを目標にする

運用フロー

01

初回設定

Security Command Center が診断をしてくれるように設定する。

02

診断・検知

存在するリソースやログに対して、診断する。

脆弱性や好ましくない設定、不審なイベントを検知する。

03

通知

検知されたものを Slack 通知して、必要な人にすぐに気づかせる。

04

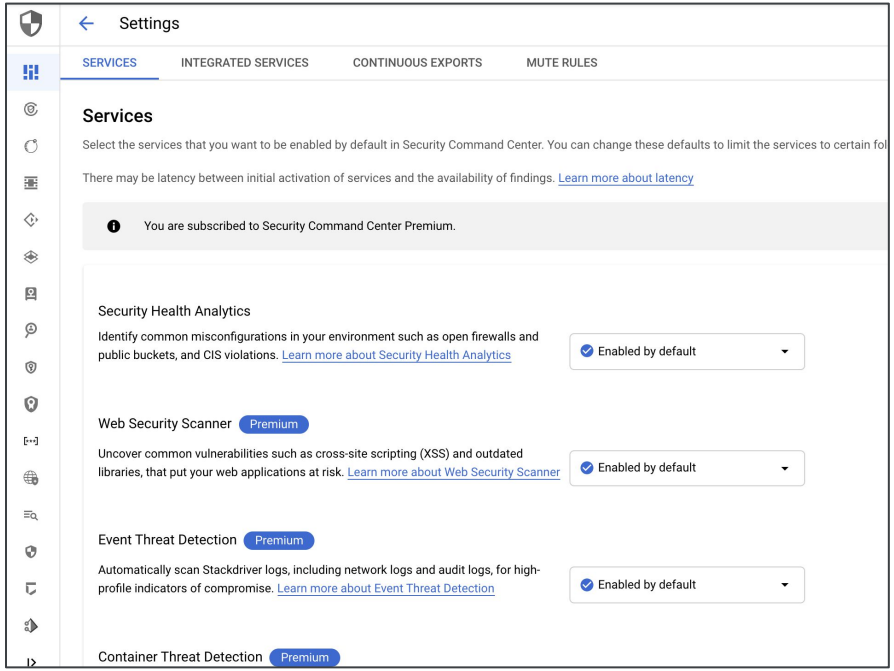
対応

発生済みの脅威を解決する。

運用フロー 01: 初回設定

- Settings で4つの機能を有効にする
  - Security Health Analytics
  - Web Security Scanner
  - Event Threat Detection
    - ログの設定
  - Container Threat Detection

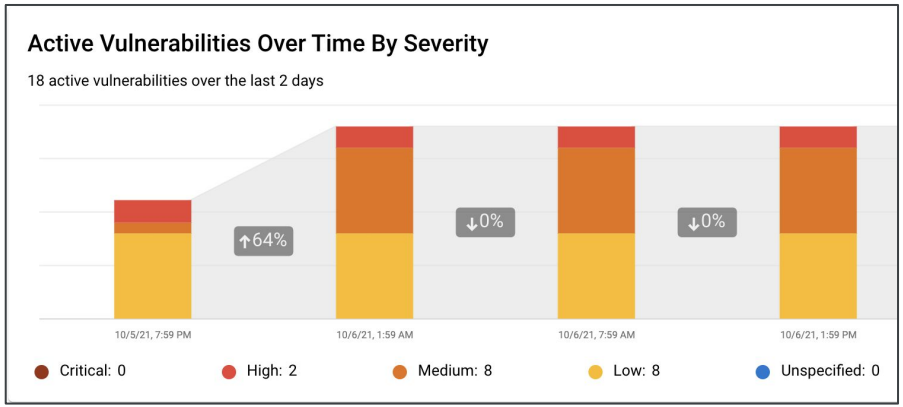
- そのほかに細かい設定もある
  - スキャン対象の Project を指定
  - Web Security Scanner がスキャンできるようにネットワーク設定
  - etc...



運用フロー 02: 診断・検知

Security Command Center が自動でやってくれる

- 定期スキャン
- 新規リソース作成時のスキャン



FINDINGS BY CATEGORY | FINDINGS BY RESOURCE TYPE | FINDINGS BY PROJECT

Filter Enter property name or value

Severity	Finding Category	Total Findings
!!!	PUBLIC_IP_ADDRESS	2
!!	COMPUTE_PROJECT_WIDE_SSH_KEYS_ALLOWED	2
!!	COMPUTE_SECURE_BOOT_DISABLED	2
!!	DEFAULT_SERVICE_ACCOUNT_USED	2
!!	OS_LOGIN_DISABLED	1
!!	PRIMITIVE_ROLES_USED	1
!	AUDIT_CONFIG_NOT_MONITORED	1
!	BUCKET_IAM_NOT_MONITORED	1

運用フロー 03: 通知

必要な人にすぐに Slack 通知して気づかせる

- Security Command Center の検知結果を Pub/Sub + Cloud Functions でメッセージ送信
- Severity に合わせてメンション先をわけて、対応のレベルを変える

※アーキテクチャは後述

[通知のイメージ]

@google-cloud-admin
検出したよ！

category
PUBLIC_IP_ADDRESS
project_name
security-summit-demo
severity
HIGH
チケット
{チケットのURL}

検出したよ！

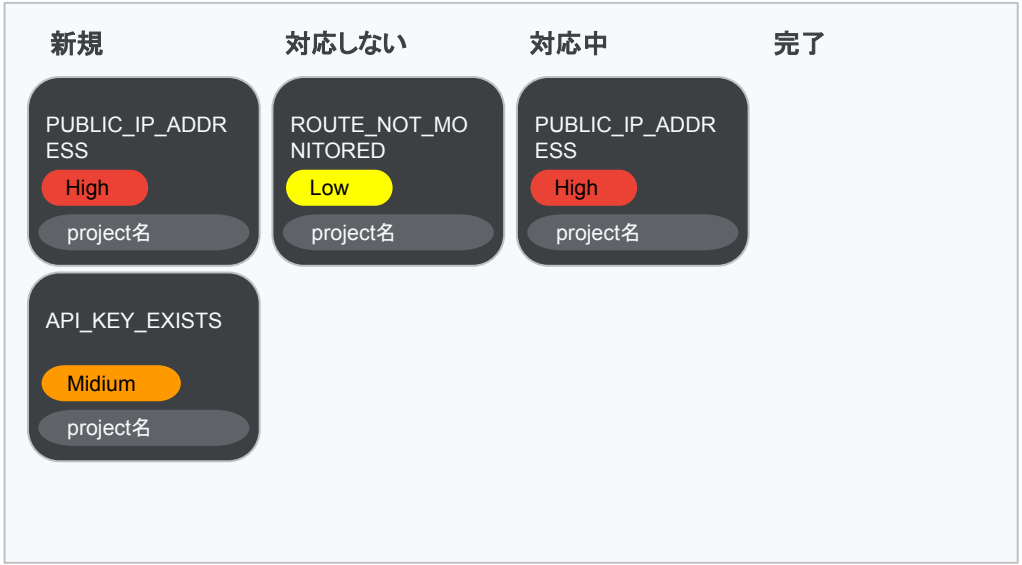
category
PUBLIC_IP_ADDRESS
project_name
security-summit-demo
severity
LOW
チケット
{チケットのURL}

運用フロー 04: 対応

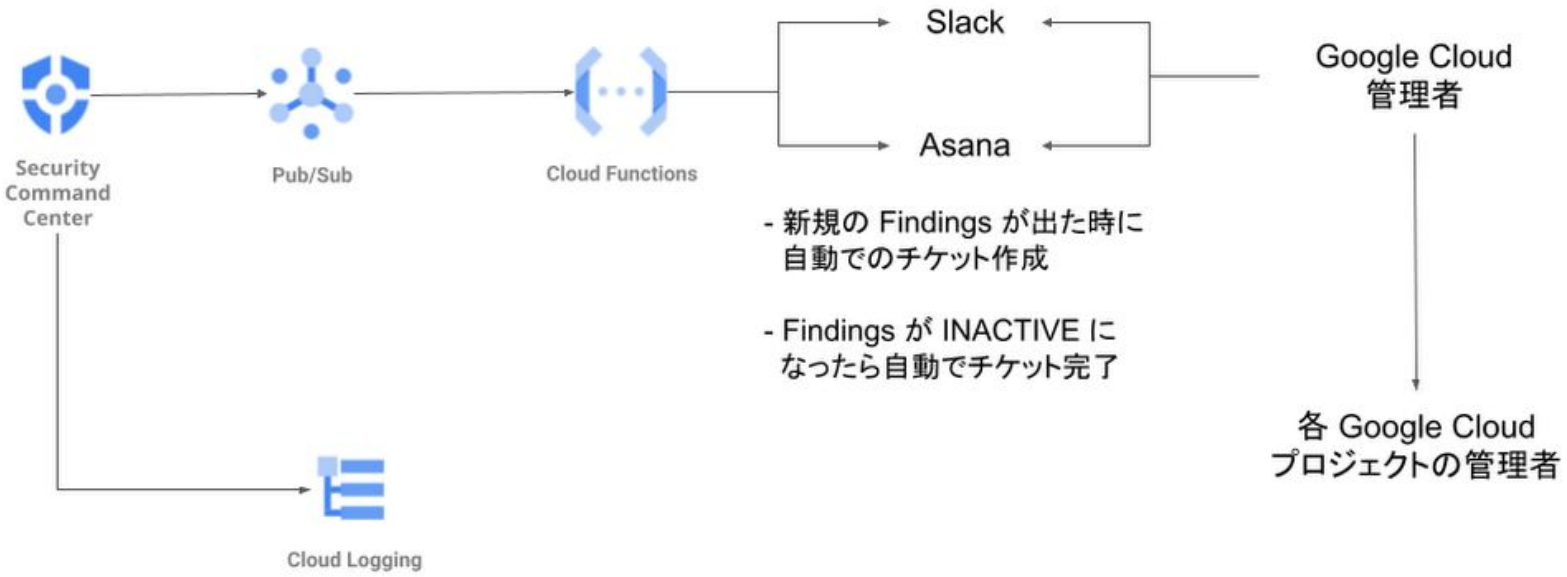
カンバンで対応状況を管理

- チケットは通知と同時に自動起票
  - 対応方法へのリンクなどを記載
- 重大なものは即時対応
- その他は週次定例で
  - チケットの仕分けとアサイン
  - ダッシュボードの確認
- 対応完了時は Security Command Center からステータスが連携され、自 動で完了レーンに移動する

[カンバンのイメージ]



通知・起票システムのアーキテクチャ



引用元: https://cloud.google.com/blog/ja/topics/customers/classi-strengthen-security

この運用を回してみてわかったこと

- Pros
  - 提案される対応方法をベースにモブプロすることでメンバー内に知見が溜まった
  - Security Command Center のグラフとチケットの残り具合で対応の進捗が可視化された
  - Security Command Center で出来ることの範囲がわかってきた

- Cons
  - 起票されるチケットが多くて対応しきれない
  - 同じ原因のチケットをそれぞれ個別で対応することが多い
  - リソースの管理者が不明確な場合があり、誰が対応するかを決めにくい
  - Security Command Center に対応することで「何がどこまで対策できている」のかがわからない

→ 最初の運用方針の狙い通りに行って、運用として次のフェーズに進めたが、まだまだ改善点は多い

Security Command Center で出来ないこともある

- 出来ること
  - 既存リソースの把握
  - 診断・検知
  - 対応方法の提案
- 出来ないこと
  - 脆弱性・セキュリティ脅威の発生予防
  - 検知結果の対応・仕組み化
  - 操作の追跡 (ログ)
  - 社内のポリシー・運用指針
    - 策定
    - 守れているかの確認
  - etc...

出来ないことに対して、Google Cloud の他機能や他サービスも組み合わせる必要がある

運用の改善

ここまでは「Security Command Center で出来ること」にスコープを絞っていたが、
運用の Cons と Security Command Center が出来ないことに対する対策に取り組んだ

1. セキュリティ ガードレール
2. 検知結果の対応の方針決め
3. ログ
    a. 収集
    b. 保管・活用
4. 社内ポリシー
    a. 策定
    b. 社内ポリシーを守る仕組み

※以前から対応済みのものや現在対応中のもの含む

対策1: セキュリティ ガードレール

組織ポリシーにより、組織全体におけるリソース作成を制限、セキュリティ脅威を事前に防ぐ

ポリシー例:

- デフォルト ネットワークの作成をスキップ
  constraints/compute.skipDefaultNetworkCreation
- デフォルト サービス アカウントに対する IAM ロールの自動付与の無効化
  constraints/iam.automaticIamGrantsForDefaultServiceAccounts
- バケットの公開防止
  constraints/storage.publicAccessPrevention

Google Cloud でのセキュリティ ガードレールについてもう少し詳しく知りたい場合のおすすめ：

GCPでセキュリティガードレールを作るための方法と推しテク

対策2: 検知結果への対応方法の改善

Security Command Center の検知結果に対して、より確実に対応する

- 実際に誰が何をいつまでに対応するか
  - Google Cloud Project 一覧と管理者 (Owner)の明確化
  - 管理者への Security Command Center 検知項目に対する対応依頼
- 対応しないものを決める
  - **全部を最初から対応するのは現実的ではない**
  - (今は) 対応しないと決めたものをシステム的に除外する
    - Mute rules
      - query ベースで検知を無効にする条件を設定できる

対策3-a: ログの収集

利用状況をみたい時や「何か」があった時のためにログは必要
デフォルトでは無効になっていて、自分たちで設定をしないと収集されないログがあるので注意
基本的には Event Threat Detection のための[ログの設定]をすれば十分
例:

- Flow Logs
  - VPC Network のログでSubnet ごとに有効にする必要がある
  - sample rateの設定がある
- [Audit Logs]
  - Data Access Log
    - 各サービスのデータへアクセス (Read/Write) した履歴
    - デフォルトでは BigQuery のみで有効になっている
- 各インスタンスのログ
  - GKE などではデフォルトで入っているが、Google Compute Engine ではエージェントを入れる必要がある
- etc...

対策3-b: ログの保管・活用

集めた上で、信頼できて使いやすい状態になっている必要がある
用途によって保管場所を分ける

- ログの集約
  - Project ごとでなく、Organization 全体で1箇所に集める
  - Log Router を使って 監査用の Project に集約する
- ログの保管場所
  - 直近のログ
    - 使いやすい場所に保管
      - Log Bucket, BigQuery
  - 長期間のログ
    - 安くて、誰も編集・削除できない場所に保管する (完全性)
      - Google Cloud Storage にバケットロックをかけた上で、保管する

対策4-a: 社内ポリシーの策定

基本は Security Command Center とベストプラクティスに従いつつも、社内独自の部分もあるので、自 分たちで決める必要がある
社内ポリシーによって、開発者全体の基準・行動指針を統一し、属人化を排除する

例:

- Organization 管理ポリシー
  - Organization 管理者がどういう基準で何をどこまでを見ているか
- Project 管理ポリシー
  - 各 Project を利用する人に何をどこまで気にして欲しいか
  - 禁止していることは何か
- データ保護・利用ポリシー
  - データの保管場所・方法をどうしてほしいか

対策4-b: 社内ポリシーを守る仕組み

社内ポリシーに違反したものがないかシステム的にコントロールする

- サービスアカウント キーの自動削除
  - サービスアカウントキーを仕方なく使う時は、一定期間で自動削除、ローテートを実施
    - ※そもそもキーを使わないのがベストプラクティス
    - 社内でも Workload identity federation が使えるときは使っている
- Asset Inventory から社内ポリシーに違反しているリソースを検知
  - 国外に保存されたデータがないかなど
- 特定の操作発生時に Slack 通知
- etc...

Organization でのクラウドセキュリティの運用振り返り

- Security Command Center 導入前の状態
  - 部分部分のベストプラクティスは知ってはいるが、実際の運用は個人任せで管理も把握も出来ていない
  - Organization として何から手を付ければいいかわからない状態
- 現状
  - Security Command Center を中心にしつつ、足りない部分を自分たちで補っている
  - 現状の改善点もある程度見えていて、計画に落とし込めている
  - 既存の取り組みを拡充をしていくことで、属人化を排除しつつ、クラウドセキュリティ運用のレベルを上げていける状態
- 振り返り
  - どう進めていくか悩んでいたフェーズから、Security Command Center ドリブンで進めて行くことで、課題や進め方が明確になり、着実に運用のレベルを上げていくことができた

おわりに

Security Command Center を上手く使うためのまとめ

- 影響が小さいうちに導入して試行錯誤していく
- 検出項目に対して、やらないことを決める
- Security Command Center が出来ないことを把握して、仕組みとシステムでカバーする

伝えたいこと

- 「Security Command Center があれば全てが安心」というわけではないけれど、心強い味方になる

- Security Command Centerドリブンに進めることができるので、クラウドセキュリティをどうすればいいかわからない人こそ
「まず Security Command Center を試してみる」がオススメ

Google Cloud に今後期待すること

- Security Command Center
  - 検知後も含めて本機能で完結させるための機能拡張
  - 「この検知項目に対応したいけど、この機能だとサポートされてない…」というケース削減
  - Data Loss Prevention と組み合わせたデータ保護
- 組織ポリシーの充実
- 運用事例の拡充

Security Command Center からクラウドセキュリティ運用を始めよう

Thank you.