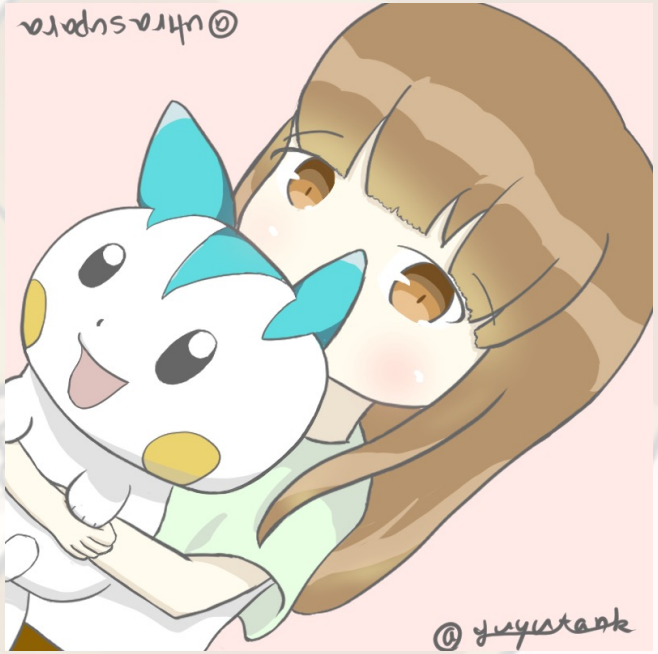
とあるユーザー企業におけるリスクベースで考えるセキュリティ業務のお話し

@4su_para

自己紹介　asu_para

- 2020年入学組
- RiST OB（-2022夏まで在籍）
- その後エンジニアインターンとしてキャリアスタート
- Security Engineer（24卒）
- PSIRTとして脆弱性診断の内製化や各種サービスのクラウドのアラート調査などに従事，一部コーポレートIT
- キーワード：#Threat Hunting #Cloud Security
- キャリアとして「Cloud Security」を軸に名を挙げて行きたいと考えています
- seccamp（21全国修了生，23チューター）
- SecHack365（CIのSASTツールの開発）



一般公開用・RiSTとは　@realRiST

- 立命館大学・情報理工学部プロジェクト団体
- 主にセキュリティ技術全般に関する個人・団体活動を、本学部セキュリティ・ネットワークコースと連携
- CTFを中心としたコンテストで良い実績を出すことを主眼に置いている
- セキュリティに関したLT会や研究・講習会などで親睦を深める
- 社会で多方面で活躍しているOBを多数輩出しています
- 顧問：毛利 公一 （セキュリティ・ネットワークコース教授）
- https://risec.github.io/

本イベントの始まり

とあるDMのやり取りが始まりまして・・・

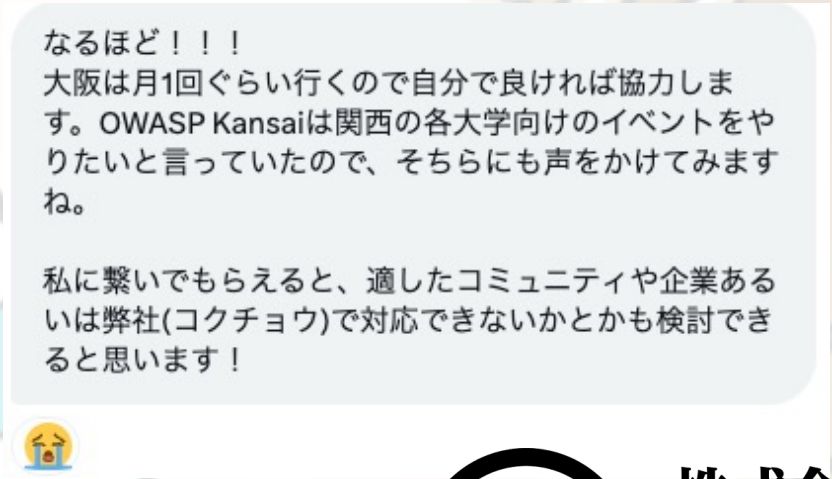
RiST運営目線で言うと、自分も含めて「セキュリティエンジニア」と呼ばれる分野でのキャリアパスが見えないor見えてない人がかなり多く、M進した人もあまりいないので在野のエンジニアのキャリアについて聞ける機会があればとは思っています.
セキュリティで仕事をすることに対しての実イメージがよくわかっていなかったり... SWEなら開発というイメージがありますがSIRTというのがよくわかっていない人も多いと思います. Threat researchなどセキュリティに関する仕事に何があるのか、など...
asu_paraさんなど現職のセキュリティエンジニアと交流できる機会というのもあまりないのでRiST向けにお話聞きたいと思っているのですが可能でしょうか?

さっそくなんですが、今回のお話について、具体的にお伺いしたいと思います。
ぜひ一度、当団体のメンバーに向けて登壇いただく機会をいただけないかと考えております。
「セキュリティエンジニア」としての働き方やキャリア、技術的なトピックについてお話を伺えればと思うのですが、いかがでしょうか？

/darallium/ · Fri 1:45 PM

ありがたいけど、自分だけだと心もとないかな・・・

シニア層にも声をかけてみた



株式会社
コクチョウ

ありがとうございます。
助かりました。。。

「強さ」への憧憬

最近見たり聞いたりしたRiSTの活動や実績

- 任意のCTFで国際CTFのFinalsに行けるような順位になった
- CTFで2位，3位，14位・・・
- CTF以外のセキュリティコンテストでも決勝に勝ち進んだ
- seccampに5人(/年)参加者を輩出した
- スポンサーを味方につけてうまくやっていそう
- 部員が50人超・・・（アクティブは15人くらい？）

それでも人は周りと比較し悩み始める・・・

- CTFサークルなので、CTFで言えばBun●yo ●esternsのような強いチームの存在とか・・・
- エグい量、かつインパクトの脆弱性報告をしているあの人
- 技術書や同人誌で有名なあの人
- バグバウンティで実績を残しているあの人
- 自作のツールやOSSで有名なあの人

この「強さ」は事業組織での「強さ」とは、ややベクトルが異なる気がする・・・

実際のプロダクトや組織で求められる「強さ」

- 相変わらず「手を動かし続ける」ことの大切さは変わらない
- 「リスク」や「事業」を正しく理解する
- 「リスク」を正しく捉えて優先順位をつけて取り組む
  - やらないことを決断する（Money Forward CISO）
- 開発組織のメンバーに専門性や人柄を信頼されて協力して取り組む
- そのための「人に伝える」コミュニケーション能力
- 技術・攻撃のトレンドを追って思考する
- 現状のステークホルダーを理解する
- **他にもいろいろ・・・自分自身できてないことだらけですが・・・**

https://recruit.moneyforward.com/times_mf/article/interview0010

なぜ、ユーザー企業に就職する道を選んだのか

- ベンダーだけで根本的なセキュリティの改善が行えるとは思っていない
  - インシデントが発生した際に「次に同様な問題を起こさないようにはどうするか」という組織的な改善・振り返りにまでコミットできる
  - そこまでベンダーは踏み込むことができない
- **クラウドの分野でやっていく**意志とのマッチング
  - クラウドセキュリティだけに関わらず、さまざまなトピックでUser Groupの方がベンダーよりも知見がある
  - クラウドはリスク受容の観点でリソースの「使われ方」に着目することが多い
  - 160以上のクラウドリソースを多様している（所属先）
  - ユーザーグループにおけるクラウドの先進的な活用はアジリティとセキュリティの両立を追求しやすい・そこに面白さの余地がたくさんある
- **ユーザーから盛り上げていきたい**

https://fit.nikkin.co.jp/post/detail/fj0048

リスクを正しく捉えるって何？

おさらい：情報セキュリティの概念

情報セキュリティとはJIS Q 27000:2019によると「**情報の機密性、完全性および可用性を維持すること**」と定義され、情報は有効に利用されてこそ価値を生む。

機密性

認可されていない個人エンティティまたはプロセスに対して、情報を使用せず開示しない -> **情報へ認可されたもののみがアクセスできる**こと

完全性

正確さや完全さの特性 -> **情報が矛盾や改ざんがなく正確であること**

可用性

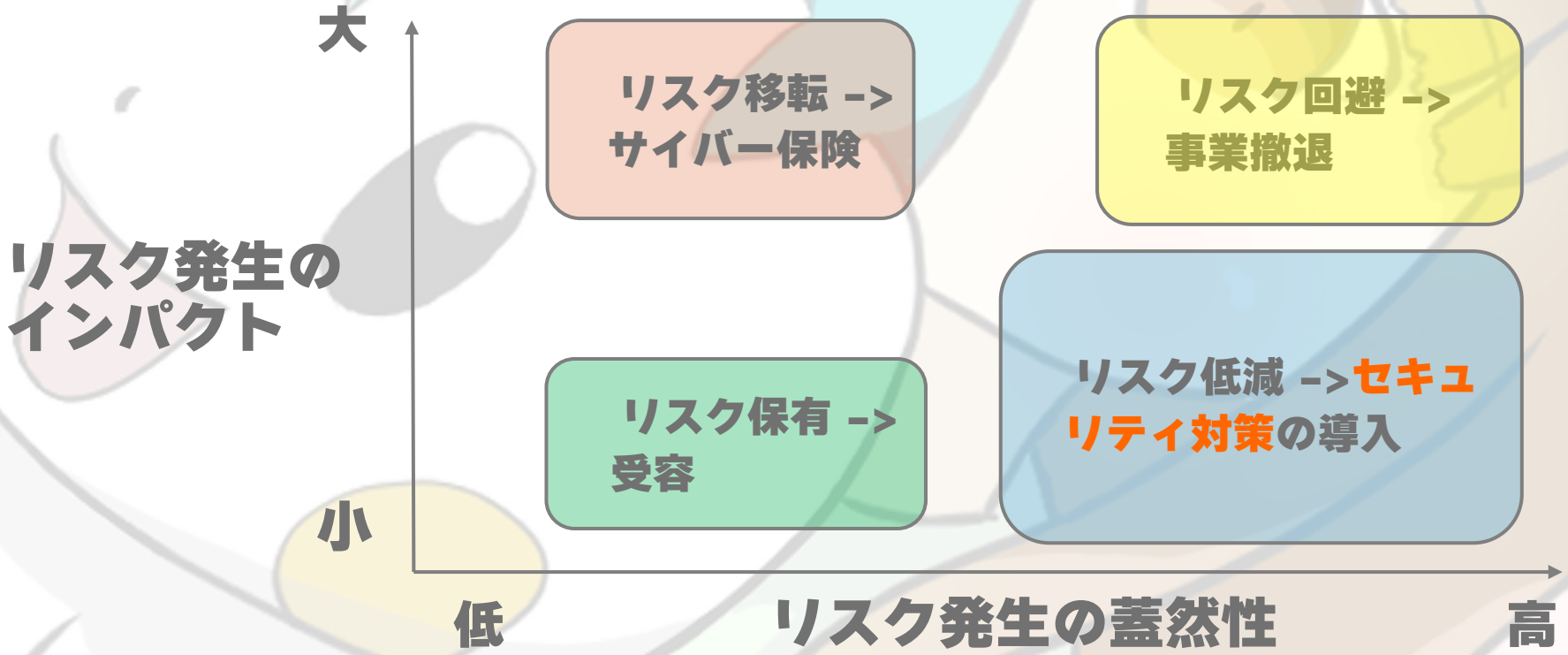
認可されたエンティティが要求したときに、アクセス及び使用が可能である特性 -> **情報へ必要なときにアクセスできること**

リスクの考え方

正確さや完全さの特性 -> 情報が矛盾や改ざんがなく正確であること

リスクへの対応

リスクへの対応としては4種類ある。リスクのインパクトと発生確率によって適した対応をとっていくことが企業の方針として重要である。



最終的には「残存リスク」を定量的に捉えること

FISC（「金融機関等」コンピューターシステムの安全対策基準・解説書」第12版）

TLP:RED

情報資産の整理と脅威の特定

情報資産の整理

TLP:RED

脅威の特定

- ○脅威の種類
  - □なりすまし
  - □改ざん
  - □情報漏洩
  - □サービスの停止
- ○攻撃者
  - □外部/内部/自然災害
- ○動機
  - □意図しない操作ミス
  - □金銭の取得
  - □社内活動の妨害

業務では、不要になった情報資産の整理やクラウドリソースの調査といったことも行なっています^^

脆弱性の分析

システムにおける対象の切り分けと対策

TLP:RED

セキュリティ診断の補完

オペレーションの自動化

そもそもなぜ、補完が必要か

- 「ある時点」での安全性の担保にしかなっていない
- 自動化されていないので再現性の担保に乏しい
- 値段が高いわりに成果が出ているかはわからない
- プロダクトや事業コンテキストにおける優先をわかっている人が診断をしているわけではない
  - ベンダーはユーザーの気持ち(真のペイン)がわからない
  - そもそも自分たちのシステムは自分たちの責任で守っていく必要がある。

その前に・・・
発見的統制と予防的統制

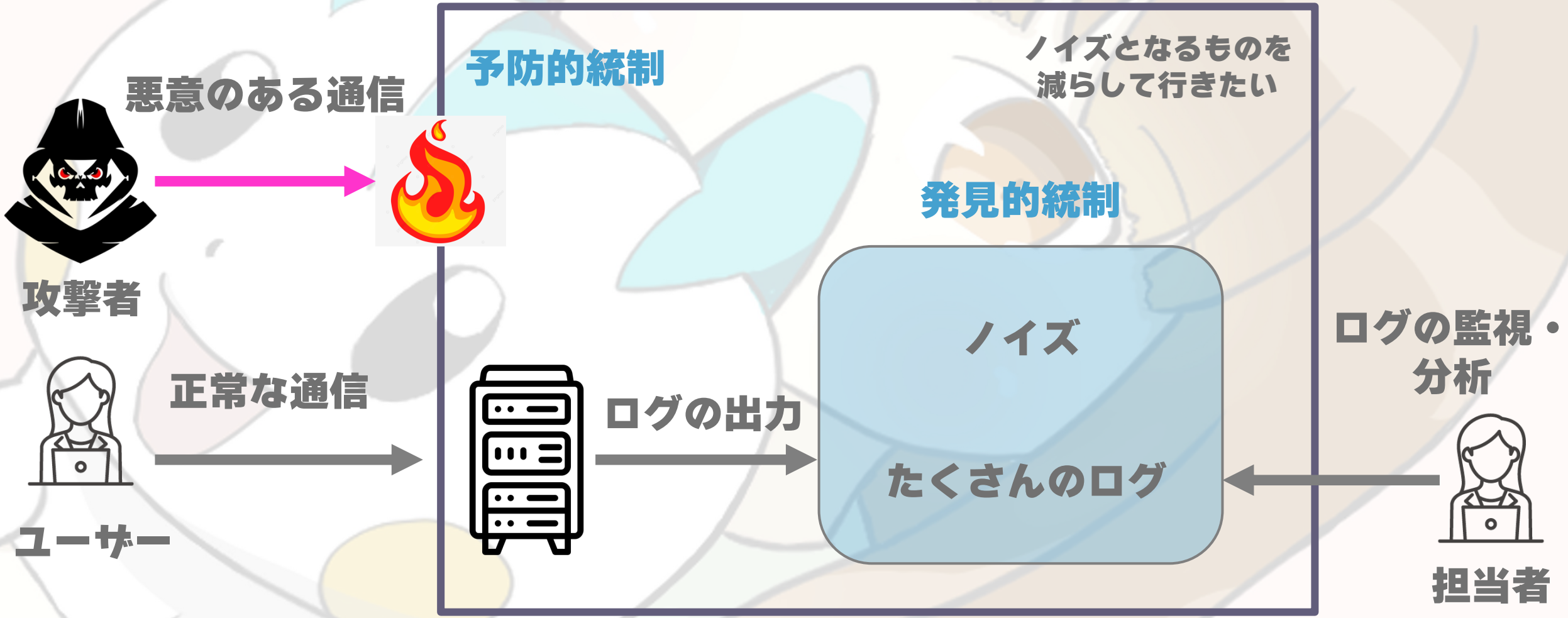
発見的統制

- ○事後に顕在化したリスクをいち早く発見し、適切に是正するための対策，発見から予防に繋げていく（以下はこの後話す項目）
  - □secret scanning
  - □Depandancy Managemant
  - □Threat hunting
  - □クラウドで言えば・・・
    - GuardDutyなどネットワークベースの検知ソリューション
    - Security HubなどのCSPM，ガイドラインを用いたベースラインアプローチ

予防的統制

- ○事前に潜在的なリスクが起きないよう、発生そのものを抑止・防止する対策（以下はこの後話す項目）
- ○発見的統制よりもこっちの方がよっぽど重要。そもそも被害を受けないようにするためのアクション。
  - □secret scanning
  - □クラウドで言えば・・・
    - IAMの保護
    - Security Groupによる不必要な通信の遮断
    - KMSによる暗号化のための鍵の生成・管理・制御

発見的統制と予防的統制の組み合わせイメージ



SASTによる解析をGitHub Organization全体へ

- ソースコード類に対して機械的な分析を実行して脆弱性やその温床となりうる箇所を洗い出す
- 対象のコードの種類に応じてツールを選定したり不足分は自作するなどしてgithub actionsで実行する
  - Goだったらgolangci lint
  - SQLだったらsqlfluff
- Shift left
  - 対策の方法論の1つ
  - 動かす前になるべく問題を見つける

```yaml
run:
  deadline: 3m
  issues-exit-code: 1
  tests: false

output:
  formats:
    - format: checkstyle
      path: golangci-lint.xml
  print-issued-lines: true
  print-linter-name: true

linters:
  disable-all: true
  enable:
    - gofmt # TODO remove
    # デフォルトのGoのLint
    - govet
    # 型情報のチェック
    - typecheck
    # チェックされてないエラーの検出
    - errcheck
    # strictな静的解析
    - staticcheck
    # 使用されていない定数、変数、関数、Type を検出
    - unused
    # セキュリティ脆弱性の静的解析
    - gosec
    # 意味のない代入の検出
    - ineffassign
    # 繰り返し処理の中で定数化できるものを検出
    # - goconst
    # net/httpのレスポンスボディがCloseされてないことを検出
    - bodyclose
    # Go1.13以降のエラーのWrapをしてないままreturnしているケースを検出
    # - errorlint
    # sql.Rowsやsql.StmtがCloseされてないのを検出
    - sqlclosecheck
    # http.Requestがcontextなしで送られていて、タイムアウトハンドリングなどができないのを検出
    - noctx
    # sql.Rowsのエラーがハンドリングされていないケースを検出
    - rowserrcheck
    # Non-ASCII characterを検出
    #- asciicheck
    # enumの値でハンドリングされていないものを検出
    #- exhaustive
```

クラウドでインシデントが起こりうるケース

- 1. 設定ミスが原因で発生する情報漏洩
  - 誤公開してしまったS3のリソースに機微情報が埋まっていて〜・・・
  - 誤公開でEC2インスタンスのポートがフルオープン(0.0.0.0/0)になっていて〜（vs CSPM, ASM）
- 2. 何らかの原因でアクセスキーが漏洩
  - 場合によっては一気にシステム全体が掌握される可能性もありうるので非常に危険（vs secret scanning）
  - SSRFでメタデータサーバー経由に引っこ抜かれることもあるかも
- 3. デプロイされたアプリケーションの侵害からクラウド側が侵害される
  - 侵害したサーバーを踏み台にしてネットワーク内に攻撃を水平展開していくイメージ（vs web脆弱性診断, クラウドPlatform診断）

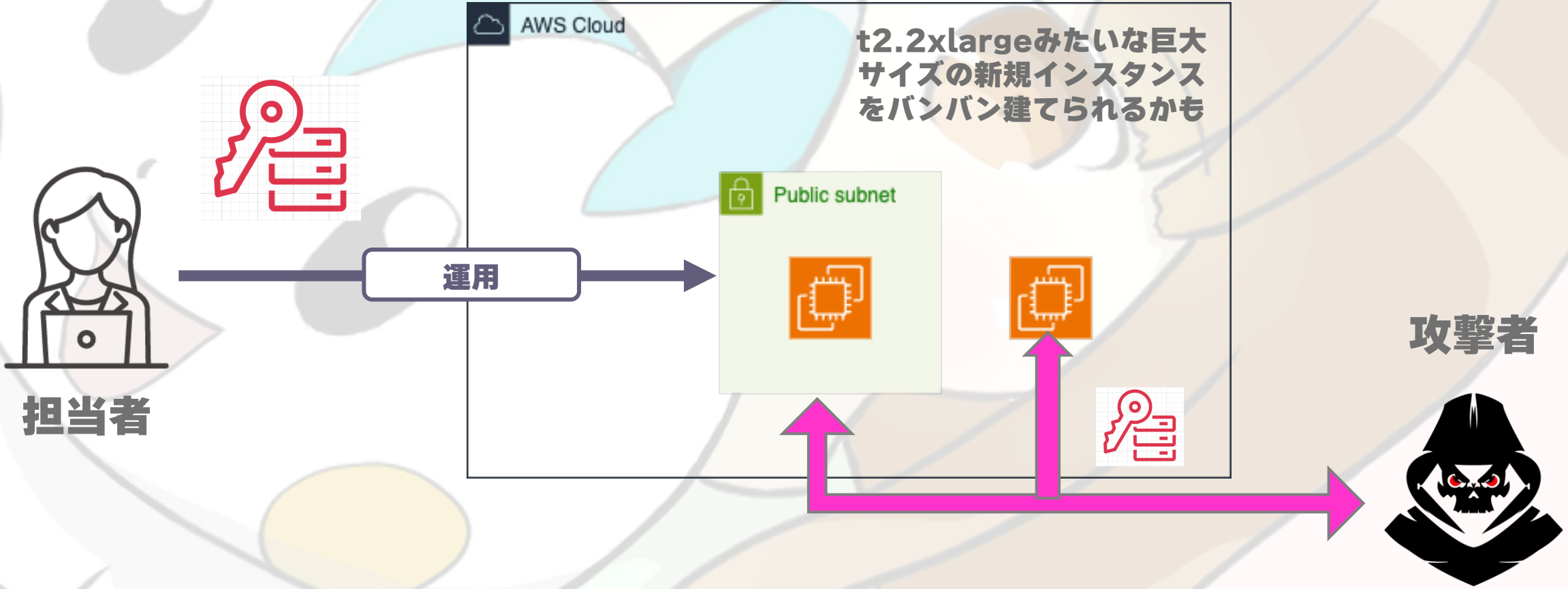
Secret Scanning（SAST+検知ロジック）

ペネトレーション対策としてミスオペレーションや侵害があった場合にすぐに発見して適切に対処できる仕組みを内製している

TLP:RED

キーが漏れて悪用されたらどうなる？？

アカウントの2FAやってないとか、IAMに不用意に強い権限がついているとか、場合によってはシステム全体が一気に掌握されてしまう可能性もある



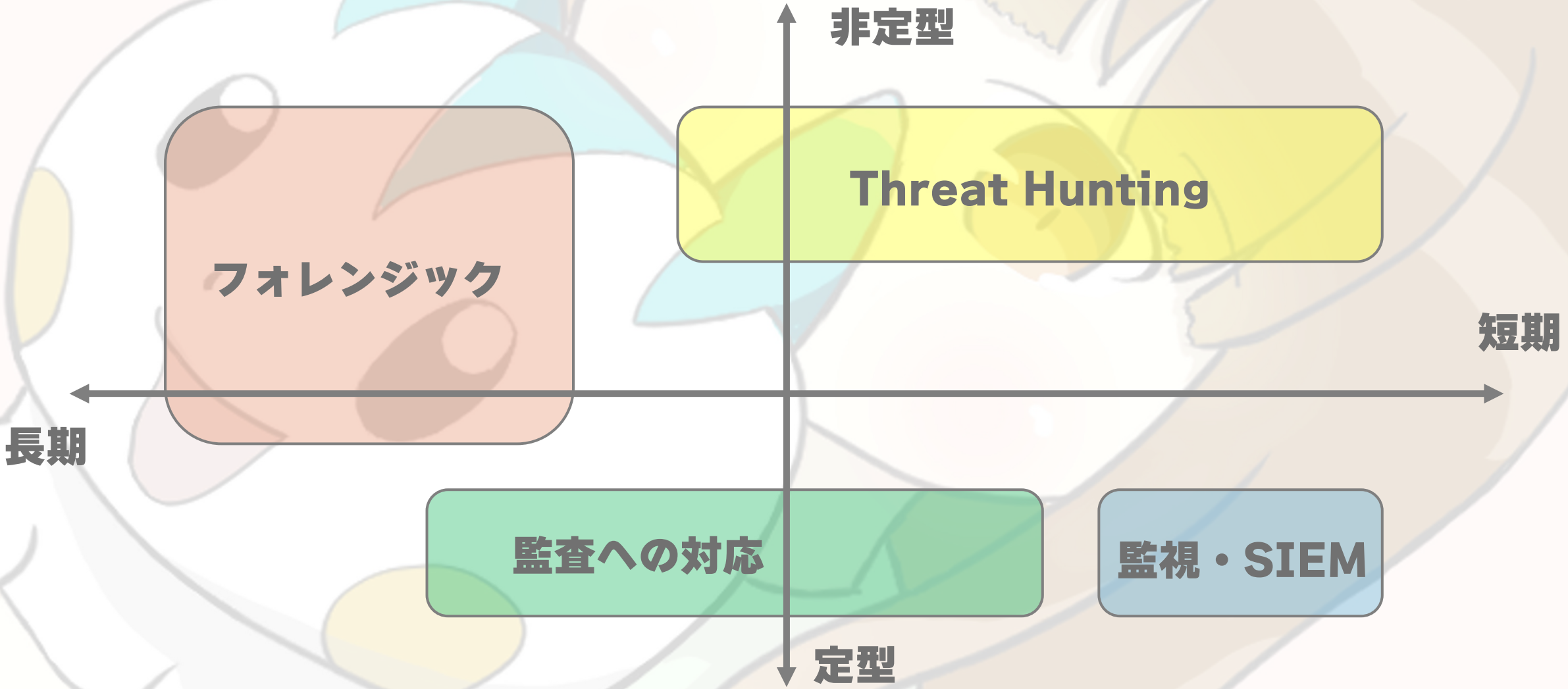
とにかくログを取るのは

いざという時のため！

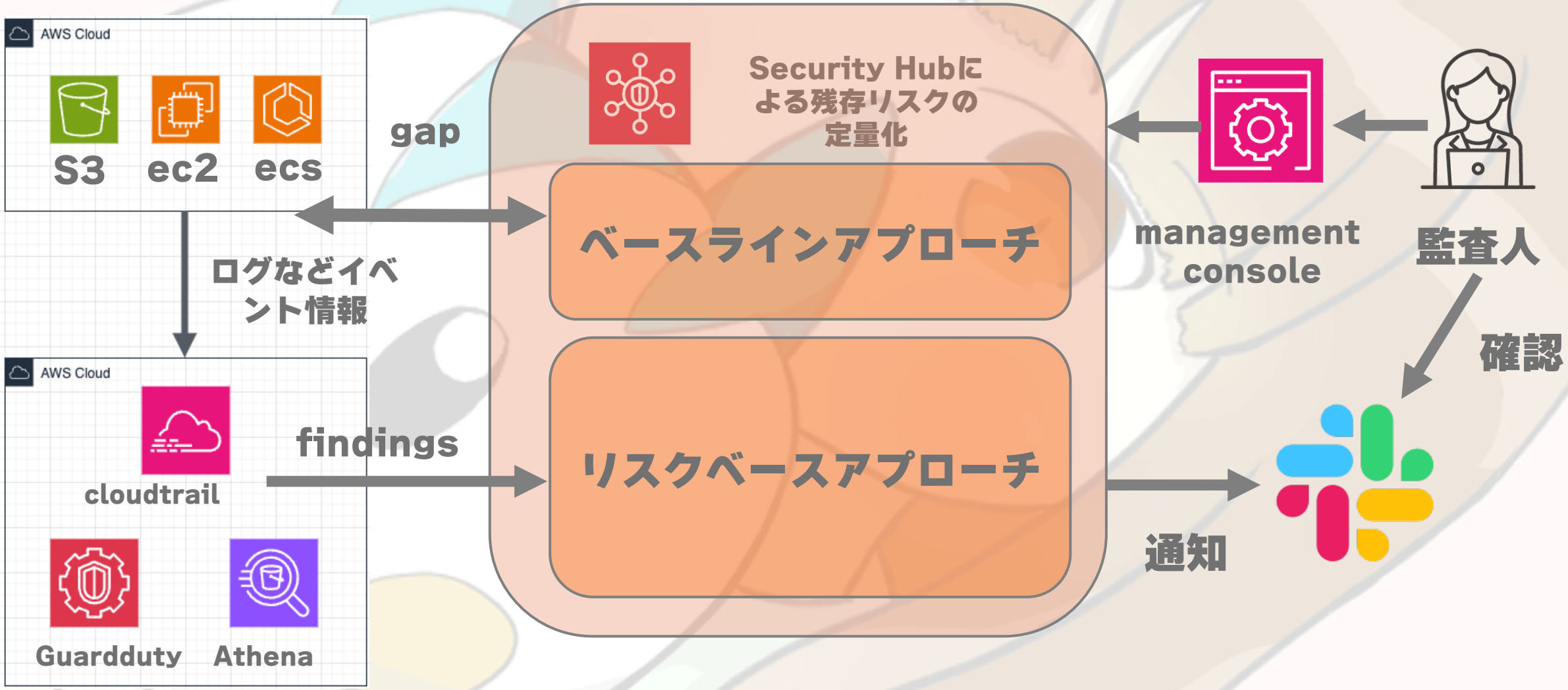
ログについてのあれこれ

- ○システム操作のトレーサビリティを確保するため
- ○ログをたくさん集める（だけではダメで・・・）
  - □ログに対する権限を正しく管理する（機密性の確保）
    - 適切な場所に集約しておく
  - □それらのログが改ざんされないようにすることを担保する（完全性の確保）
  - □それらのログを容易に分析できる状態にしておく必要がある（可用性の確保）
    - SIEMに流し込んでsigma?yara?
    - S3 + AthenaでSQLクエリを書いていく？
    - ログのモニターを内製化+slack通知?

ログを駆使した業務イメージの分類

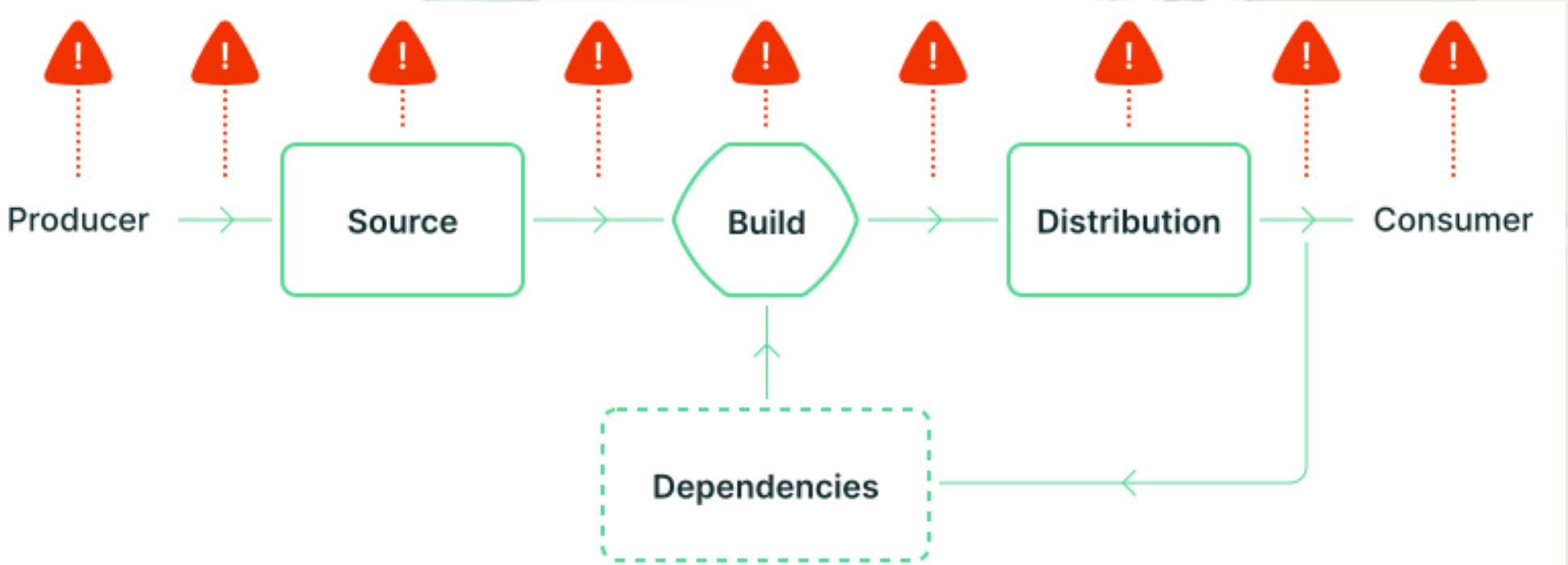


CSPM（Cloud Security Posture Management）



Dependency Management

Software Supply Chain Securityと呼ばれる分野です



- ビルドプロセスのどこかに悪用可能な脆弱性のある依存ソフトウェアのパッケージが紛れ込んでいるかも・・・
- updateされてないままになっているものは必ず存在する

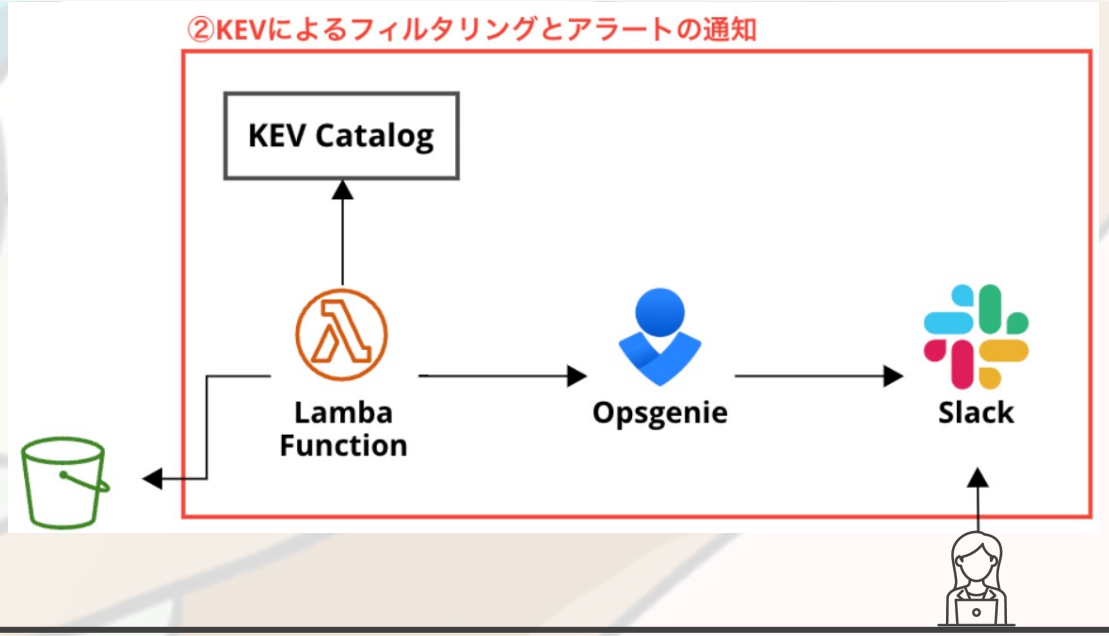
https://slsa.dev

Dependency Management

- 大量に上がってくるGitHubのDependabotとかECR(コンテナ)のimage scanningで取得したものから実際に悪用可能なものに絞り込んで通知をする仕組みを作っている
- アラート疲れを減らしつつ本当に対応に緊急性があるものに着手しやすくする（リスクに）

```
const kevJsonURL = "https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json"

type kevVulnerability struct {
    CveID             string `json:"cveID"`
    VendorProject     string `json:"vendorProject"`
    Product           string `json:"product"`
    VulnerabilityName string `json:"vulnerabilityName"`
    DateAdded         string `json:"dateAdded"`
    ShortDescription  string `json:"shortDescription"`
    RequiredAction    string `json:"requiredAction"`
    DueDate           string `json:"dueDate"`
    Notes             string `json:"notes"`
}
```



コーポレートシステムも含め様々な取り組みを行う

今日は話しきれなかった項目たちがたくさん・・・

- ○バグバウンティプログラムへの出展
- ○Attack Surface Managemantの内製化に向けた取り組み
- ○パスワードマネージャーの多用
- ○IdPによる特定SaaSのログイン経路の確保& password lessでのログイン
- ○JamfやIntuneなどクラウドによる端末管理&EDRを配布して

などなど・・・

まとめ

まとめ

- CTFサークルとして技術を極めていくところとユーザー企業で求められることのギャップの確認
- ユーザー企業の取り組みは**幅が広くて面白い**
- より**開発サイドに近い知見とか考え方**が必要になってくる
  - 開発におけるアジリティとセキュリティの両立
  - ホワイトボックスにおける診断
  - クラウドのインフラであればリソースの使われ方
- 攻撃のことを知って防御に活かしていく必要がある