Webサービス事業会社における EDRの検討と導入の事例

クックパッド株式会社
三戸健一、水谷正慶

cookpad

本日のトピック

1. EDR導入のきっかけ
2. PoCの評価項目と評価結果
3. CrowdStrike Falconの導入
4. CrowdStrike Falconの運用
5. 今後の展望＆製品への期待

自己紹介

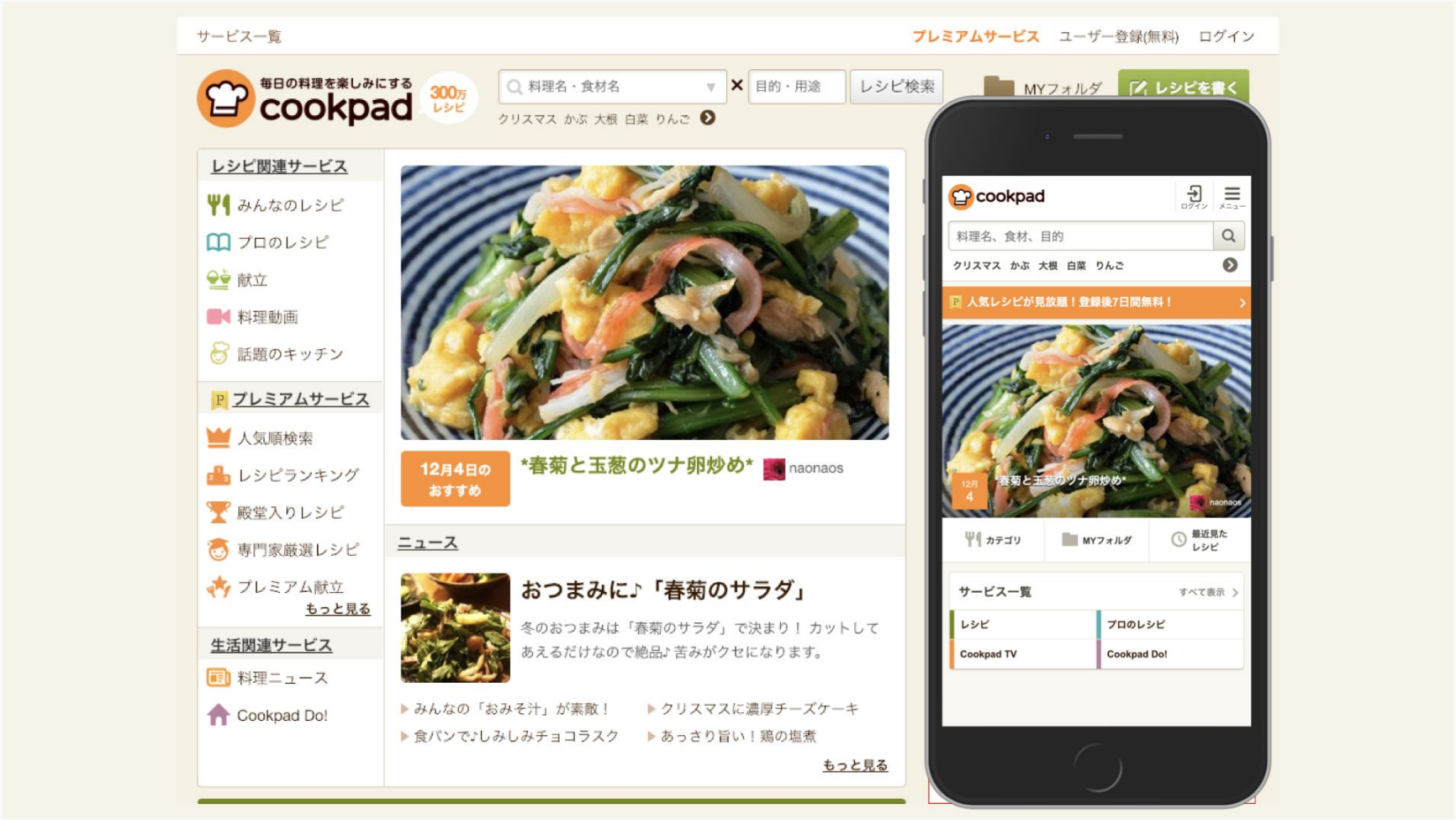
インフラストラクチャー部 セキュリティグループ

三戸健一

- 2014年6月入社
- サービスの脆弱性診断やISMS審査の対応などを担当
- 元は社内IT部門から異動

水谷正慶

- 2017年11月入社
- 社内のセキュリティ監視の仕組みの設計と実装を担当
- 前職ではSIEM構築・SOCアナリスト・研究開発などを経験


サービス一覧
プレミアムサービス
ユーザー登録(無料)
ログイン
毎日の料理を楽しみにする
cookpad
300万
レシピ
料理名・食材名
目的・用途
レシピ検索
MYフォルダ
レシピを書く
クリスマス かぶ 大根 白菜 りんご
レシピ関連サービス
みんなのレシピ
プロのレシピ
献立
料理動画
話題のキッチン
プレミアムサービス
人気順検索
レシピランキング
殿堂入りレシピ
専門家厳選レシピ
プレミアム献立
もっと見る
生活関連サービス
料理ニュース
Cookpad Do!
12月4日の
おすすめ
春菊と玉葱のツナ卵炒め
naonaos
ニュース
おつまみに♪「春菊のサラダ」
冬のおつまみは「春菊のサラダ」で決まり！ カットして
あえるだけなので絶品♪ 苦みがクセになります。
みんなの「おみそ汁」が素敵！
クリスマスに濃厚チーズケーキ
食パンで♪しみしみチョコラスク
あっさり旨い！鶏の塩煮
もっと見る
cookpad
ログイン
メニュー
料理名、食材、目的
クリスマス かぶ 大根 白菜 りんご
人気レシピが見放題！登録後7日間無料！
12月
4
春菊と玉葱のツナ卵炒め
naonaos
カテゴリ
MYフォルダ
最近見た
レシピ
サービス一覧
すべて表示
レシピ
プロのレシピ
Cookpad TV
Cookpad Do!


国内展開
● レシピ数 約300万品
● 月間利用者数 約5,400万人


海外展開
● 対応言語 23言語68カ国
● 月間利用者数 約3,800万人

EDR導入のきっかけ

cookpad

社内環境

- MacBook (macOS): 500台
- Surface Pro 4 / Laptop (Windows 10): 80台
- OSは原則として最新のバージョンを使用
- エンジニア・デザイナ・総合職に関わらず原則MacBookを支給
  - エンジニア・デザイナで全体の4割強
- WindowsはActive Directoryでアカウント管理
  - Azure AD Joinに移行中
- macOSはローカルユーザのみ(ディレクトリ管理していない)
- 恵比寿オフィスの社内ネットワークにはPaloAltoのNGFWを導入
  - ブラックリスト方式で運用

Falcon以前の製品選定

- Falcon以前に利用していたアンチウイルスの選定基準
  - macOS / Windows両対応
    - WindowsよりもmacOSでの機能面
    - 特に開発環境でのオーバーヘッドが低いことを重視
  - オンプレミスの管理サーバを必要としないこと
- APIやログの保存などは重視していなかった
  - 製品から提供されるコンソールをそのまま使用
  - 定期的にデータをエクスポート(csv)し、手作業で集計

Falcon以前の導入製品における課題 (1/2)

1. 開発環境を含め、社内環境はほぼmacOSに揃えている
    - Windowsに比べ、macOSのサポートが貧弱な製品が多い
    - macOSのアップグレードへの対応が遅かった
        - iOSアプリ開発などでOSのアップグレードが必要
    - iOS SimulatorなどXcode関連でfalse positiveが度々発生
        - 複数台から突然大量のアラートが発生することもあった

Falcon以前の導入製品における課題 (2/2)

2. アラートに含まれる情報が不足
    - ファイルシステム上の活動のみ（パスやファイルハッシュ）
    - 追跡調査できずにインシデントをクローズすることが度々
3. 通信イベントの監視を行いたい
    - ファイルに対する監視だけでは不十分
        - どのような経路で該当ファイルがドロップされたのか、外部との通信は存在したか、などの追跡を行いたい
    - NGFWなどの通信ログと突合したい
        - それまでは時系列で判断するしか無かった

PoCの評価項目と評価結果

cookpad

評価にあたって

- PoC実施前に、予め必要項目を提示
  - 最低限出来て欲しい機能と、これが出来ると嬉しい機能
  - 一覧を表にして共有
    - いわゆる星取り表という意図ではなく、PoCでどういった点を見るのか、どのような機能を重視しているのかを予め整理して明確化するための資料
- 社内環境の説明
  - 前述の選定要件など

No.	大項目	優先度	小項目	備考
1.1.1	エージェント	Should	MDM経由でインストールできるか(macOS)	端末のキッティングにJamf ProおよびIntuneを用いる予定のため (MDMに正式対応 or 最低でもコマンドラインに依るサイレントインストールが可能だと良い)
1.1.2	エージェント	Should	MDM経由でインストールできるか(Windows)	
1.2.2	エージェント	Should	プロセスが長期間ネットワーク接続していないことを 検知 or コンソールから確認できる	
1.3	エージェント	Should	プロセスが強制終了させられた場合に自動で復帰する	
1.4	エージェント	Must	負荷が十分に小さい	特にCPU、ファイルI/Oの高負荷時に影響がでないか。具体的にはコンパイルやファイル内検索のような作業においてオーバーヘッドが10%以下であることが目標（以前某プロダクトを使用していた際、git grepやAndroidのビルドが非常に遅くなり困った）
1.5	エージェント	Should	Macを移行アシスタントで端末を交換したときに問題が起きないか	ターゲットディスクモード、あるいはTimeMachineバックアップを用いて、故障したMacBookのユーザデータを復元してハードウェア交換を行うことがあるため
1.6	エージェント	Should	macOSの最新版対応までの時間が十分に短いか	
1.7	エージェント	Should	十分な解像度のイベントを取得できる	
2.1	コンソール	Must	Windows, Macについてのデータが透過的に閲覧できる	
2.2.1	コンソール	Should	プロセス名（＋時刻）をキーにイベント一覧を閲覧できる	
2.2.2	コンソール	Must	ファイルハッシュ値（＋時刻、ホスト名）をキーにイベント一覧を閲覧できる	
2.2.3	コンソール	Must	通信先IPアドレス（＋時刻、ホスト名）をキーにイベント一覧を閲覧できる	
2.2.4	コンソール	Must	通信先ドメイン名（＋時刻、ホスト名）をキーにイベント一覧を閲覧できる	
2.2.6	コンソール	Should	あるファイルに対して読み書きしたプロセスを追跡できる	
2.2.7	コンソール	Must	あるアラートにスコープをあわせて検索ができる	
2.3	コンソール	Must	イベントの検索条件をURLのQueryStringで指定できる	
2.4	コンソール	Should	任意のイベントについて視覚化できる	
2.5	コンソール	Should	視覚化のページに対してpermanent linkを生成できる	
2.6	コンソール	Must	アラートに対してpermanent linkを生成できる	
3.1	検知	Should	脅威の検知ができる	
3.2	検知	Must	ハッシュ値でホワイトリストを作成できる	
3.3	検知	Should	柔軟な条件でホワイトリストを作成できる	複数条件を組み合わせて指定できるか、など
3.4	検知	Must	誤検知しすぎない	ホワイトリストでの調整は必要だが、あまりにも数が多いと厳しい
4.1	API	Must	脅威検知情報（アラート）をAPIで取得できる	push型で通知されるのが望ましい
4.2	API	Must	ファイルハッシュ値（＋時刻、ホスト名）をキーにイベント一覧を取得できる	
4.3	API	Must	通信先IPアドレス（＋時刻、ホスト名）をキーにイベント一覧を取得できる	
4.4	API	Must	通信先ドメイン名（＋時刻、ホスト名）をキーにイベント一覧を取得できる	
4.5	API	Should	あるプロセスに対してcode injectしたプロセスをAPIから取得できる	プロセスをキーに検索できるか
4.6	API	Should	あるファイルに対して読み書きしたプロセスをAPIから取得できる	書き込み・読み込みされたファイルをキーに検索できるか
4.7	API	Must	APIの呼び出し回数制限は十分余裕がある	平均で1000 req/day 程度の見込み
5.1	ログ転送	Must	完全な状態のログを転送できる	コンソールに出ている情報と差分がないか
5.2	ログ転送	Must	暗号化してログ転送できるか	httpsなど標準的な暗号化方式
5.3	ログ転送	Must	ログを他の環境へ転送できる	ストリーム型とバッチ型のどちらかでもいいが、どちらか必須
5.4	ログ転送	Must	24時間以内にログが転送できるか	転送可能になるまでの時間は短いほど（実時間が理想）良いがあまりに長いと許容できない
5.5	ログ転送	Must	（グローバルIPアドレスから送られてくる場合）転送元IPアドレスを制限できる	ログ収集サーバはAWS上で動かしており、アクセス制限を行いたいため
6.1.1	対応	Should	Webコンソールから端末のローカルshellを操作できる (macOS)	
6.1.2	対応	Should	Webコンソールから端末のローカルshellを操作できる (Windows)	
6.2.1	対応	Should	shell操作時に情報の吸い上げが容易にできる (macOS)	
6.2.2	対応	Should	shell操作時に情報の吸い上げが容易にできる (Windows)	

評価項目と結果 (1/5)

1. 負荷が十分に小さい
    - git grepやJavaプロジェクトのビルドが遅くならないこと
        - 10秒程で終わるはずのコマンドが5分になってしまう
        - .classファイルを監視から外すなど、微調整したことも
    - 通信をフックして独自のローカルプロクシで監視する製品
        - 遅延の他に、特定条件下でHTTPパケットが壊れる
        - エージェントが暴走して通信不能になることも
2. 開発環境におけるfalse positiveが少ない
    - 開発の妨げになる・件数が多いと狼少年に

評価項目と結果 (2/5)

- 負荷・false positiveについて
  - エージェントを入れた端末に、実際に開発環境を一から構築
    - rubyビルドやライブラリの取得、iOS/Android開発環境
    - 正常に起動するか、また起動時間が長くならないか
    - 他の製品では環境構築自体ができないものも存在した
  - 候補がFalconにほぼ絞られてきた時点で、社内各部署のエンジニアの協力を得て、実際の開発環境にも導入し検証
    - false positiveも含め問題は起きなかった

評価項目と結果 (3/5)

3. イベントの検索機能
    - ファイルハッシュ、IPアドレス、通信先ドメイン名などをキーに横断的に検索ができるか
        - アラートに含まれる情報を用いて周辺イベントを調査
        - 検索結果に対してpermanent linkを作成できると嬉しい
4. APIによるイベントの取得
    - コンソールで出来ることがAPIで提供されているか
        - アラート情報もAPIで取得したい
        - 時間あたりの回数制限にも注意が必要

評価項目と結果 (4/5)

5. ログの転送
    - コンソールで得られる情報がそのままログとして外部から取得できるか
        - httpsなど標準的な暗号通信経路で取得したい
            - 保存先がクラウド環境なので、syslogは厳しい
    - ログはs3に保存して永続化
        - 監査にも対応できるよう年単位で保持しておきたい
    - ログ出力のタイミング
        - 転送可能になるまで時間がかからないものが望ましい

評価項目と結果 (5/5)

- イベントの検索・APIについて
  - コンソールの検索機能は問題なかった
  - APIから利用できる検索機能は希望を満たすものではなかったが、転送したログに対して自前で検索を行う方針に転換したため、APIによる検索を必要としなくなった
- ログの転送
  - Falcon側のs3バケットに保存されたログを、Data replicatorを用いることで任意のs3バケットにコピー可能(5分毎)

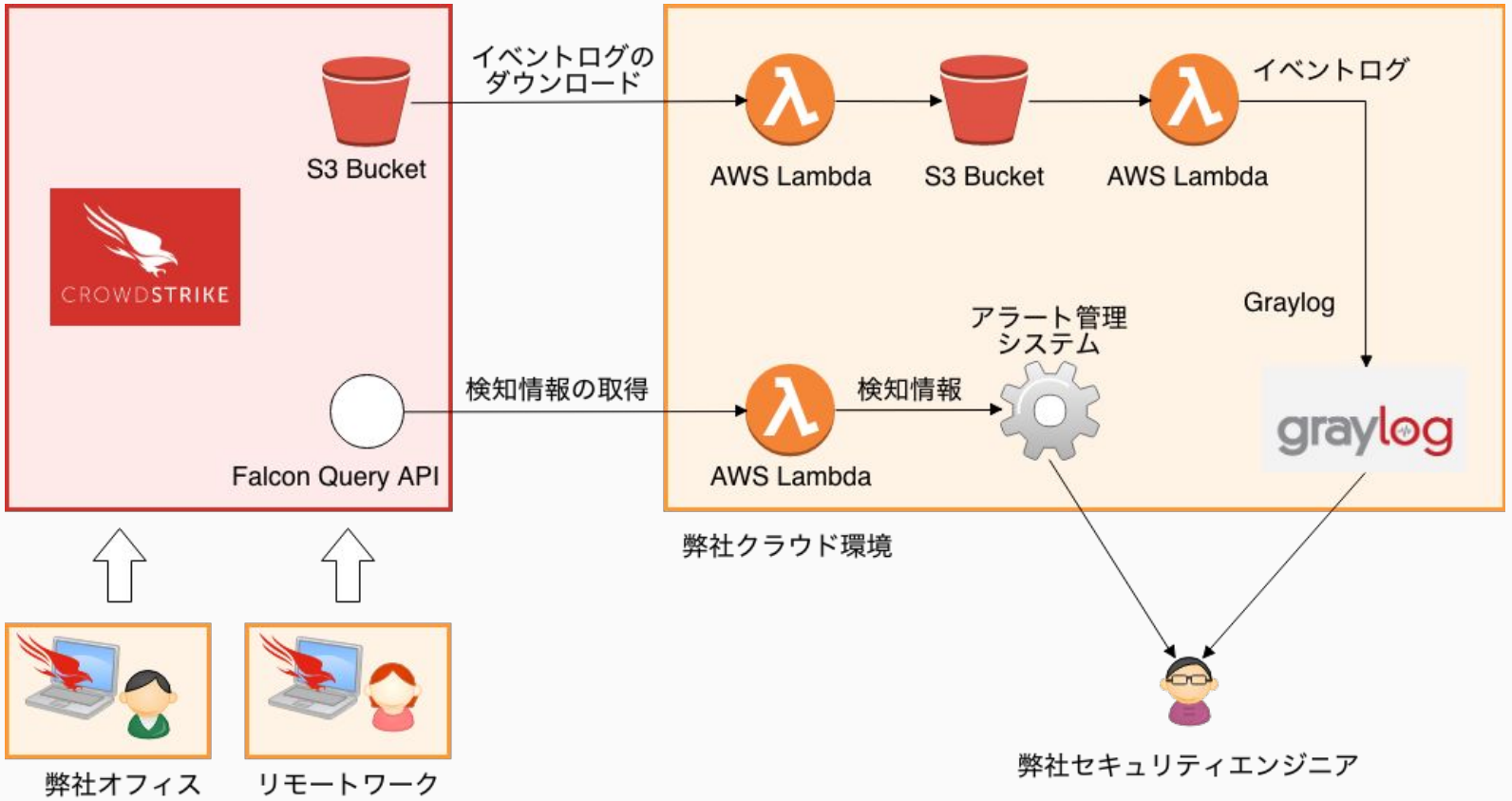
CrowdStrike Falconの導入

cookpad

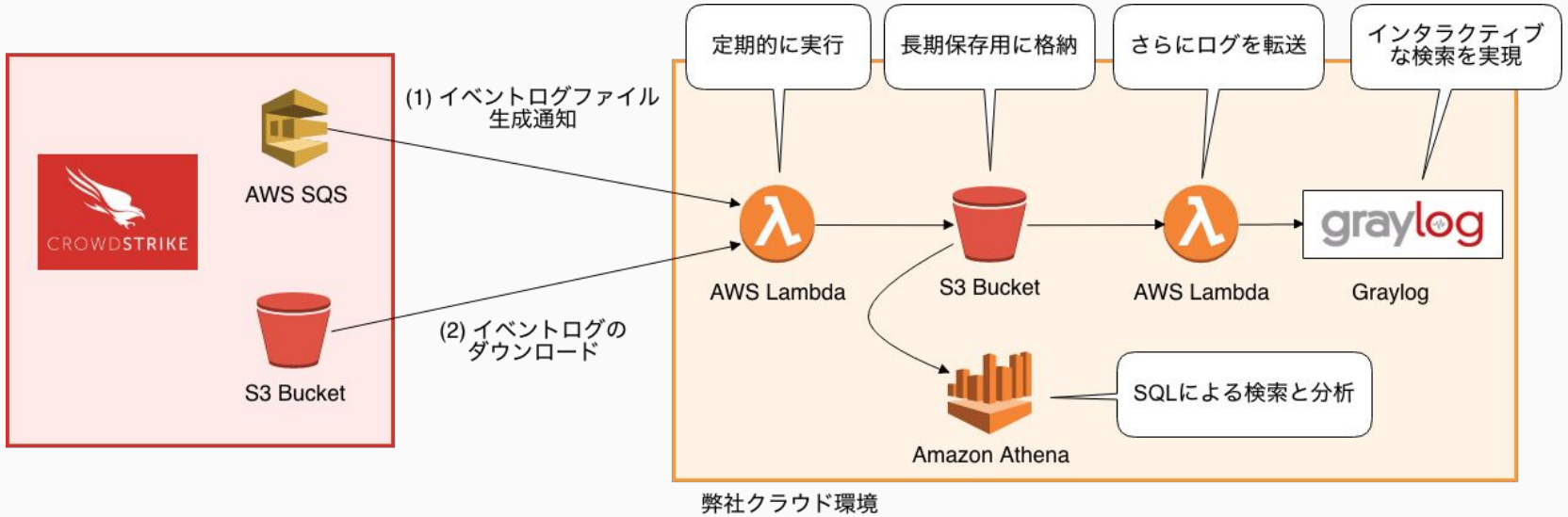
EDRのシステム統合における弊社の機能要件（おさらい）

1. APIでアラート（検知情報）を受け取れること
    - 既存の弊社セキュリティ監視システムと統合する
    - アラートの通知や対応状況の管理
2. APIからログの検索ができること
    - Falconが検知したアラートについての調査
    - 他のセキュリティ監視装置が検知したアラートの調査

全体のアーキテクチャ概要

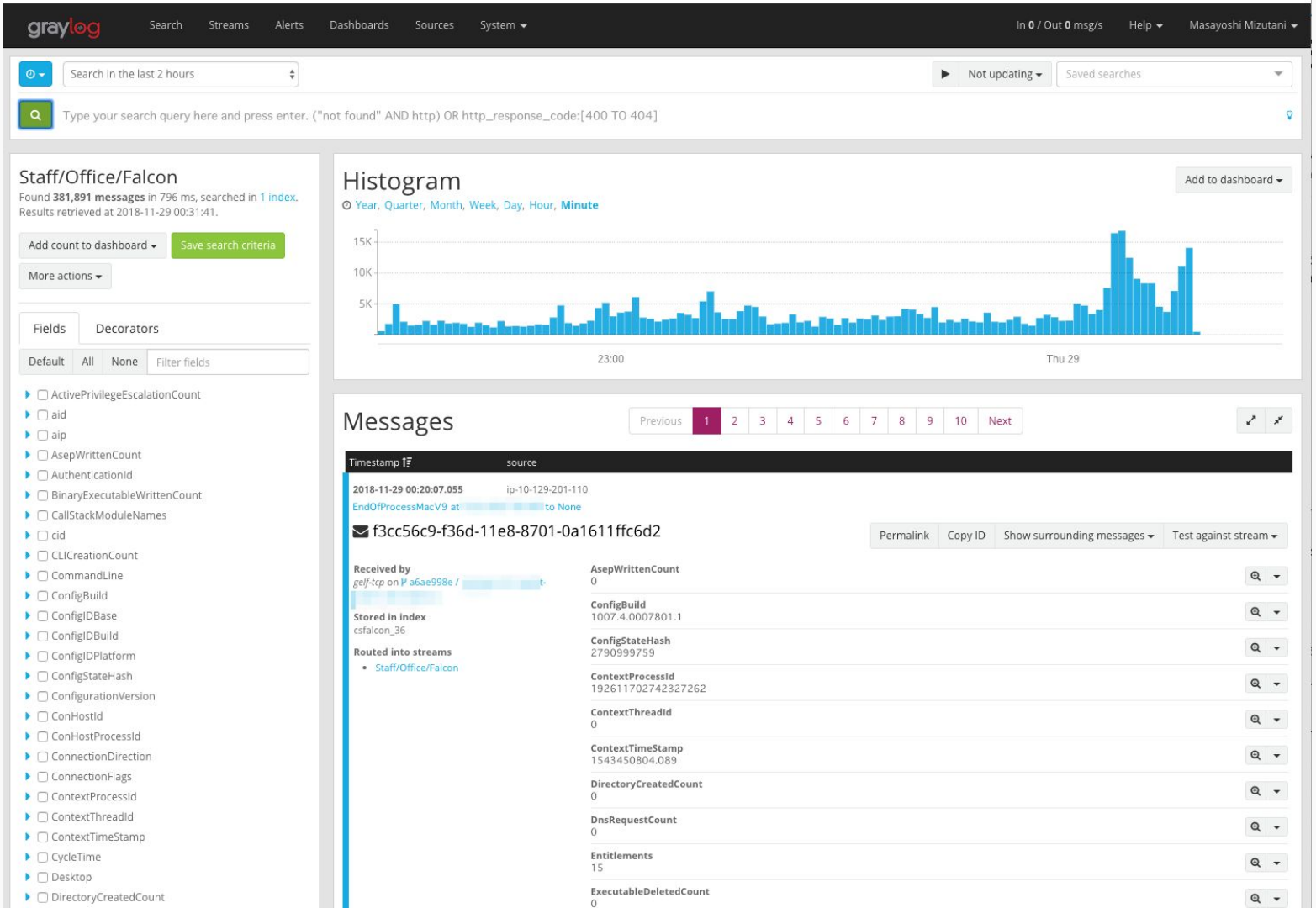


イベントログの取り込み詳細



SQS: キュー管理サービス
S3: オブジェクトストレージ
Lambda: サーバレスコード実行サービス
Athena: S3からのデータ検索サービス

実装 https://github.com/m-mizutani/aws-falcon-data-forwarder


graylog
Search
Streams
Alerts
Dashboards
Sources
System
In 0 / Out 0 msg/s
Help
Masayoshi Mizutani
Search in the last 2 hours
Not updating
Saved searches
Type your search query here and press enter. ("not found" AND http) OR http_response_code:[400 TO 404]
Staff/Office/Falcon
Found 381,891 messages in 796 ms, searched in 1 index.
Results retrieved at 2018-11-29 00:31:41.
Add count to dashboard
Save search criteria
More actions
Fields
Decorators
Default
All
None
Filter fields
ActivePrivilegeEscalationCount
aid
aip
AsepWrittenCount
AuthenticationId
BinaryExecutableWrittenCount
CallStackModuleNames
cid
CLICreationCount
CommandLine
ConfigBuild
ConfigIDBase
ConfigIDBuild
ConfigIDPlatform
ConfigStateHash
ConfigurationVersion
ConHostId
ConHostProcessId
ConnectionDirection
ConnectionFlags
ContextProcessId
ContextThreadId
ContextTimeStamp
CycleTime
Desktop
DirectoryCreatedCount
Histogram
Add to dashboard
Year, Quarter, Month, Week, Day, Hour, Minute
15K
10K
5K
23:00
Thu 29
Messages
Previous
1
2
3
4
5
6
7
8
9
10
Next
Timestamp
source
2018-11-29 00:20:07.055
ip-10-129-201-110
EndOfProcessMacV9 at
to None
f3cc56c9-f36d-11e8-8701-0a1611ffc6d2
Permalink
Copy ID
Show surrounding messages
Test against stream
Received by
gelf-tcp on
Stored in index
csfalcon_36
Routed into streams
Staff/Office/Falcon
AsepWrittenCount
0
ConfigBuild
1007.4.0007801.1
ConfigStateHash
2790999759
ContextProcessId
192611702742327262
ContextThreadId
0
ContextTimeStamp
1543450804.089
DirectoryCreatedCount
0
DnsRequestCount
0
Entitlements
15
ExecutableDeletedCount
0

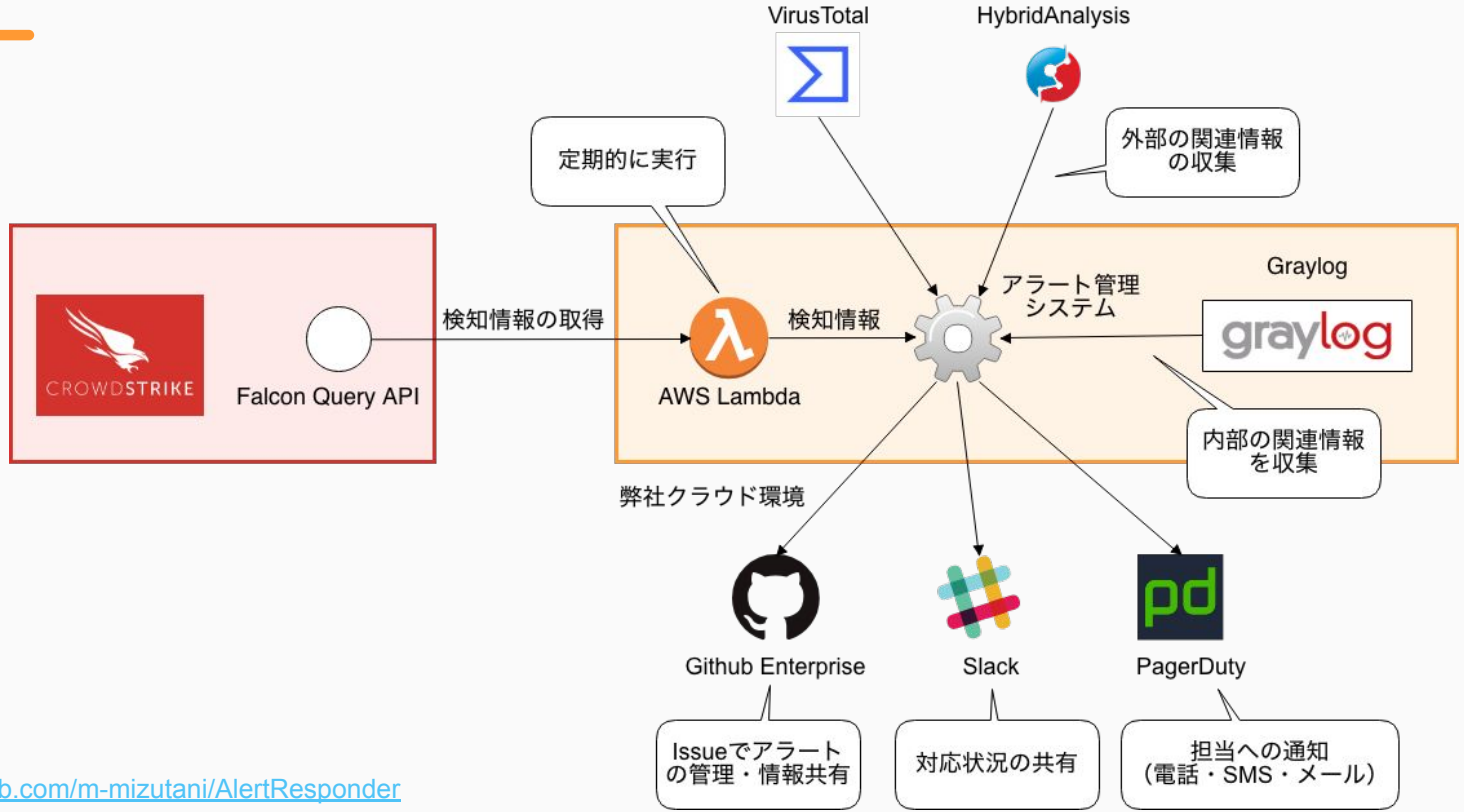
イベントログの取り込み設計のポイント (1/2)

- Falconの S3 + SQS というログの出力方法を活用
  - 他製品だとsyslogで転送する形式が多かったが障害発生時のリカバリ対応が困難という問題があった
  - Falcon側のS3に一定期間ログが保存されているので転送のやり直しが容易・弊社側で流量の調整が可能
  - S3に保存してから各種処理をする既存の仕組みとの相性が良かった
- 弊社環境へ転送した際もまずはS3に格納する
  - S3の可用性＆スケーラビリティの高さ、価格の安さを活用

イベントログの取り込み設計のポイント (2/2)

- 短期的なログはGraylogに格納して他のログも含めた横断的検索を実現
  - ログ種別に応じて1週間～1ヶ月程度
  - 高速でinteractiveな横断的ログ検索が可能
    - 例）C&Cサーバの IPアドレスでログ抽出し通信が発生していたかなど確認
    - 例）アラートに関連したユーザ名でログ抽出しアラート前後の行動を把握
- Athenaで長期保存されているログの検索
  - GraylogはDB（ストレージ）が高価なので長期的な検索はS3を利用
  - 監査的な利用およびインシデント発生時の追跡用

検知情報（アラート）の取り込み詳細



実装 https://github.com/m-mizutani/AlertResponder

検知情報（アラート）の取り込み設計ポイント

- 共通したアラート対応基盤を利用する
  - Falcon以外のアラートも共通のインターフェースで扱える
- 対応開始までの時間差を許容できるかの検討
  - アラートが発生してから発報されるまで4～5分遅延がある
  - 自社内の対応スピードと照らし合わせて許容範囲と判断
- 自動化できる部分はなるべく自動化する
  - 検出されたIPアドレスやドメイン名の調査などは自動的に実行
  - 調査のための時間を短縮し、人間の時間を有効に使う

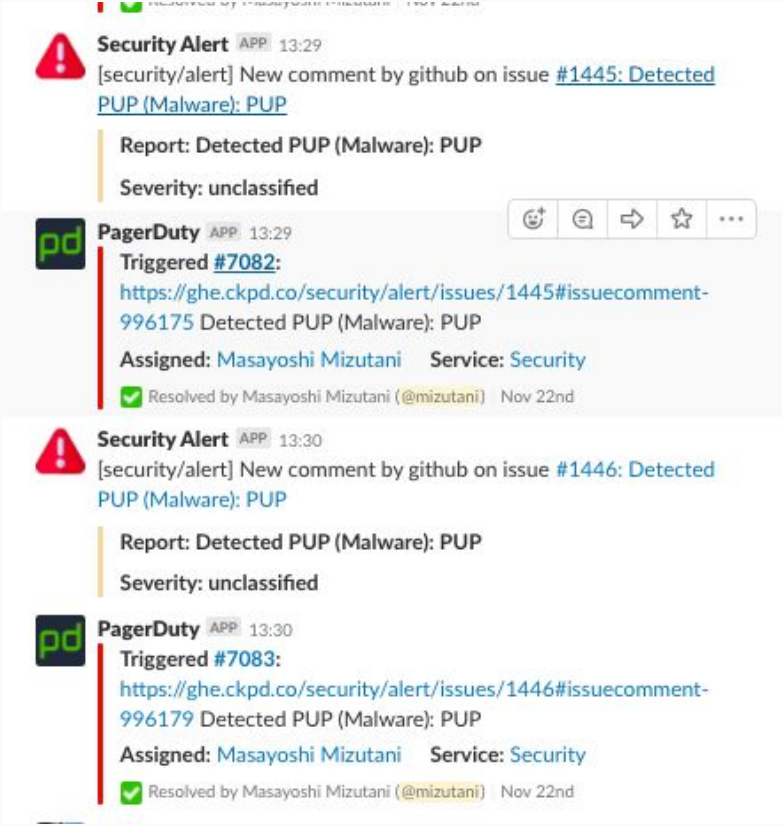
CrowdStrike Falconの運用

cookpad

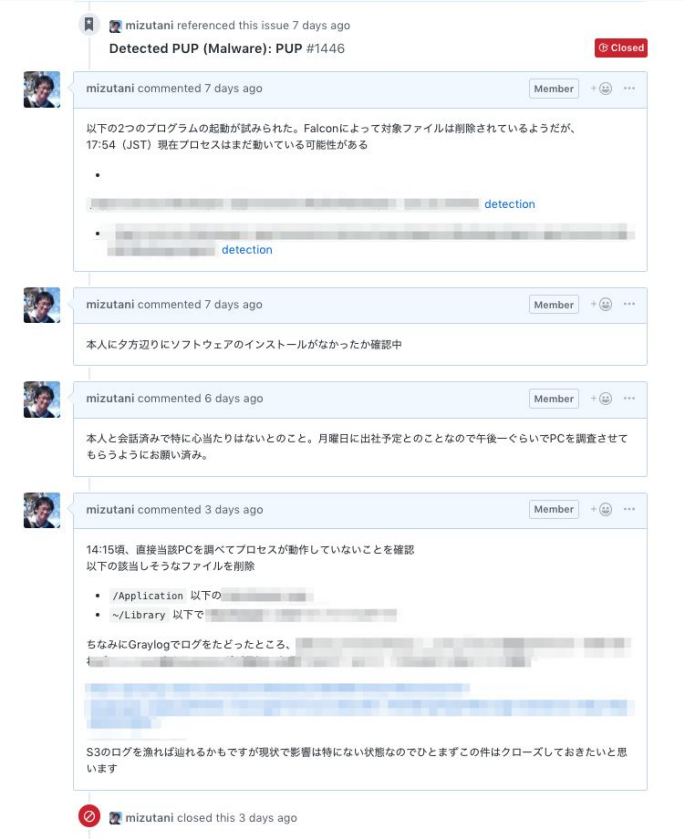
対応フロー

1. アラート管理システムからPagerDutyに通知
    - メール、SMS、電話などで担当者に通知
    - 状況共有のためSlackにも通知
2. 担当者が調査（3人でdailyで担当を交代）
    - 一部情報は自動的に取得されてGithub EnterpriseのIssueに書き込み
    - Graylogなどを使って担当者がアラートを調査
    - 調査結果や進捗をIssueにコメントし、対応完了したらCloseする

Slack上で初動対応状況の共有



Github EnterpriseのIssue上で対応の記録＆共有



今後の展望＆製品への期待

cookpad

今後の展望

- 検知結果の（より）自動的な分析・対応
  - 社内や外部の情報を分析前に自動的に集める
  - ある条件を満たしていたら自動的にFalse Positiveにする
- IoC情報の検索
  - 取り込んだログを使ってIoC情報を検索する
  - 後から判明するIoC情報も多いため時間差での対応が必要

製品への期待

- より強力なAPI連携
  - デバイスの管理：デバイスの情報変更や削除
  - イベントの検索：Event Searchの機能にAPI経由でアクセス
- 監査目的でのログ収集
  - ログ収集における制限の緩和・撤廃
- 外部サービスとの連携
  - SSO連携の拡充

Thank you