



AWS Lambda環境への攻撃デモとその対策 Trend Micro Cloud One - Application Security の活用方法とユースケースのご紹介

トレンドマイクロ株式会社
AWSアライアンス担当
姜 貴日



姜 貴日 - Kwiil Kang

トレンドマイクロ株式会社
セールスエンジニアリング部
AWS Alliance Tech Lead



「Security Automation」、「DevSecOps」、「Container」など
よりクラウドと親和性が高い領域に特化したソリューション提案を行う。

AWS Summit Tokyo 2019



JAWS DAYS 2020



トレンドマイクロ Webinar 2020



トレンドマイクロ
& New Relic共催セミナー



AWS Lambdaにおける責任共有モデルの考え方

- AWS Lambda については、AWS が基盤となるインフラストラクチャ、基盤サービス、オペレーティングシステム、アプリケーションプラットフォームを管理します。
- お客様 は、**コードのセキュリティ、機密データの保管とアクセス、AWS Lambda サービスに対する アイデンティティとアクセスの管理 (IAM)、関数内のアイデンティティとアクセスの管理**について責任を負います。



トレンドマイクロが
お手伝い出来るところ

- ・ Webアプリケーション自身の保護
- ・ Webアプリケーション経由でのデータ保護

~クラウドセキュリティはまとめてシンプルに~ Trend Micro Cloud One™



Trend Micro
Cloud One™

Trend Micro Cloud One

- Workload Security

クラウドワークロードおよびコンテナの保護



Amazon Elastic Compute Cloud (Amazon EC2)



Amazon Elastic Container Service (Amazon ECS)



Amazon Elastic Kubernetes Service (Amazon EKS)



AWS Elastic Beanstalk

- Network Security

クラウド向けネットワークIPS



Amazon Virtual Private Cloud (Amazon VPC)

- Container Security

コンテナイメージのスキャンとデプロイの制御



Amazon Elastic Container Registry (Amazon ECR)



Amazon Elastic Kubernetes Service (Amazon EKS)

- Application Security

サーバレスおよびアプリケーションの保護



AWS Fargate



Amazon Elastic Container Service (Amazon ECS)



Amazon Elastic Kubernetes Service (Amazon EKS)



AWS Lambda

- File Storage Security

クラウドストレージの不正プログラムスキャン



Amazon Simple Storage Service (Amazon S3)

- Conformity

クラウドの設定不備を可視化、コンプライアンス対応支援



AWS Cloud

Cloud One - Application Security

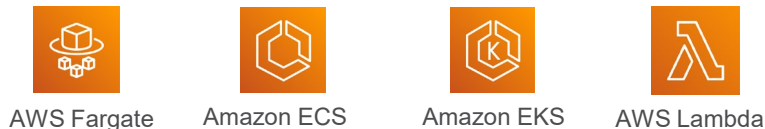
※以降、Application Securityと記載

アプリケーション自身にセキュリティを実装（RASP）することで、アプリケーションを保護。
コンテナマネージドサービスやサーバレス環境も保護することができます。

対応言語



対象サービス



■ システム要件はこちらを参照ください

<https://cloudone.trendmicro.com/docs/application-security/install-agent/>

提供機能

- アプリケーションに対する下記攻撃の検知・防御
 - 悪意のあるペイロード（IPS/IDS機能相当）
 - SQLインジェクション
 - リモートコマンド実行
 - オープンリダイレクト
 - 不正なファイルアクセス
 - 不正なファイルアップロード
 - IPフィルタリング

特徴

- 様々な環境・言語をサポート
各言語にパッケージとして提供
- 数行のコードを書き込むだけで完了
 - ソースコードの大きな変更は不要
 - パフォーマンス低下と展開負荷を最小限に

Application Securityの実装例

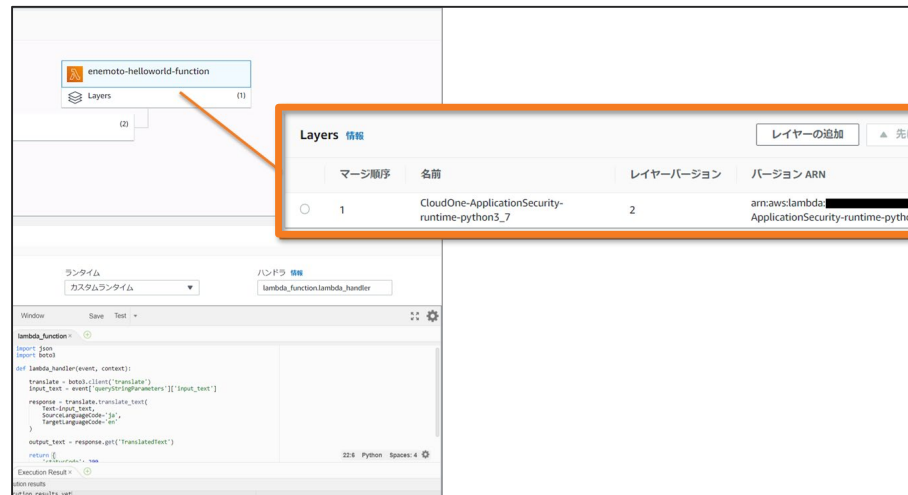
※RASP (Runtime Application Self Protection) とは？

セキュリティをアプリケーションの機能の一部として組み込むことで、外部からアプリケーションに対する攻撃を検知・ブロックする保護手法。

例 1 ライブラリをインポートする

```
1 """
2 WSGI config for projmanager project.
3
4 It exposes the WSGI callable as a module-level variable named
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/1.7/howto/deployment/wsgi/
8 """
9 import trend_app_protect.start
10 import os
11 os.environ.setdefault("DJANGO_SETTINGS_MODULE", "taskManager.
12
13 from django.core.wsgi import get_wsgi_application
14 application = get_wsgi_application()
~
~
```

例 2 AWS Lambda Layerを利用する



The screenshot shows the AWS Lambda console interface. A function named 'enemoto-helloworld-function' is selected, and its 'Layers' section is expanded. A table titled 'Layers 情報' (Layers Information) is displayed, showing the attached layer details.

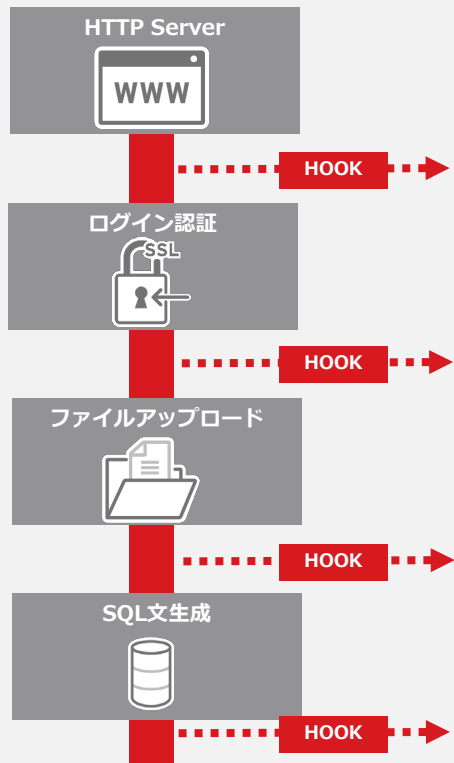
マージ順序	名前	レイヤーバージョン	バージョンARN
1	CloudOne-ApplicationSecurity-runtime-python3_7	2	arn:aws:lambda:ap-northeast-1:123456789012:layer:ApplicationSecurity-runtime-pytho...

The console also shows the function's handler as 'lambda_function.lambda_handler' and a preview of the handler code, which includes a call to 'translate.translate_text'.

アーキテクチャ概要

Webアプリケーション（コンテナ）

アプリケーションStack例



Application Security Agent



スキャンレポート
(検出結果等)



ポリシー設定

管理コンソール（SaaS提供）



Cloud OneはSaaSとなりますので
管理サーバの構築や運用は不要です。

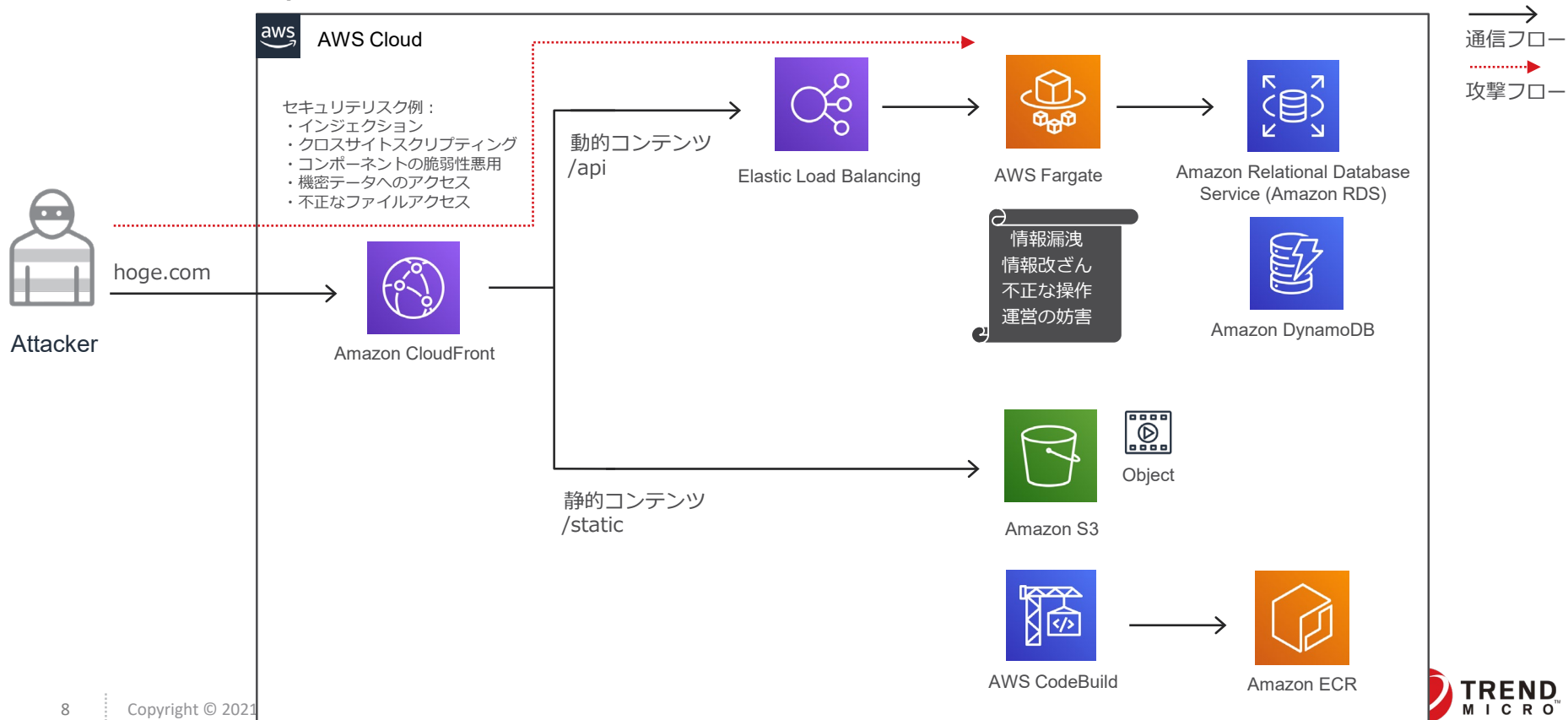
管理者



管理コンソール経由で
スキャン結果やポリシー設定
を操作

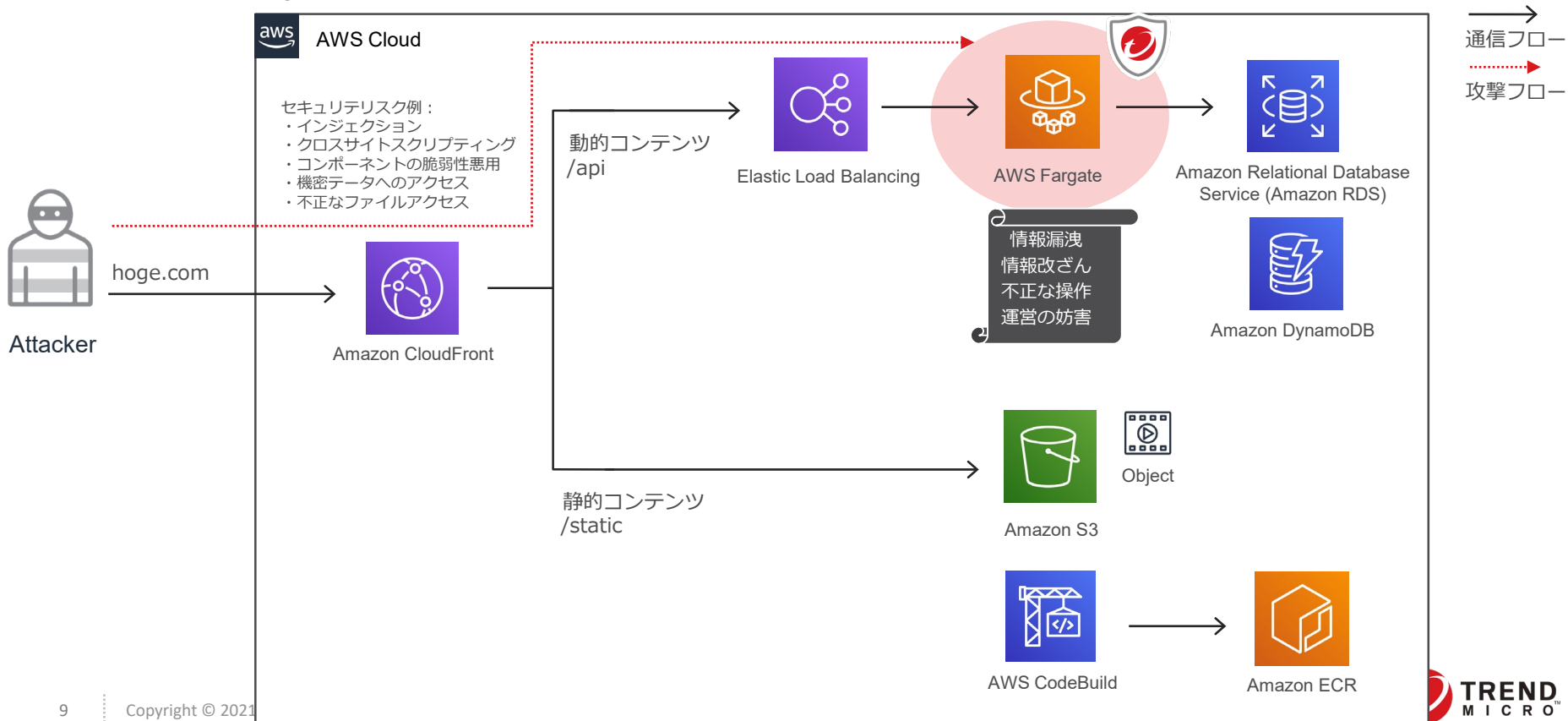
Webアプリケーション起点の攻撃例

例) ECサイトのシステム



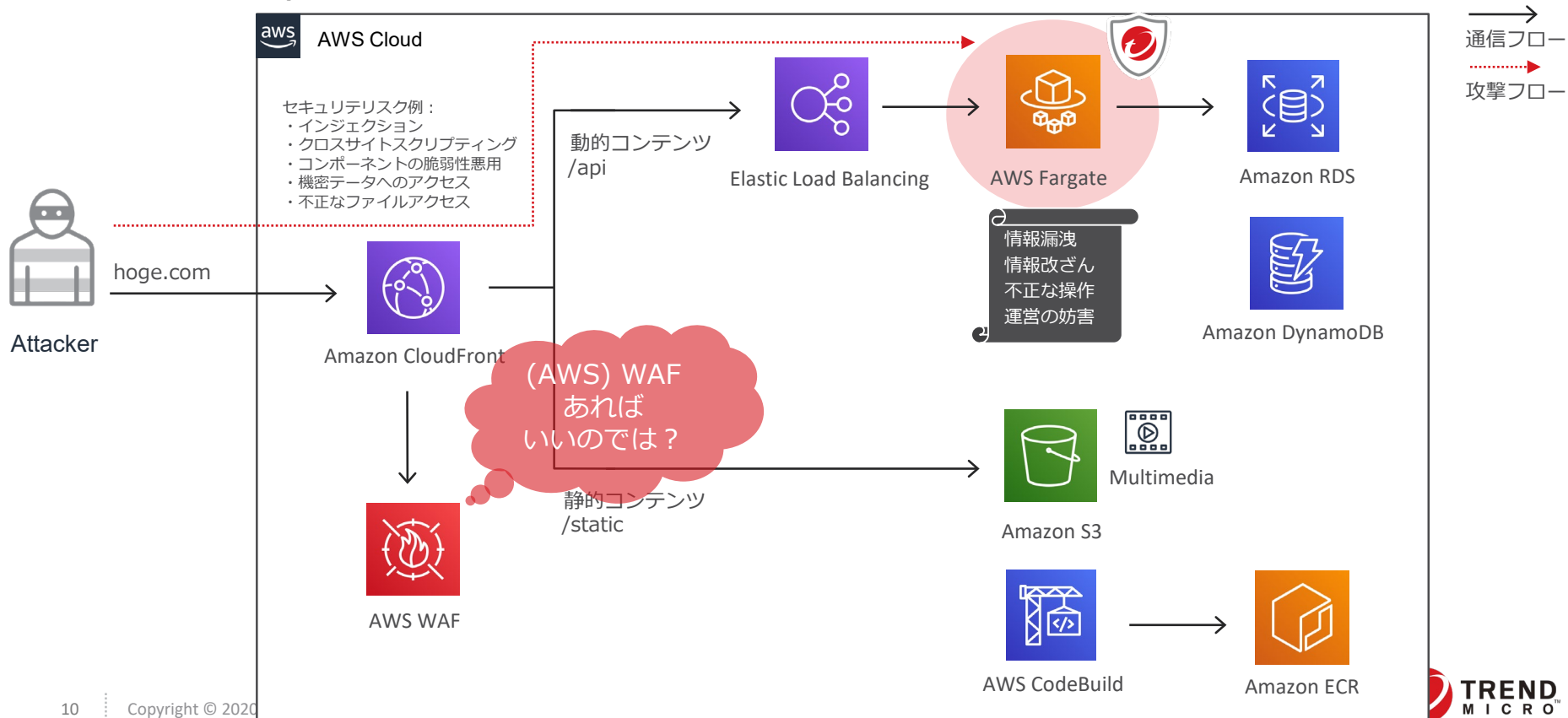
Webアプリケーション起点の攻撃からシステムを守る

例) ECサイトのシステム

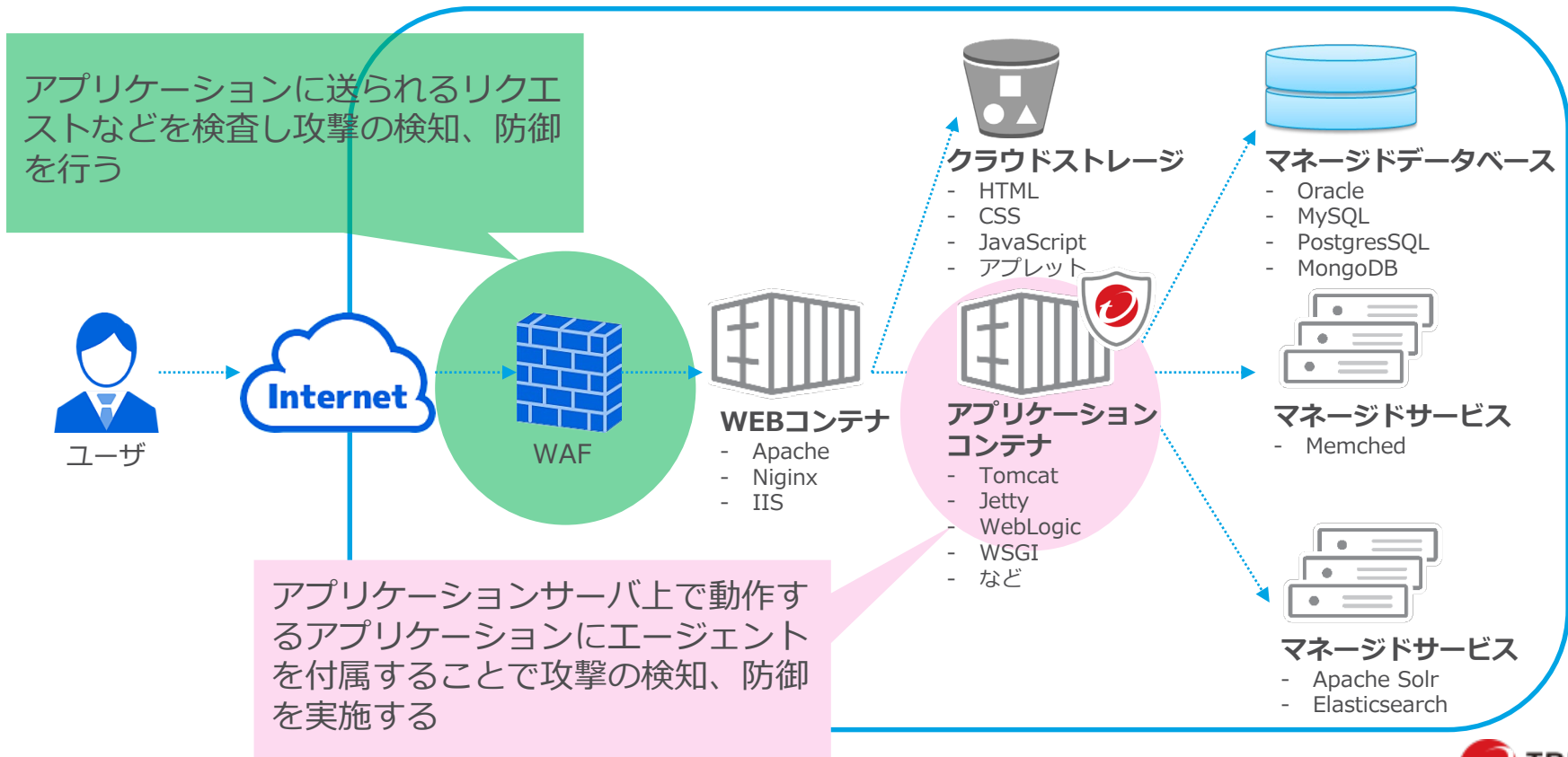


Webアプリケーション起点の攻撃からシステムを守る

例) ECサイトのシステム

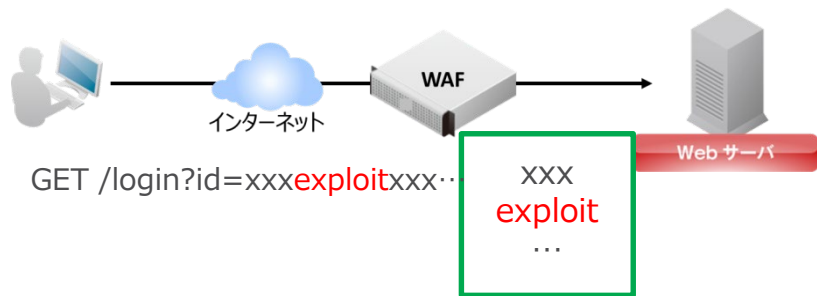


動作するレイヤーの違い



検知方法の違い

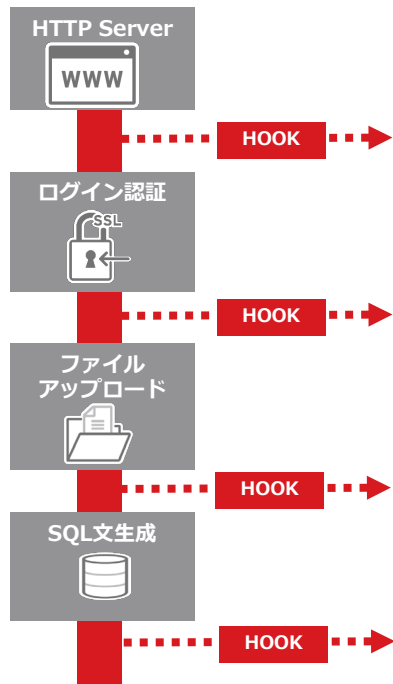
WAF



- WAFは通信のシグネチャベース。
- Application Securityはシグネチャマッチに加えてアプリケーションの動きも制御。

Application Security

アプリケーションStack例



Application Security モジュール



ポリシー違反の動作ではないか確認

違反の場合



検知

あるいは

ブロック

提供しているセキュリティ機能の違い

WAF

セキュリティ機能	内容
 Web Application 保護	Webアプリケーションへの攻撃からウェブサイトを保護します。
 IPS/IDS(侵入防御)	あらかじめ設定したルールに該当する通信を検知・ブロックします。

➡ WAF製品によって、IPS/IDSの有無は異なります。

Application Security

セキュリティ機能	内容
SQLインジェクション対策	クエリを検査し攻撃検知をします。
リモートコマンド実行防止	意図しないコマンド実行を防ぎます。
リダイレクト攻撃対策	意図しないリダイレクトを防ぎます。
不正なファイルアクセス対策	意図しないファイルアクセスを防ぎます。
不正なファイルアップロード対策	アップロードされたファイルがウィルスでないか確認します。
悪意のあるペイロード対策	仮想パッチ技術を用いて、脆弱性を突いた攻撃からサーバを保護します。 Web Application保護 + IPS/IDS

- 一般的なWAFとApplication Securityは搭載している機能が異なります。
- WAFとApplication Securityを組み合わせることによって、攻撃フェーズに沿って多層的な対策を実施することができます。
- 保護対象の環境にはどのような対策が必要なのか考慮した上で製品選定する必要があります。



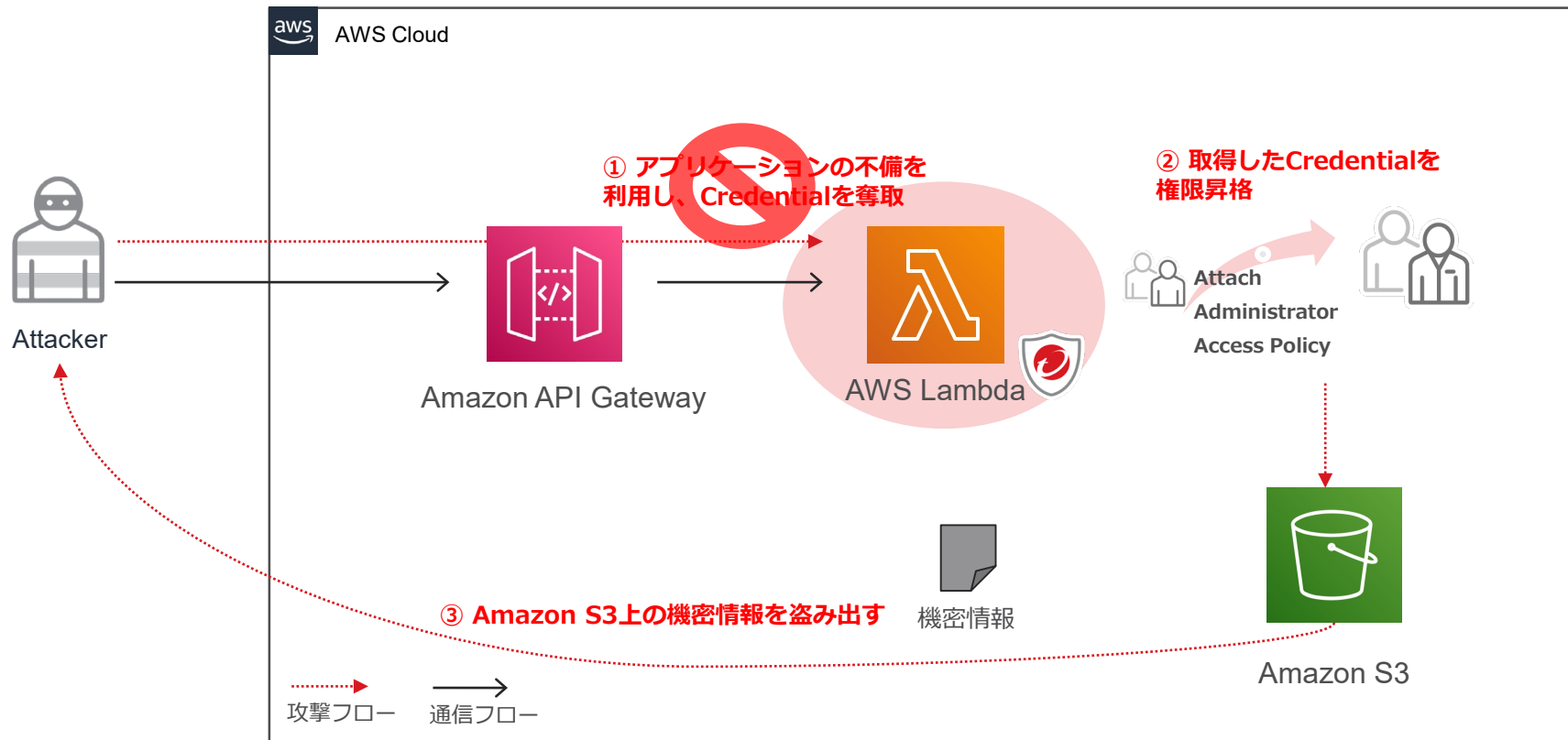
デモ

AWS Lambdaを狙った機密情報の搾取とその対策

Webアプリケーションの不備から機密情報を窃取



Application Securityにて脅威を遮断





案件事例

■業種

サービス業

■案件背景

- ・新規スモールスタート案件
→人員やリソースが限られている
- ・開発者主体チーム
→セキュリティナレッジも限られている

■きっかけ

- ・自分達の運用にマッチしたサーバレス向けセキュリティ製品を探していた

■運用課題

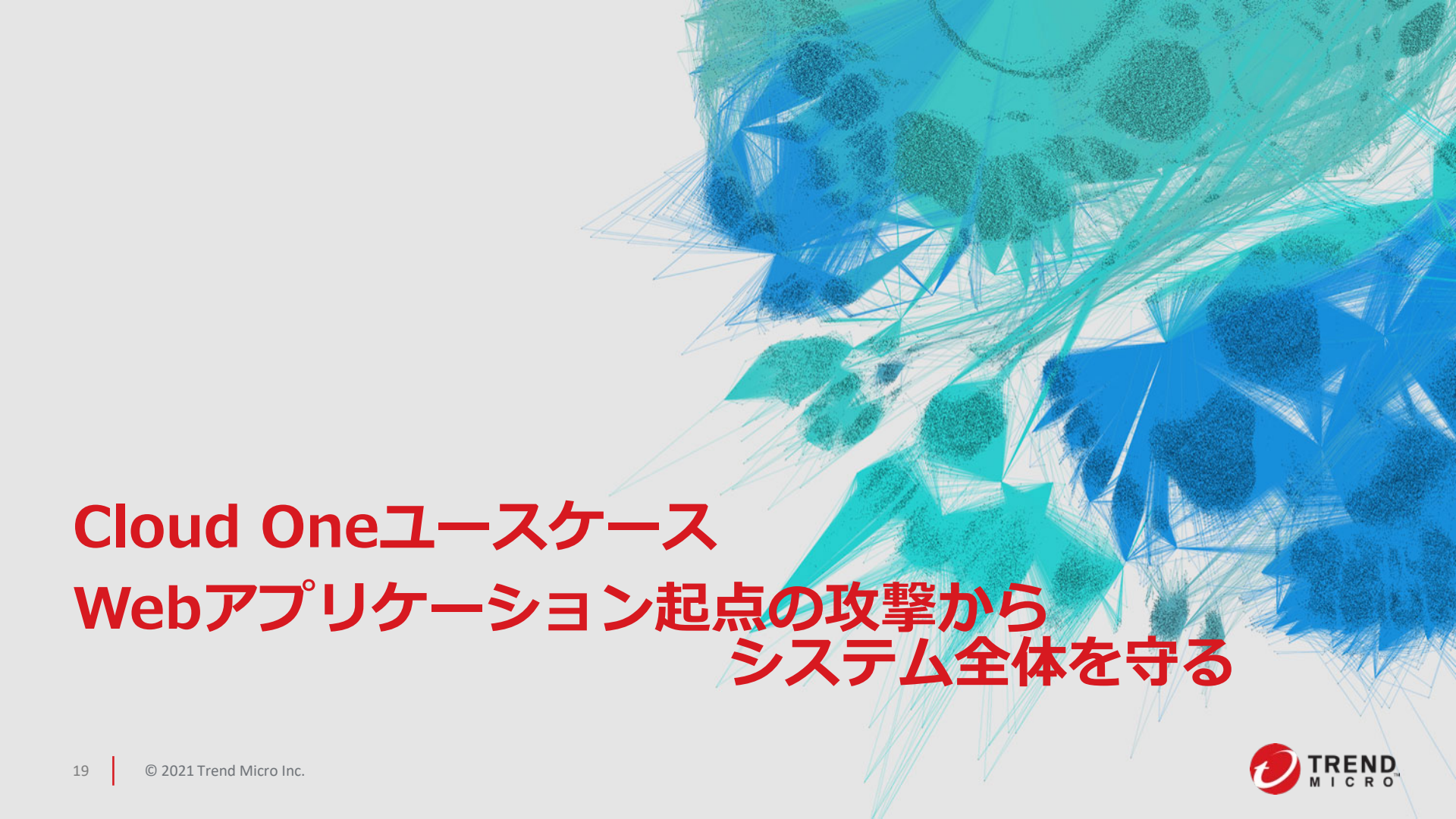
- ・導入、運用共にシンプルにしたい
- ・製品導入に伴い保守対象を増やすのはNG

■選定理由

- ・Agentをアプリケーションに組み込む（RASP）だけで実装出来る
- ・SaaS製品なので管理サーバや脆弱性を管理する為のサーバ（DB）が不要

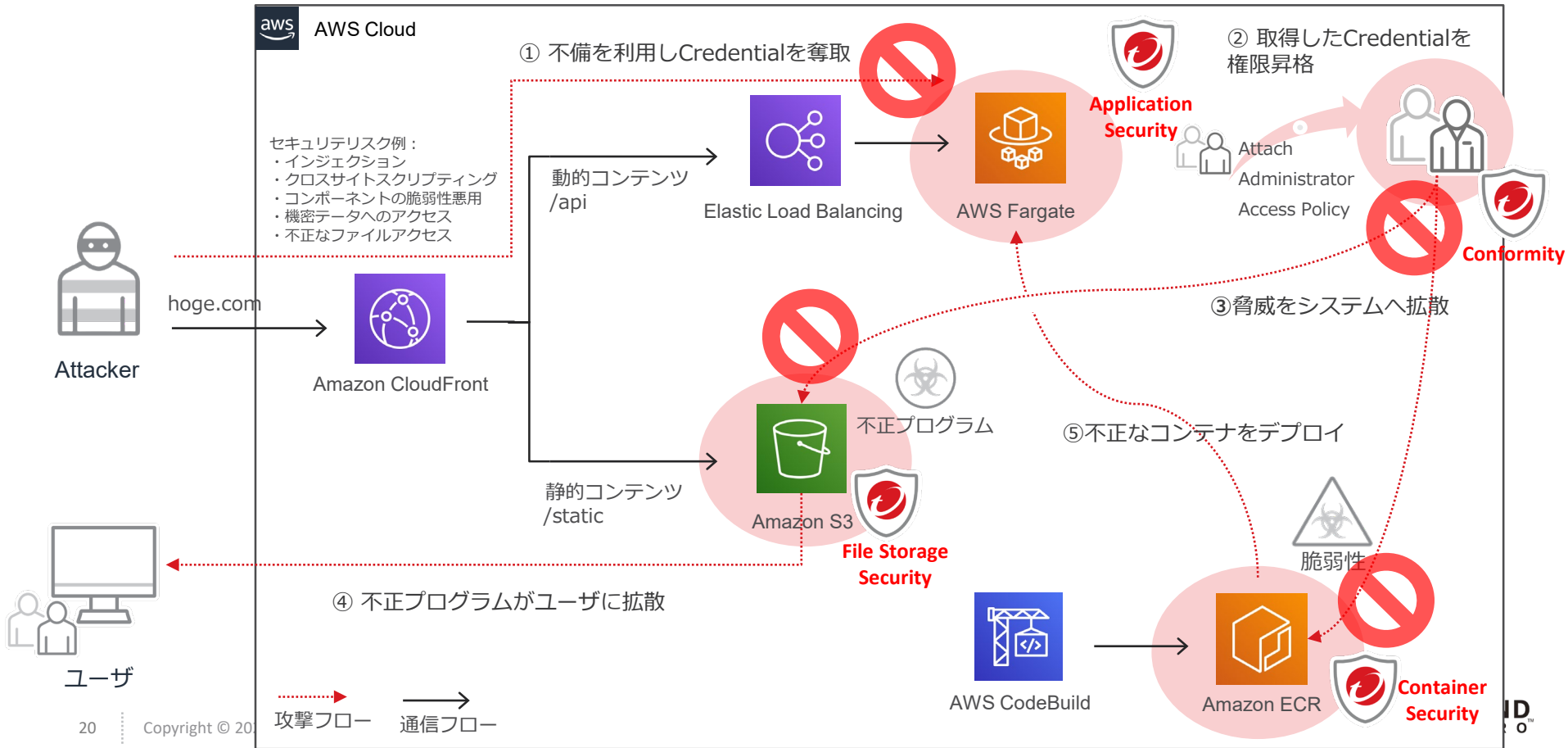
■お客様の声

- ・導入がシンプル
- ・WAF運用領域を狭める事で運用負荷軽減につながった

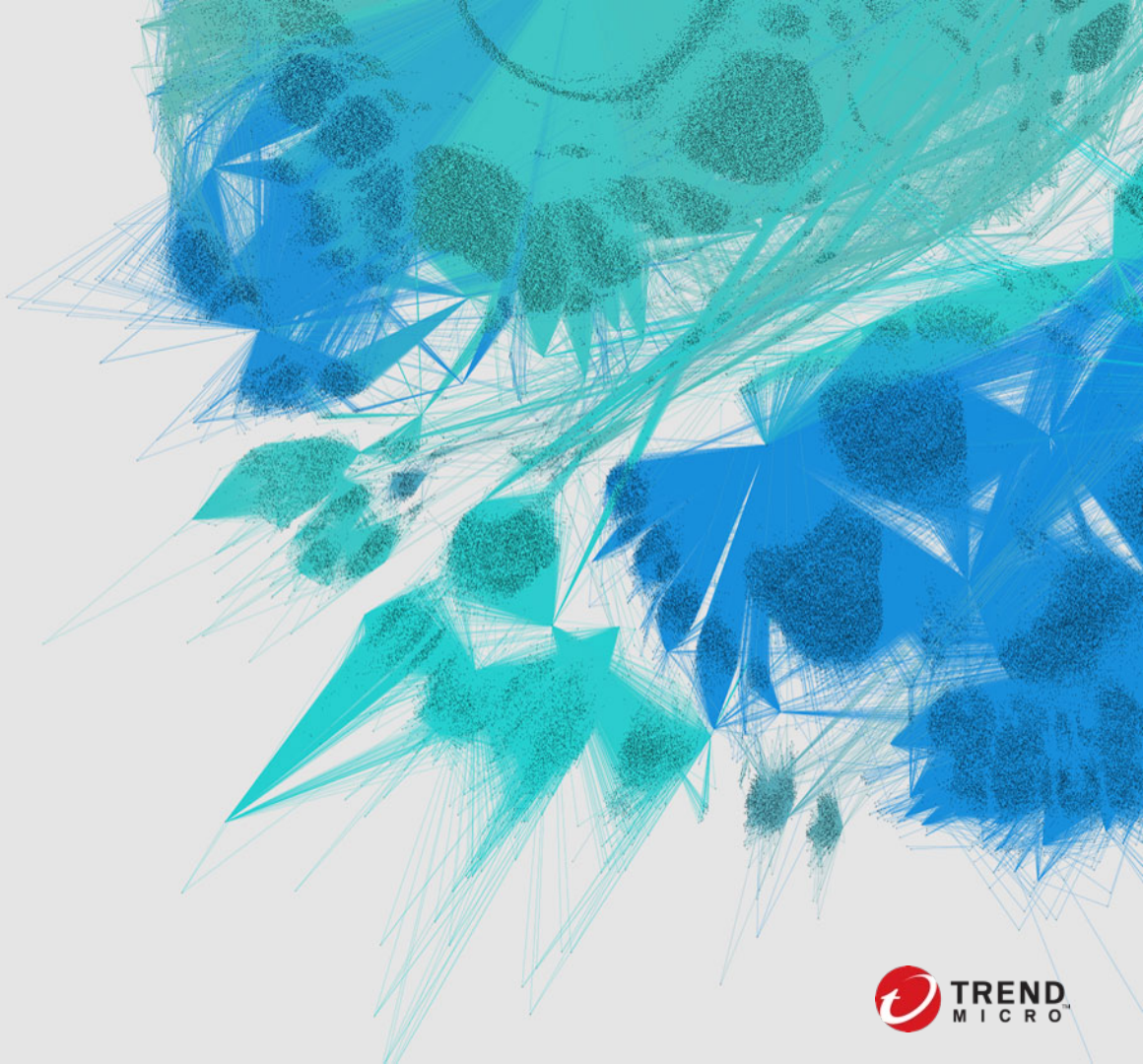


Cloud Oneユースケース Webアプリケーション起点の攻撃から システム全体を守る

Cloud Oneユースケース：Webアプリケーション起点の攻撃からシステム全体を守る



まとめ



まとめ

- ✓ サーバレス（AWS Lambda）でも責任共有モデルの考え方は重要
- ✓ サーバレスの特徴に合わせて防御戦略をとる
 - ✓ 稼働は短命、OSの脆弱性管理不要、自由度も限定的
- ✓ サーバレスユーザが注意すべきはアプリケーションを狙う脅威
 - ✓ 脆弱性からの侵害
 - ✓ 過大な権限付与からの権限昇格
- ✓ サーバレスでも基本のセキュリティ対策は有効
 - ✓ FaaS上で多層防御
(脆弱性の検査、ロギング、可視化、データへのアクセス制御)

AWS環境でのセキュリティ対策に関するご相談は
aws@trendmicro.co.jp へご連絡ください!!

AWS Summit Online 2021 にも出展します



■ セッション情報

AWS Lambdaを攻撃してみた ～サーバレスのセキュリティの考え方～

5月11日（火）15時30分～ ※これ以降は随時オンデマンド視聴が可能になります

ライブ配信 セッションのアジェンダ

1

すべてのセッション マイアジェンダ

Q PAR-2d X

日付を選択 全セッションタイプ 全ソリューション

May 11, 2021 15:30 - 16:00 JST

AWS Lambdaを攻撃してみた～サーバレスのセキュリティの考え方～ (パートナー...)

サーバレス環境のセキュリティの考え方

2

「サーバレス環境のセキュリティは必要ないと思いませんか？ そんな方にご参加いただきたい「サーバレス環境のセキュリティの考え方」です。サーバレス環境での責任共有モデルに基づくセキュリティの考え方をご紹介します。AWS... [詳細情報](#)

📅 カレンダーに追加する ☆ マイアジェンダに追加

サーバレスコンピ... セキュリティ、アイ...

石原博平氏 藤井日氏
トレンドマイクロ株式会社

AWS Lambdaを攻撃してみた～サーバレスのセキュリティの考え方～

Security
JAMにも
出展！



■ オンラインブース出展

本日よりご紹介しましたCloud Oneシリーズのご紹介資料を掲載しております。AWS環境に合わせたセキュリティ展開を体系的に展示しておりますので是非ともお立ち寄りください！

Thank you !!