株式会社サイバーセキュリティクラウド

クラウドWAFの機能解説と自動運用サービスについて

CYBER SECURITY CLOUD

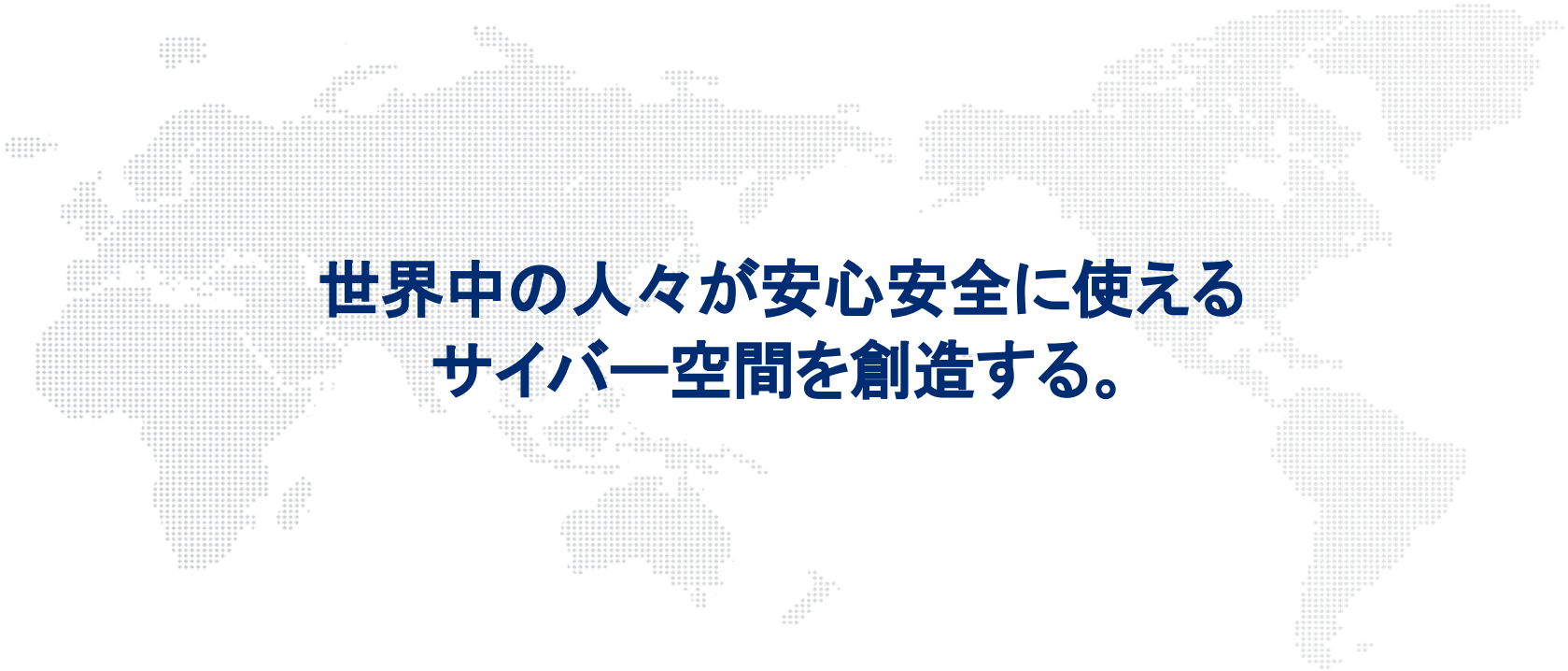
目次

- ■ 会社紹介
- □ WAFについて
- □ パブリッククラウドのManaged WAF
- □ 運用上の悩みポイント

会社概要

社　　名	株式会社サイバーセキュリティクラウド
設　　立	2010年8月11日
上 場 日	2020年3月26日
代 表 者	代表取締役社長 兼 CEO　小池 敏弘 代表取締役CTO　　渡辺 洋司
役　　員	取締役CFO　倉田 雅史（公認会計士） 社外取締役　伊倉 吉宣（弁護士） 社外取締役　栗原 博 常勤監査役　関 大地（公認会計士） 社外監査役　泉 健太 社外監査役　村田 育生
所 在 地	東京都品川区上大崎3-1-1 JR東急目黒ビル13階
事業内容	AI 技術を活用した サイバーセキュリティサービスの開発・提供
グループ会社	Cyber Security Cloud Inc.（USA）

ミッション



世界中の人々が安心安全に使える
サイバー空間を創造する。

当社のセキュリティサービスラインアップ

4つのプロダクトを自社開発・自社サポートで安心を提供する国産セキュリティーメーカー

クラウド型WAF

攻撃遮断くん

Webサイトへのサイバー攻撃の可視化・遮断ツール

国内シェア
No.1 ※1

パブリッククラウド WAF 自動運用サービス

Waf Charm

AIによるAWS/ Azure / Google Cloudの各種WAF自動運用ツール

AWS WAF 自動運用サービス
導入ユーザー数
国内
No.1 ※2

AWS WAF 専用 ルールセット

AWS WAF
Managed Rules

サイバーセキュリティクラウド独自のAWS WAF専用のルールセット

導入ユーザー数
累計**90カ国**以上
3,273 ユーザー ※3

脆弱性情報 収集・管理ツール

SIDfm

脆弱性情報を収集し、
パッチ情報や回避方法を
提供するサービス

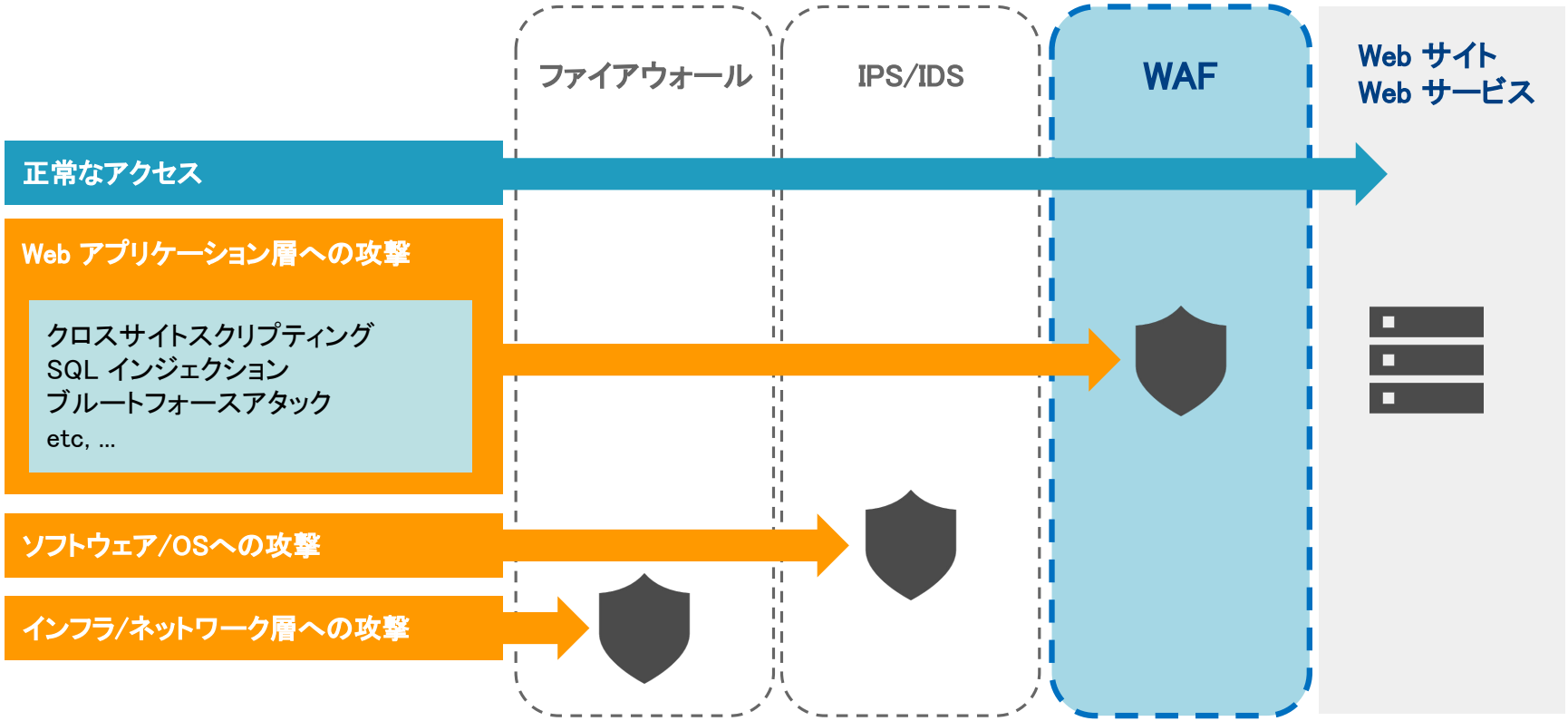
脆弱性情報配信サービスシェア
脆弱性情報提供実績
脆弱性オリジナルコンテンツ数
3部門国内
No.1 ※4

※1 日本マーケティングリサーチ機構調べ 調査概要:2021年10月期_実績調査
※2 日本マーケティングリサーチ機構調べ 調査概要:2020年7月期_実績調査
※3 2023年3月時点
※4 日本マーケティングリサーチ機構調べ 調査概要:2021年8月期_実績調査

目次

- □ 会社紹介
- ■ WAFについて
- □ パブリッククラウドのManaged WAF
- □ 運用上の悩みポイント

WAFとはL7の防御層



OWASP Top 10（最新版 2021年）

- ▲ アクセス制御の不備
- ▲ 暗号化の失敗
- ✔ インジェクション
- ▲ 安全が確認されない不安な設計
- ▲ セキュリティの設定ミス
- ✔ 脆弱で古くなったコンポーネント（ゼロデイ攻撃含む）
- ▲ 識別と認証の失敗
- ▲ ソフトウェアとデータの整合性の不具合
- ✔ セキュリティログとモニタリングの失敗（WAFが監視とロギングも担う）
- ▲ サーバーサイドリクエストフォージェリ

OWASP Top 10（最新版 2021年）

- ▲ アクセス制御の不備
- ▲ 暗号化の失敗
- ✓ インジェクション
- ▲ 安全が確認されない不安な設計
- ▲ セキュリティの設定ミス
- ✓ 脆弱で古くなったコンポーネント（ゼロデイ攻撃等も）
- ▲ 識別と認証の失敗
- ▲ ソフトウェアとデータの整合性の不具合
- ✓ セキュリティログとモニタリングの失敗（WAFが監視とロギングも担う）
- ▲ サーバーサイドリクエストフォージェリ

- HTTPヘッダーインジェクション
- LDAPインジェクション
- OSコマンドインジェクション
- SQLインジェクション
- SSCインジェクション
- XPathインジェクション
- コマンドインジェクション
- 改行コードインジェクション
- メールヘッダ・インジェクション
- NULLバイトインジェクション

チームでのセキュアコーディング
の徹底OSSの実装調査には限界がある。

SQLインジェクションの例

SELECT * FROM user WHERE id='$ID'

実装者の意図

$IDには、UIからユーザのID「**taro**」が渡ってくることで以下のようなクエリが実行されるはず

SELECT * FROM user WHERE id='**taro**'

この結果、'taro'に関連した情報だけがUIに返却される

攻撃者の意図

$IDに、必ず成立する条件文を付与して指定する（**taro' or 'A'='A**）ことでクエリの条件文を誤魔化そう

SELECT * FROM user WHERE id='**taro' or 'A'='A**'

この結果、本来閲覧できるべきではない情報までが抽出されてしまう

・基本はプログラム側で脆弱性のないよう実装する、セキュリティパッチを迅速にあてる
　といった対応が必要
・それでも必ずしも万全には出来ないので、WAFで検知・防御

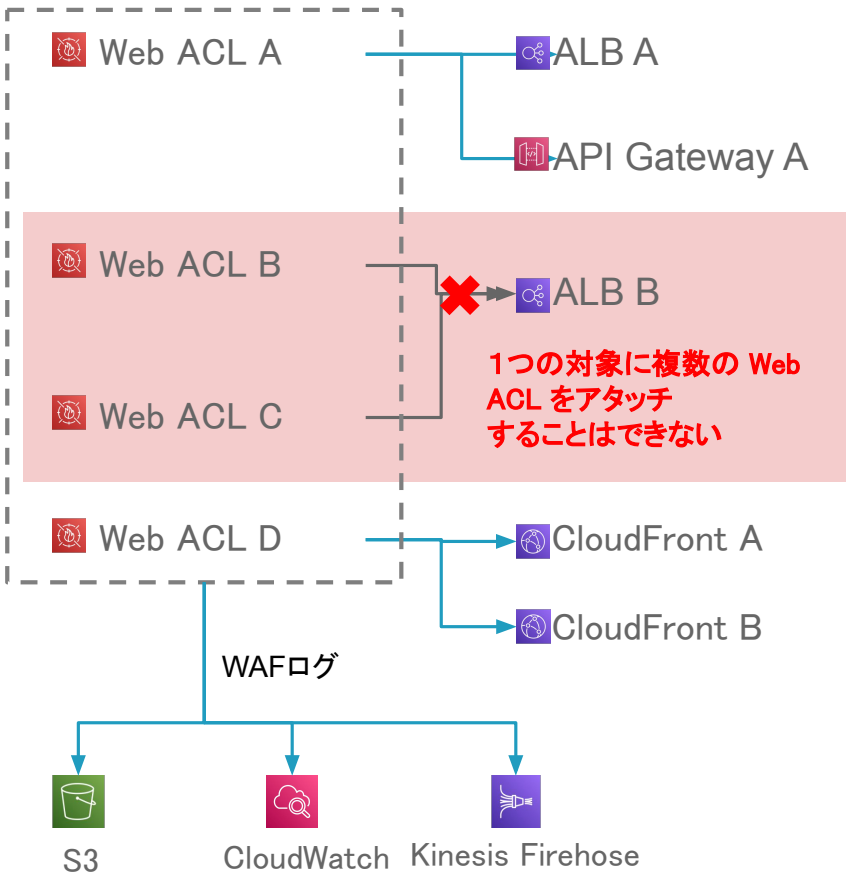
目次

- □ 会社紹介
- □ WAFについて
- ■ パブリッククラウドのManaged WAF
- □ 運用上の悩みポイント

各種パブリッククラウドのWAFと関連リソース

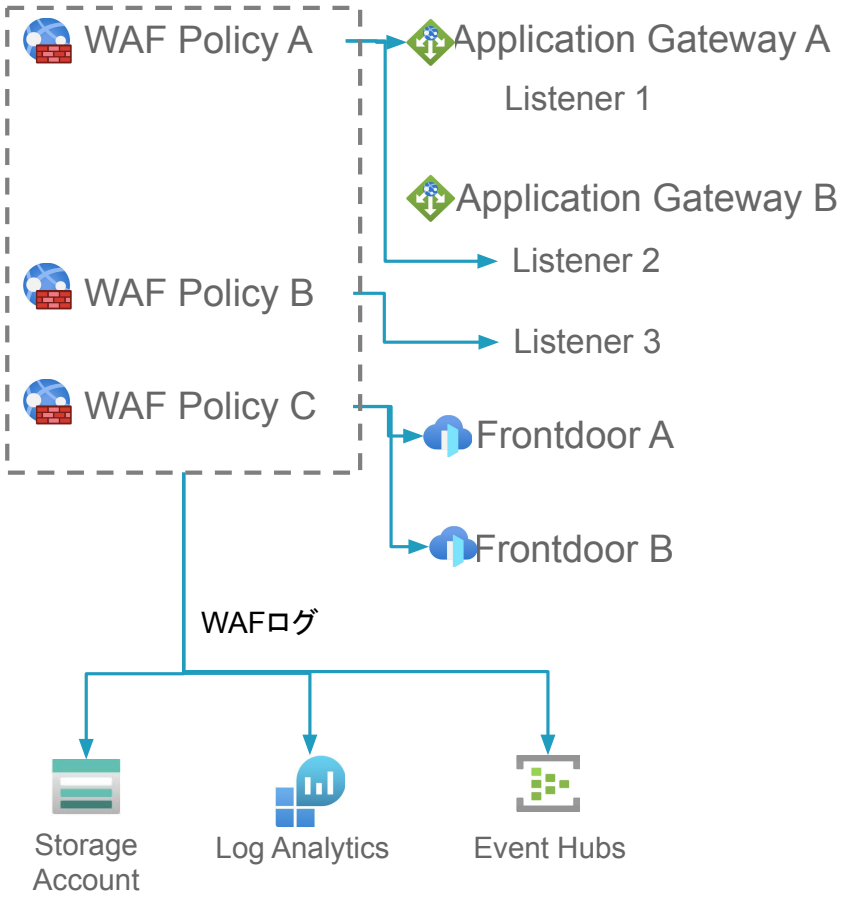
		AWS	Azure	GCP
DNS		Route53	Azure DNS	Cloud DNS
WAFに統合可能なサービス	**CDN/ エッジサービス**	Amazon CloudFront	Azure Front Door	Cloud CDN
	L7 ロードバランサ	Application Load Balancer	Azure Application Gateway	Cloud Load Balancing
	その他	Amazon API Gateway AWS AppSync Amazon Cognito user pool AWS App Runner AWS Verified Access instance		GKE Cloud Run App Engine Cloud Functions
WAF		**AWS WAF**	**Azure WAF**	**Cloud Armor**
DDoS		AWS Shield (Standard | Advanced)	Azure DDoS Protection (IP Protection | Network Protection)	Cloud Armor Standard Managed Protection Plus
ロギング		AWS Cloud Watch	Log Analytics	Cloud Logging

AWS WAFの特徴



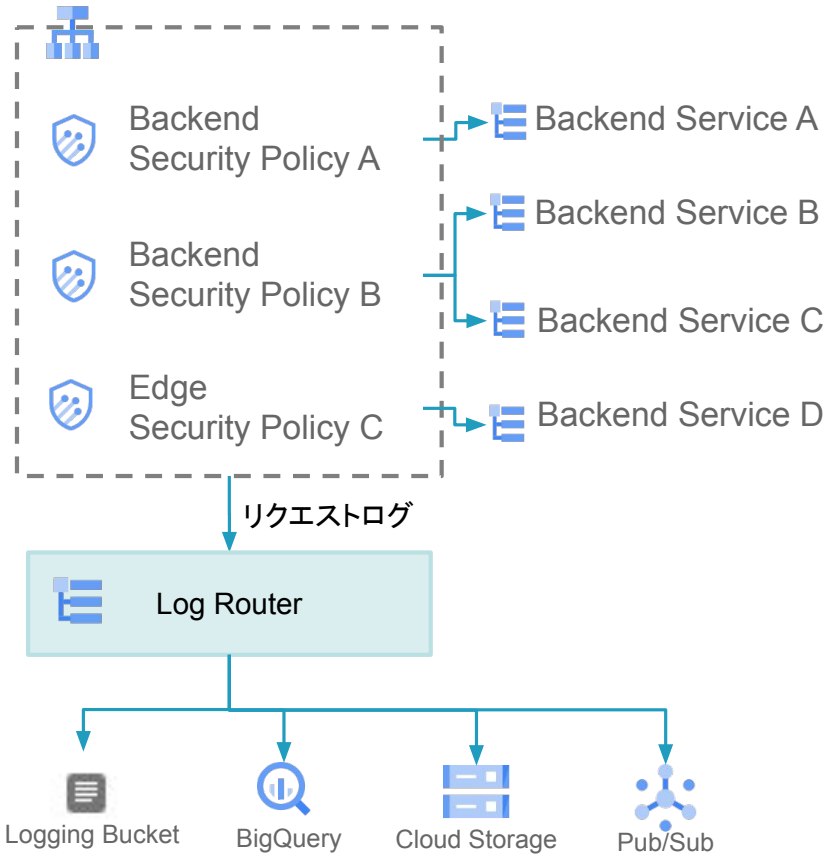
- WebACLというユニットで対象サイトを防御
- 複数のエンドポイントでWebACLは共有できる（同一のルール構成を複数サイトで共有）
- WCUという単位が各ルールごとに定義されており、これらが上限に達するまでルール投入できる
- WAFを通過したログはS3やCloud Watch、Kinesisといったサービスと連携することで解析やトリガーをしかけることが可能
- AWS自身が提供するManagedRuleの他、サードパーティベンダーが開発・保守しているManagedRuleもある

Azure WAFの特徴



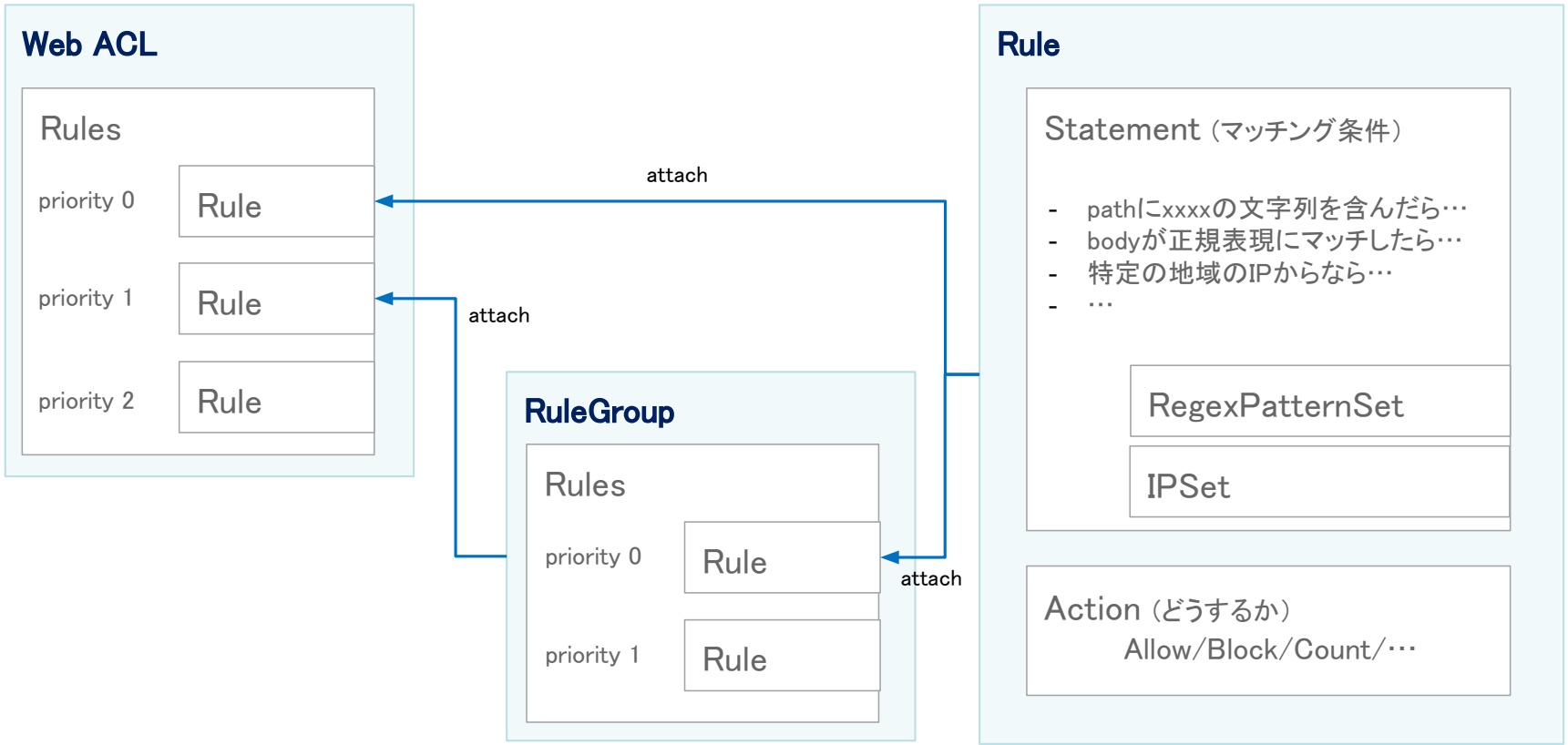
- WAF Policyというユニットを設定して対象のサイトを防御
- 複数のエンドポイントでWAF Policyは共有できる
  （同一のWAFシグネチャを複数サイトで共有）
- なおかつ、Application Gateway全体にアタッチ
  することもその配下のListener単位（さらにパスベースの
  ルールがあればその単位も）にアタッチすることもできる
- Core Rule Setをベースにした管理ルールを基本
  ルールとし、追加ルールやカスタムルールが追加できる
- Application GW向けとFront Door向けでWAFに
  設定できるルールや細かい条件などが異なる

Google Cloud Armor の特徴



- Load Balancingという大きなネットワークサービスの中の枠組み（デフォルトでEdgeレイヤーが組み込まれている）
- Security Policyというユニットを設定して対象のサイトを防御
- 複数のエンドポイントでPolicyは共有できる（同一のWAFシグネチャを複数サイトで共有）
- デフォルトのルールがあり、allow/denyを制御することはできるが、除外することはできない
- Adaptive Protectionといった機械学習モデルベースの防御機能もある

Web ACL関連リソース概念図



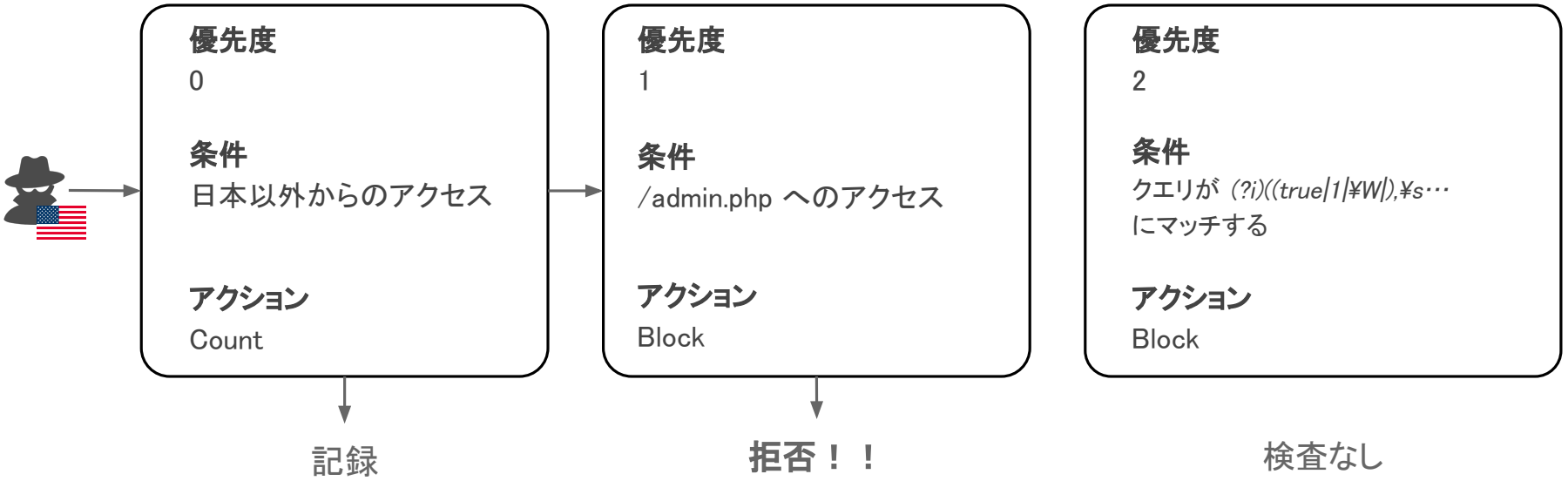
Statement詳細

内容	説明	対象
IP アドレス	CIDR 形式の IP アドレス。IP Set というリソースと関連付く。（後ほど説明）	ヘッダー - Single - All
文字列マッチ	文字列の完全一致、文頭文尾一致	Cookie - Single - All
正規表現	直接記述もしくはRegexPatternSet という正規表現をまとめたリソースと関連付く	クエリパラメータ - Single - All
レートベース	IPアドレス毎の5分間でのアクセス数で検知	ボディ JSON body
Geo ロケーション	AWS の判定によるアクセス元の国で検知 アクセス元のIPから地域レベルの情報もラベルとして付与されるように	URI パス
サイズ制約	対象のサイズで検知	メソッド
ラベル	ラベルが付与されたアクセスを検知	
SQLi, XSS	SQLインジェクション、クロスサイトスクリプティングの攻撃を検知	

Action詳細

内容	説明
Count	マッチしたものをカウント。後続のルールの判定処理を継続する
Allow	当該リクエストをサービスに転送。後続のルールの判定処理はそれ以上されない
Block	当該リクエストをブロックし、クライアントに応答する（デフォルトは403）。 レスポンス内容はカスタマイズ可能。後続のルールの判定処理はそれ以上されない
CAPTCHA	人からのアクセスであることを確認するためのCAPTCHAパズルを表示するよう、応答する。 CAPTCHAパズルが成功すると元のリクエストをサービスに転送。
Challenge	ブラウザからのアクセスであることを確認するためのJavascriptコードを返却するよう、応答する。 ブラウザからのアクセスであれば当該Javascriptコードが実行されることでChallengeに成功し、 サービスに転送。

ルール優先度



アクションが「Allow」「Block」であるルールにリクエストが該当した場合、そこで WAF の検査はストップする。
結果が「Count」の場合はログに書き出されるものの検査が止まらず、優先度に基づいて検査が続く。

ラベル機能

優先度
0

条件
日本以外からのアクセス

アクション
Count かつ
“foregin” ラベル付与

記録

優先度
1

条件
/admin.php へのアクセス　かつ “foreign” ラベル付き

アクション
Block

拒否！！

優先度
2

条件
クエリが *(?i)((true|1|¥W|),¥s…* にマッチする

アクション
Block

検査なし

ルールにラベルをつける機能ではなく、**アクセスに対して**ラベルを付与する機能。
付与されたラベルを元に後段のルールで条件を設計できる。
複数のルールにまたがる共通の前提条件として利用すると便利。

まとめ

- ✔ どのクラウドでもWAFはCDN/Edge向けのものとRegion向けのものとがあり、できることが異なる
- ✔ いずれもクラウドのネットワーク構成やサービスコンセプトが特徴として現れているが、基本要素は「**標準（Managed）ルールの提供**」「**ルールエンジン**」「**アクション**」「**統合できるサービス**」「**ロギング**」といったもので構成されている
- ✔ （当然ながら）いずれのWAFもデフォルトルールの内部詳細は非公開。どうして検知したのか（リクエストのどこに問題があったのか）については明確に判断することは難しい
- ✔ ルールエンジンやアクションについては、標準的なものはいずれのWAFでも提供されている が、AWSについてはかなり細かい制御にも対応している（ただし、WCUというコストの考え方が複雑）
- ✔ Azure, GCPについてはネットワークレベルでEdgeを組み込んだ設計であったり、単純なルールベースではないモデルベースでの防御機能がある（AWSにも部分的にはある）が、ルールベースよりもさらにどうして検知したのかの判断が難しい

目次

- □ 会社紹介
- □ WAFについて
- □ パブリッククラウドのManaged WAF
- ■ 運用上の悩みポイント

WAF運用上の課題

そもそもの導入に際して、どのようなルールをどれくらい入れればいいか分からない

- クラウド側が提供しているルールで十分なのか
- 提供されているものは全部入れるべきなのか
- 自前でルールを入れるにはセキュリティの知識に加えて、Cloud WAF特有の仕様や制限にも精通しておく必要がある

誤検知が発生した際、どのように対応すればよいか分からない

- セキュリティレベルを落とさずに誤検知対応するのは難しい
- 大手SOCサービスは高額だったり、日本語での対応に難があることも

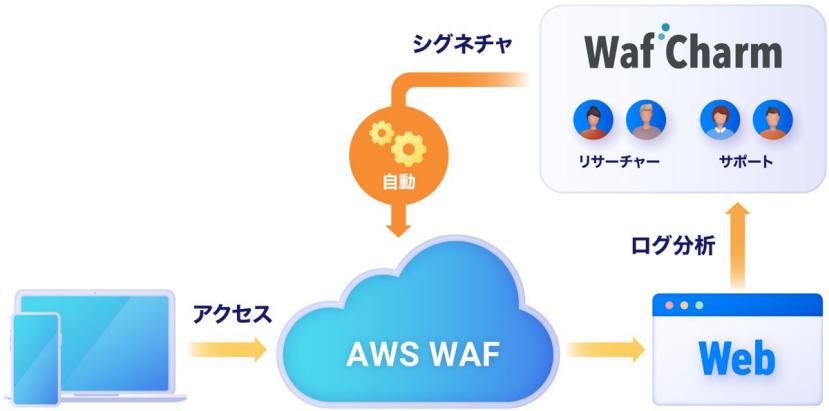
新たに登場する個別の脆弱性への対応負担が大きい

- 脆弱性を追跡し、脅威を判断・対策するのは専門知識を必要とするし負担も大きく、専任者をおかなければ難しい

WAF自動運用サービス「WafCharm」がお悩みを解決します

WafCharm

「WafCharm」は、クラウドWAFの自動運用サービスです。
アクセスログをもとに新たに発見した攻撃や新規脆弱性に対応するルール（シグネチャ）をWafCharmが自動で作成・更新するので、AWS WAFとセットで利用することで専任のセキュリティエンジニアを必要とすることなくAWS WAFの運用を円滑に行うことができます。あなたをWAF運用から解放します。



強力な防御性能

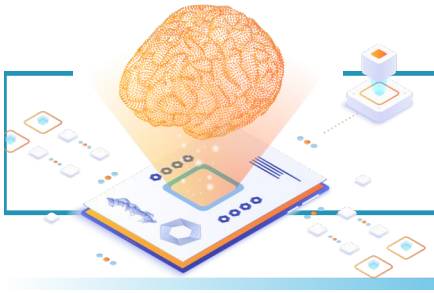
お客様の環境に最適なルール(シグネチャ)を作成し自動適用します。
新たな脆弱性にもWafCharmが対応するので安心。

お客様ごとに最適な防御をご提供

- 最適シグネチャを自動選択
- シグネチャのカスタマイズや新規作成可能
- 最新の脆弱性にも対応。新規シグネチャを迅速に作成に、すぐに適用。

数百ものシグネチャでより強力に

- 設定しているルールでは漏れる可能性も
- 設定したルール外の攻撃にも数百ものシグネチャで再マッチング
- 再マッチングで攻撃認定したものはBlacklistに自動適用



面倒なルール作成はWafCharmにおまかせ

導入から運用まで安心のサポート

導入から利用開始後もサポート面が充実しています。
24時間365日の日本語サポートがあるので安心。

導入も運用も楽に

- 専用機器設置やDNS切り替えなど
  不要で導入がスムーズ
- WafCharm管理画面とAWSマネージドコン
  ソールからの設定で即時利用可能

なにか困ったらサポートへ

- 日本語での24時間365日技術サポート
- 誤検知対応、ルールの入れ替え、カスタマ
  イズにも柔軟に対応

導入もスムーズ、運用も手放しでOK

改ざん検知機能を搭載

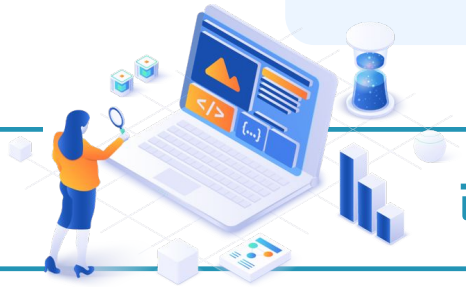
Webサイトが改ざんされていないかを毎日チェック
万が一の際には通知でいち早くお知らせします。

高性能検知エンジンで毎日チェック

- 登録されたWebサイト（FQDN）を日々継続的にチェックし、改ざんの可能性がないかを確認

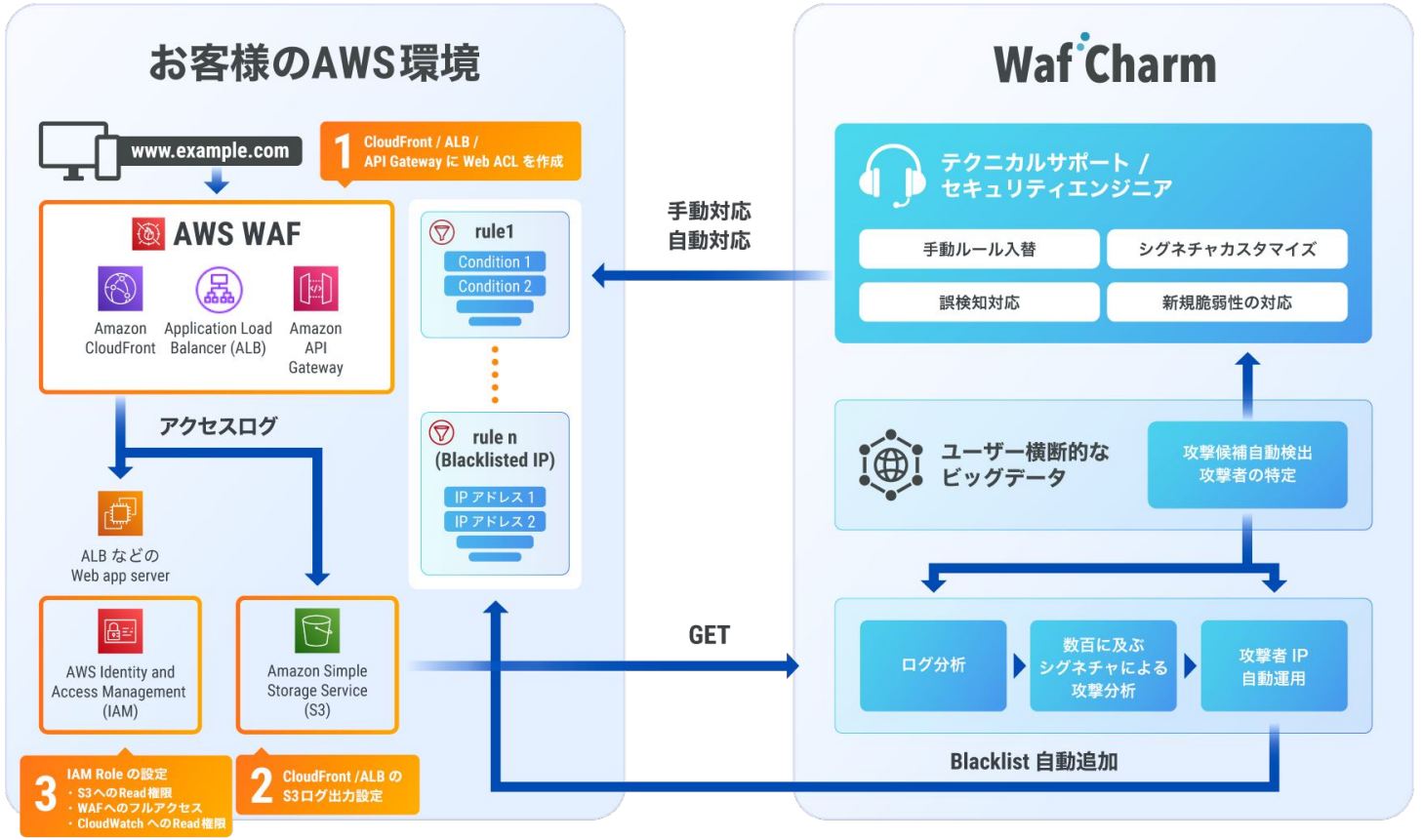
管理者へ即座にメール通知

- Webサイトの改ざんを検知した場合、登録した管理者のメールアドレスにすぐに通知
- 通知メールの内容：監視対象のFQDN、該当URL、検出された内容



改ざんにも対応、万が一もすぐにわかる

WafCharmのシステムアーキテクチャ

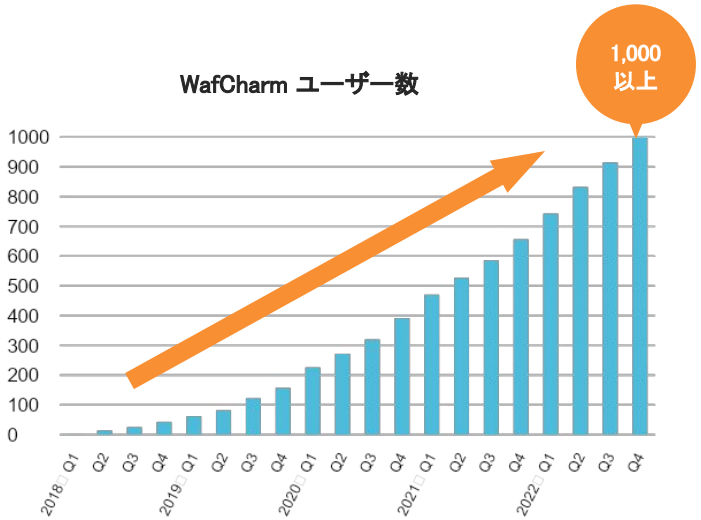


導入実績

多くのお客様に導入いただき、AWS WAFの自動運用サービス導入ユーザー数国内No.1を達成しております。

AWS WAF 自動運用サービス
導入ユーザー数
国内No.1

日本マーケティングリサーチ機構調べ 調査概要：2020年7月期_実績調査

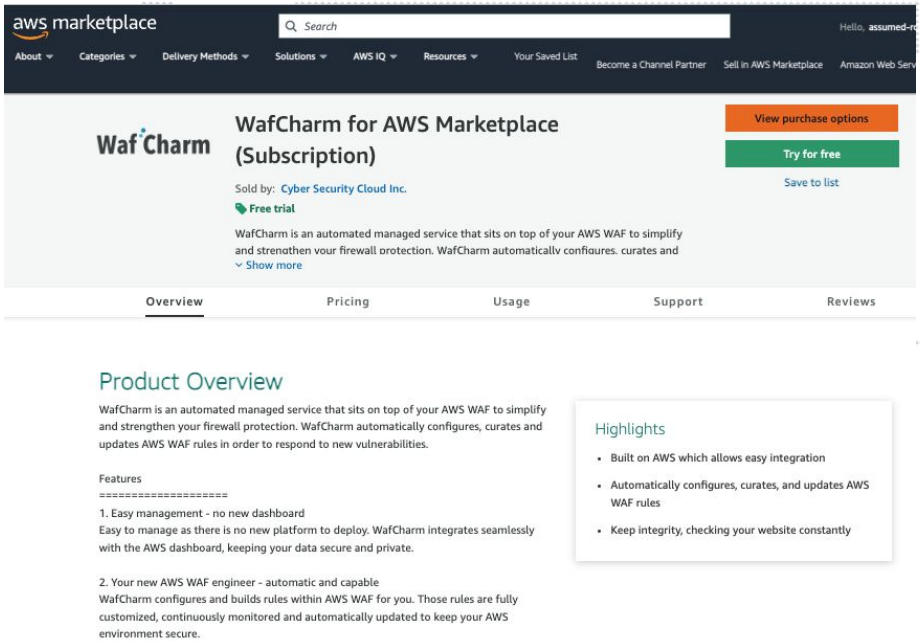


導入企業様
一例

freee coconala ほぼ日 CELLS
note マイナビ Classi KRAY

AWS Marketplaceでの販売も開始

- ✔ 昨年11月よりAWS Marketplaceで全世界向けにWafCharmの販売開始
- ✔ 通常のWafCharmの機能にWeb改竄検知機能を追加し、いくつかの機能でもパワーアップ
- ✔ 今年1月よりアイレット様にてCPPOでの提供も開始



https://aws.amazon.com/marketplace/pp/prodview-crrflizdnl6pw

https://www.iret.co.jp/news/20230131.html

総括

- ✔ WAFとはWebアプリケーション層の防御機構であり、シグネチャと呼ばれるルールを投入して使用する
- ✔ Webアプリケーション・セキュリティにおける代表的なリスクはOWASP TOP10が知られており、この中のいくつかについてWAFが主たる防御機能を果たす

- ✔ AWS/Azure/GCPといったクラウドベンダーが提供しているWAFがあり、インフラレイヤーはクラウド側によって管理されており、基本的にはルールを投入して対象サイトにアタッチして使用する。
- ✔ クラウドサービスとしてのコンセプトや特徴はあるものの、基本要素は
  「**標準（Managed）ルールの提供**」「**ルールエンジン**」「**アクション**」「**統合できるサービス**」
  「**ロギング**」といったもので構成されている

- ✔ これらを使用することで、自前でWAFを構築・運用する苦労が大幅に減らせるものの、中に投入するルールの部分については「**専門知識が必要**」「**誤検知対応や新規脆弱性対応が負担**」といった課題に関しては依然として残る

- ✔ これらのクラウドManaged WAFに「**WafCharm**」を併用することで、クラウドの可用性や信頼性を享受しつつ、運用も楽にすることができる

ご清聴ありがとうございました。