Log4j 問題で見えてきた、マルチクラウド時代に必要な Web セキュリティ対策

萩原 健矢

ヴイエムウェア株式会社

ネットワーク&セキュリティ技術本部
シニア スペシャリストエンジニア

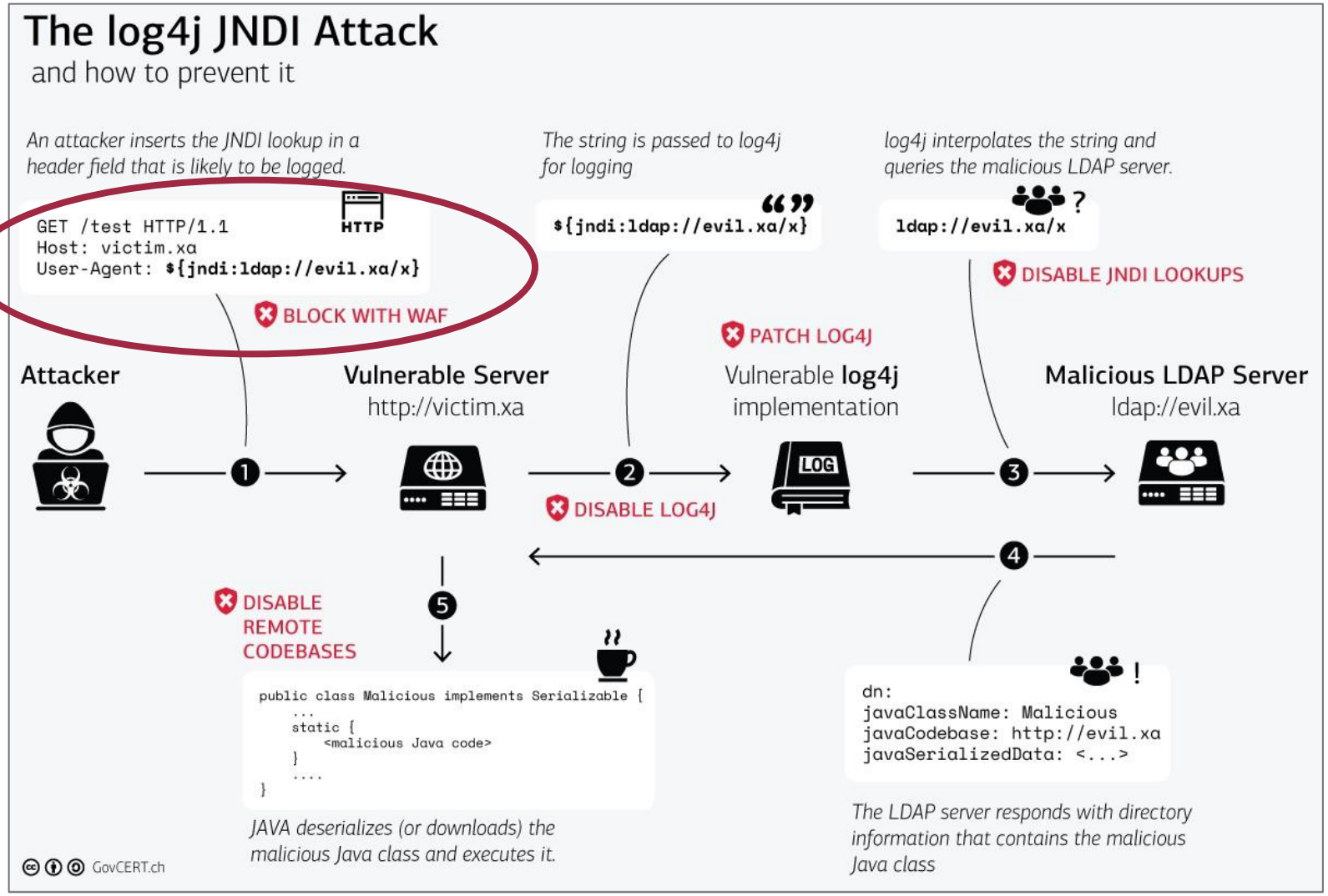
vmware

Apache Log4j 問題の振り返り

- 共通脆弱性評価システムCVSSスコアは最大の「10.0」が付与される
- 多くのJavaシステムで利用されており、影響範囲が広い
- オンプレミス・クラウド問わず、あらゆる場所で問題になった
- 脆弱性を悪用するのが非常に簡単である
- 脆弱性を開示してから数時間後にはスキャニングや攻撃が観測される
- 攻撃の文字列に記載されるプロトコルの種類や難読化のパターンが増えた
- 攻撃の起点となる国を隠蔽するために「Tor」が使用される

Apache Log4j の脆弱性対策

WAF は第一防御線として、アプリケーションの脆弱性を悪用した攻撃を防ぐことが可能



主な対策

- ✓ WAF で攻撃を遮断
- ✓ LOG4j を無効化
- ✓ LOG4j にパッチ適用
- ✓ JNDI LOOKUPS を無効化
- ✓ REMOTE CODEBASES を無効化

出典：https://www.govcert.ch/blog/zero-day-exploit-targeting-popular-java-library-log4j/

脆弱性が発見された場合の対応

脆弱性を根本的に取り除くまでの暫定対策としてWAFの導入は有効

根本対策

自社で利用している製品・サービスへの影響について調査

パッチ適用やバージョンアップした場合の互換性調査・動作検証

※互換性の問題を解消するために、アプリケーションコードの改修や機能の停止が必要なケースもある

パッチ適用やバージョンアップの実施

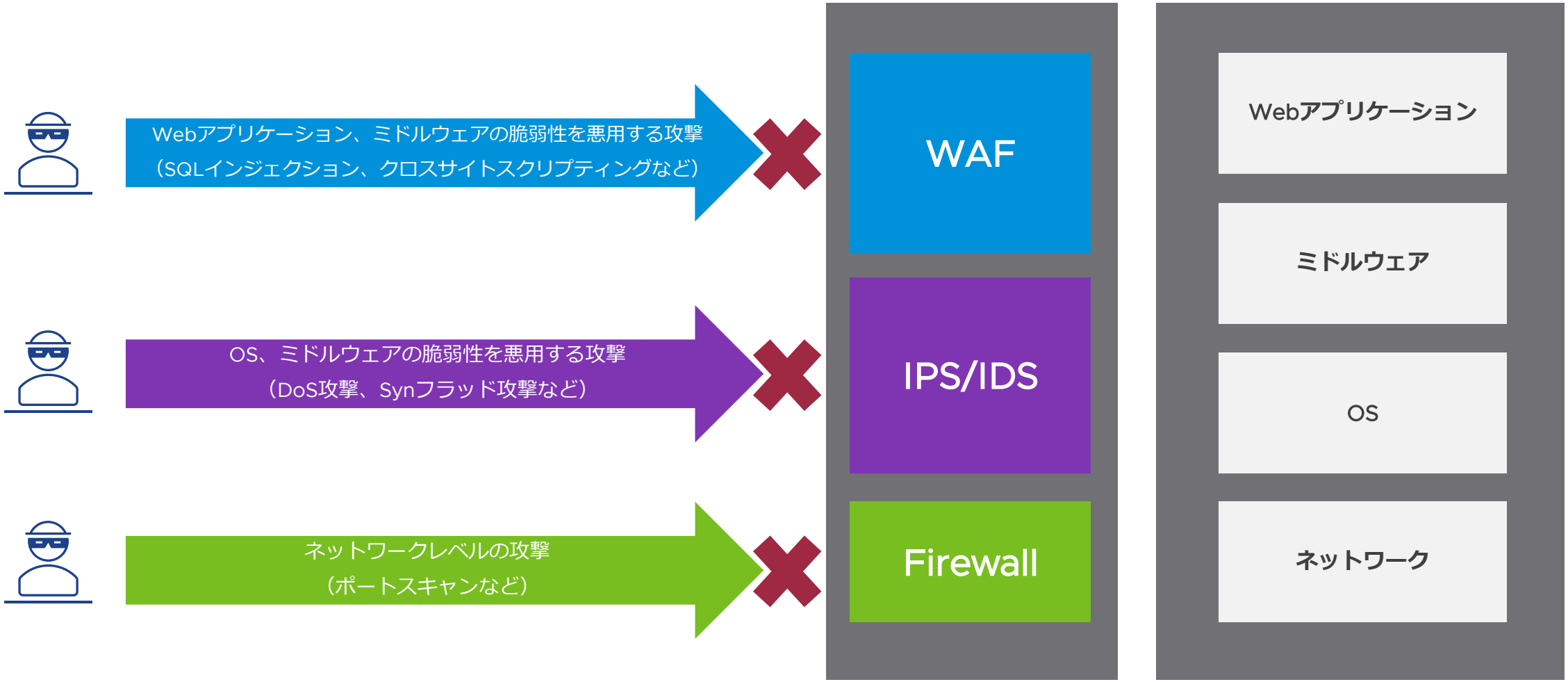
この間に攻撃される可能性がある

暫定対策

WAF を活用することで、根本対策が完了するまでの時間を稼ぐことができる

WAF (Web Application Firewall)

従来の FW や IPS では防げない、Web アプリケーションの脆弱性を悪用する攻撃を緩和



WAFの防御方法

２つのセキュリティモデルを組み合わせることで、両方のメリット・デメリットを相互に補完

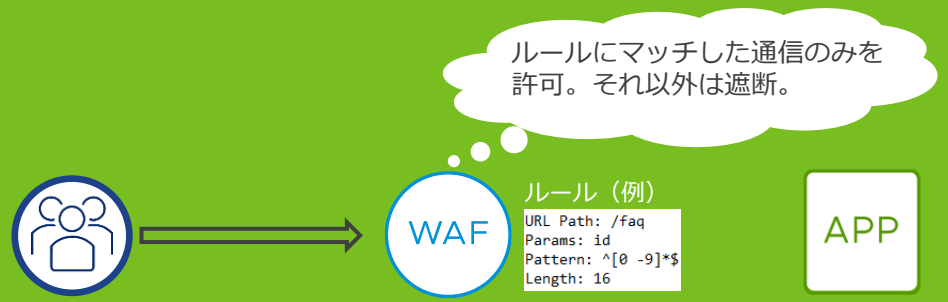
ポジティブ セキュリティ モデル

管理者が事前に定義した“正しい通信”のみを許可し、それ以外の通信は遮断する考え方

たとえば、入力パラメータごとに、取り得る値の範囲や内容を正規表現でルールとして定義する

メリット：ゼロデイ攻撃にも有効、安全性が高い

デメリット：ルールの設定が煩雑、使える場所が限定される



ネガティブ セキュリティ モデル

シグネチャと呼ばれる既知の攻撃パターンと照らし合わせ、該当するものを”不正な通信”として遮断する考え方

SQLインジェクションやXSSなどの攻撃パターンに焦点を当てたシグネチャとアプリケーションの脆弱性を突く攻撃パターンのシグネチャが存在する

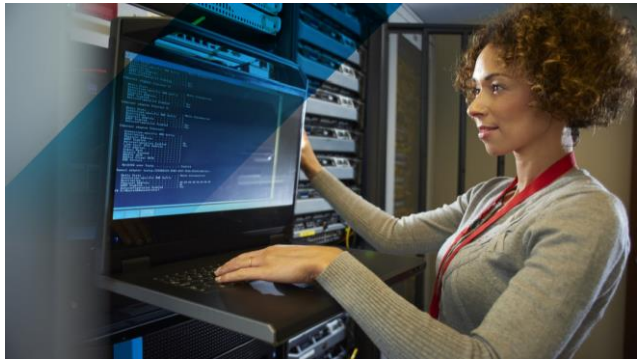
メリット：ルールの設定は比較的容易、全ての場所で使える

デメリット：既知の攻撃のみ対応、誤検知が多い、安全性はシグ ネチャに依存

WAFの課題

常に安全性向上と品質改善に取り組む必要があり、管理者の負担は大きい

WAF の設定が難しい



- ルールの設定には、アプリの構造やセキュリティの専門的な知識が必要
- 自社のサービスに合わせて、検査対象や遮断対象のルール調整が困難
- アプリの仕様変更や新しい脆弱性が発見される度にルールの見直しが必要

運用の負荷が高い

- 膨大なログの中に誤検知や検知漏れが生じていないかの確認が必要
- 攻撃の状況を体系的に把握するのが難しく、予防措置的な対策が困難
- シグネチャ更新やルール修正の際は、十分な動作検証が必要

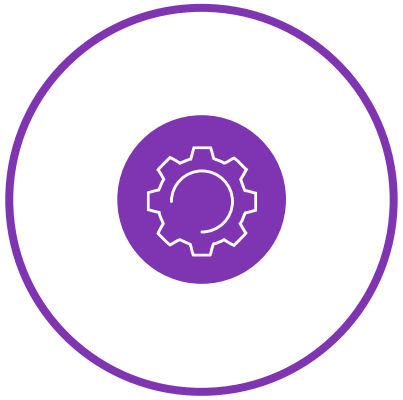
パフォーマンスが落ちる

- WAF 機能を有効化することにより、機器やサーバの処理性能が低下
- WAF サービス経由のアクセスによってレスポンスタイムが悪化
- システムに大きな変更や通信影響を伴わなずに性能拡張が困難

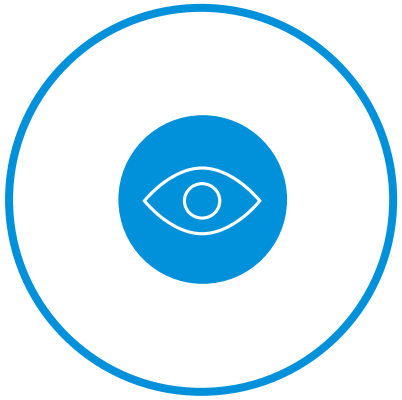
WAFの選定で重要なポイント

ハイブリッド・マルチクラウド環境に分散したアプリケーションを包括的に保護する必要がある

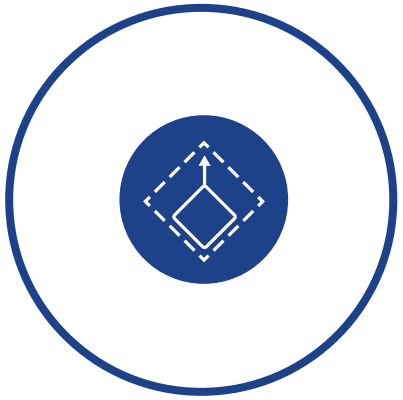
マルチクラウドで実現



新たな脅威に迅速に対応できる設定の自動化と簡略化



誤検知や検知漏れを削減できるログの分析・可視化



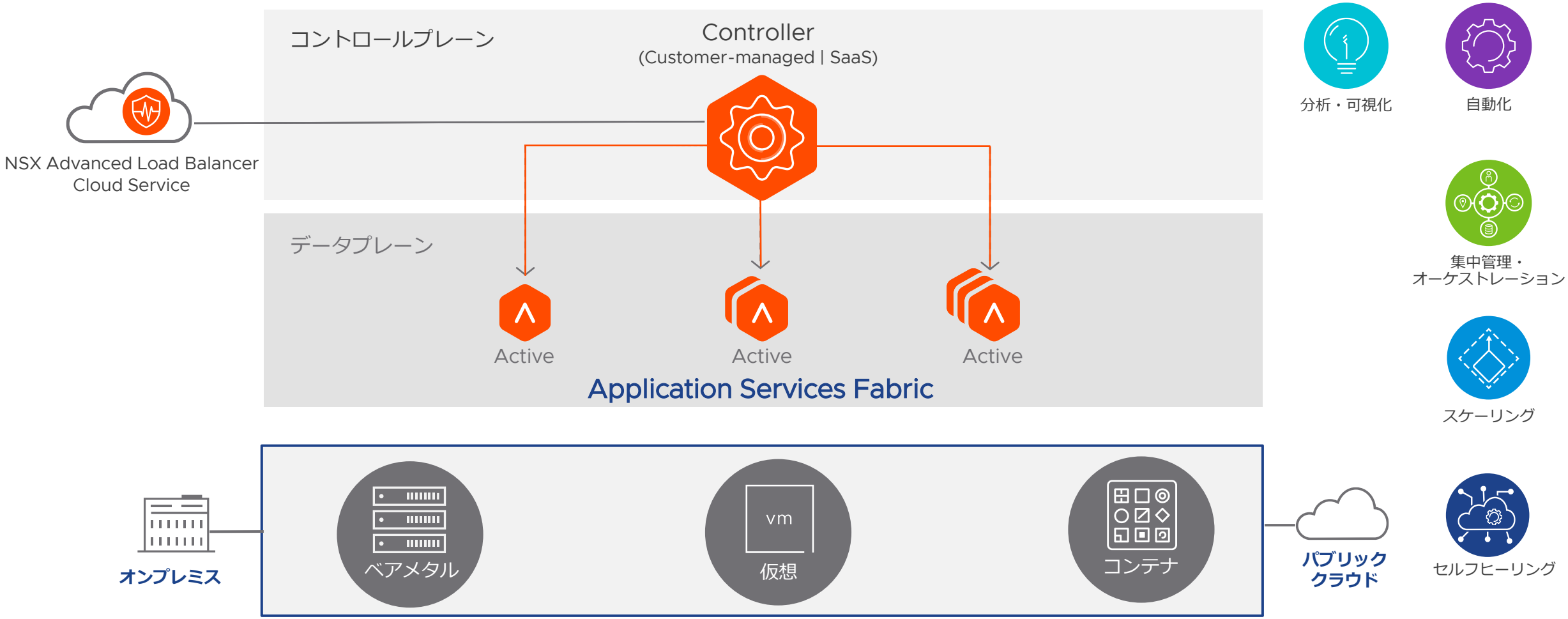
性能問題を克服できるスケールアウト型の拡張性

vmware NSX-T, aws, vmware, Azure, Kubernetes, Google Cloud Platform, OpenShift, openstack, VMware Tanzu, Linux

NSX Advanced Load Balancer WAF の概要

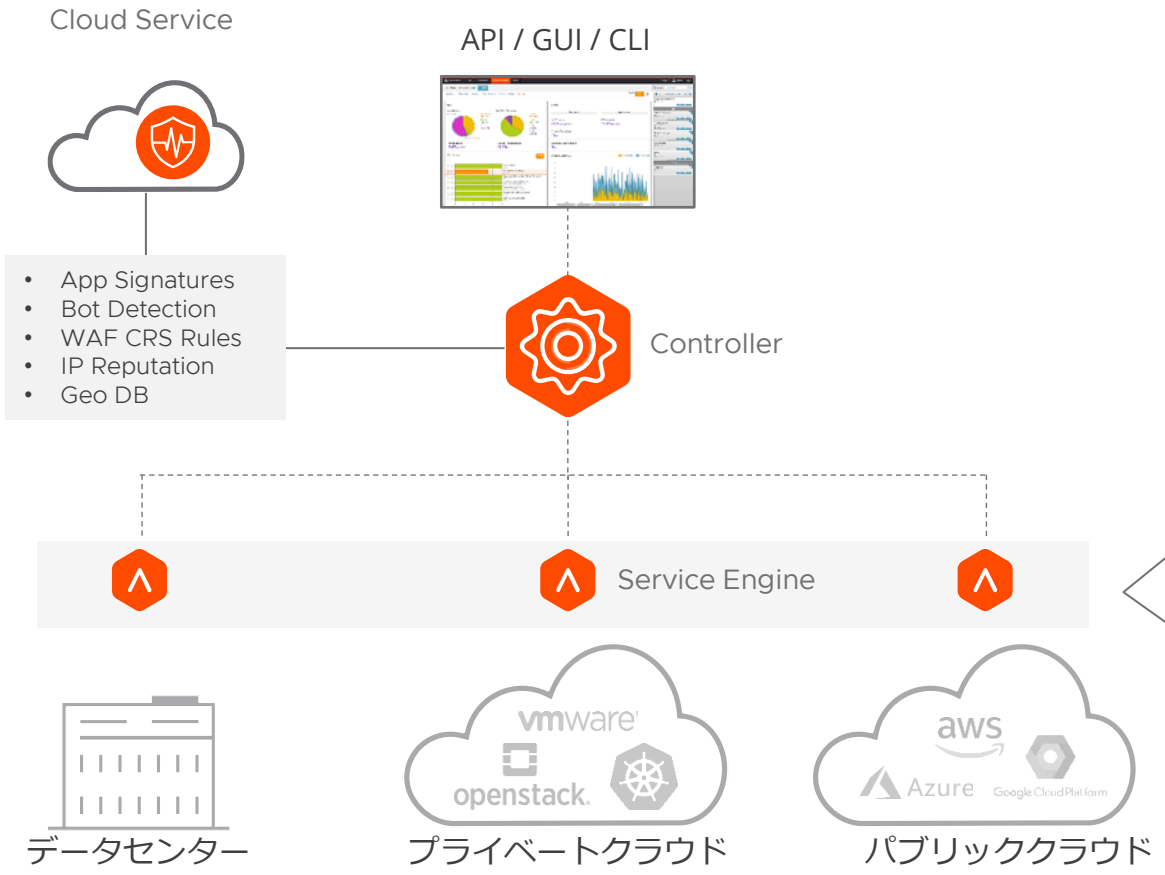
VMware NSX® Advanced Load Balancer™

マルチクラウド対応の完全ソフトウェア型ロードバランサー + WAF



NSX Advanced Load Balancer のセキュリティ ポートフォリオ

Web アプリケーションを保護するために包括的なセキュリティ機能を提供



Web Application Firewall (WAF)
OWASP Top 10 protection, Attack Analytics

ボット管理

アプリケーション レート制限
Control and restrict by application or tenants

DDoS 防御
DDoS detection and mitigation with elastic scaling

マルウェアプロテクション
via ICAP

ユーザー認証
SAML SP, HTTP Basic Auth, JWT, oAuth/OIDC

L7 ファイアウォール ポリシー
Content (URI) based security rules

L3/4 ファイアウォール ポリシー
IP-Port based Security Rules

暗号化
SSL/TLS

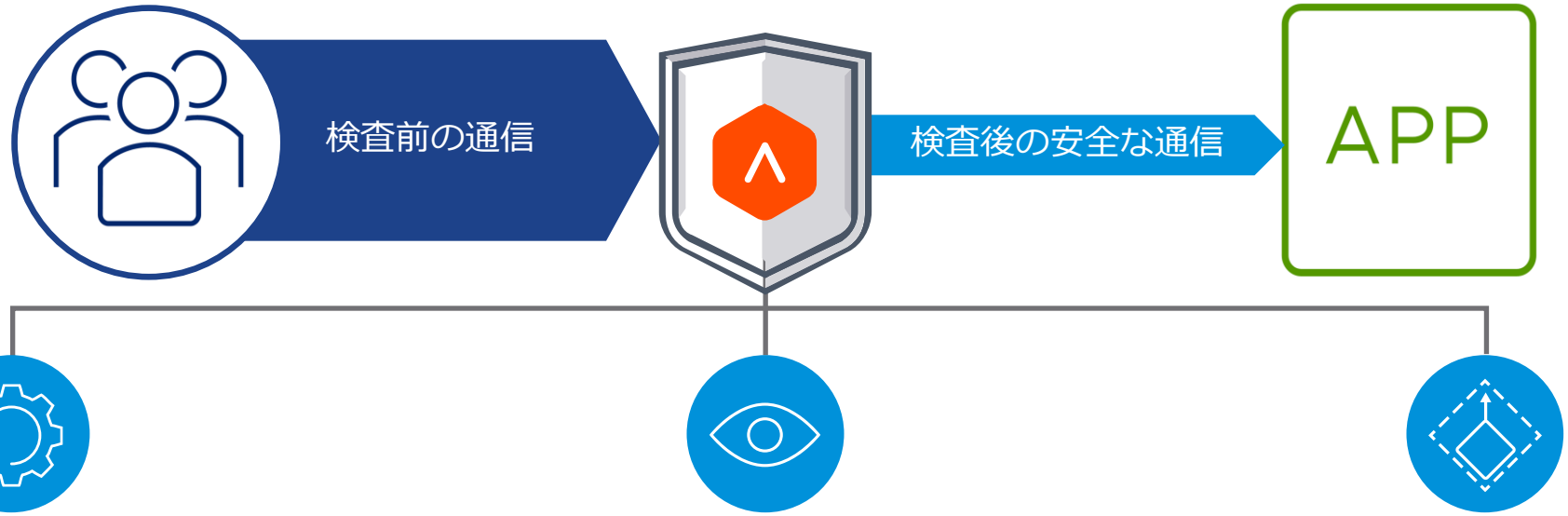
IPレピュテーション

セキュリティ可視化

Security score
Attack insights
SSL Insights
WAF analytics

NSX Advanced Load Balancer が提供する WAF の特徴

WAF の設定と運用を簡素化し、高い性能を維持するための仕組みを提供



WAF の設定を簡略化

- WAF のラーニング機能を活用することで、煩雑なルールの設定を自動化
- アプリケーションの仕様変更に伴う通信の変化を検知して、ルールを自動調整
- アプリケーションの脆弱性や攻撃パターンのシグネチャを自動更新

高度な分析・可視化を提供

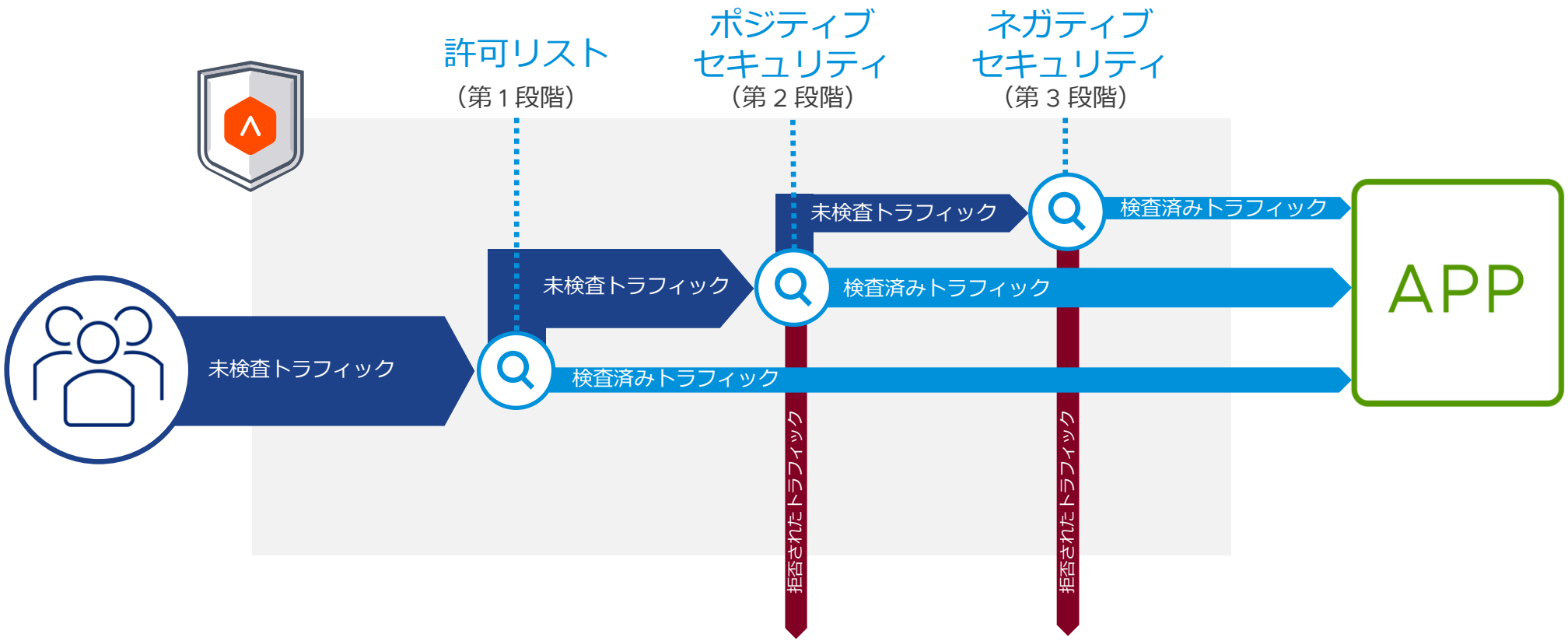
- 攻撃元の IP アドレスや攻撃の種類、攻撃された時間帯などを体系的に可視化
- 個々のトランザクション単位で検知・遮断・許可したログの詳細を可視化
- 誤検知や検知漏れのログを抽出して、ワンクリックでルールを修正可能

高性能・高拡張性を実現

- WAF の検査フローを効率化することで、検査に伴う処理負荷を軽減
- 自動またはオンデマンドでスケールアウトすることで、迅速かつ柔軟に性能を拡張
- L4-L7通信を高速処理するLBのエンジンを最大限に有効活用

WAF セキュリティ パイプライン

正常なトラフィックを段階的に許可していくことで、性能劣化と誤検知を最小限に抑えることが可能



許可リスト

- WAF による検査が不要な通信条件を定義し、合致する通信は WAF を迂回

ポジティブ セキュリティ

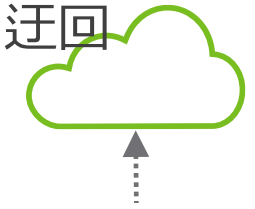
- 管理者が事前に定義したルールに合致する通信のみを許可

ネガティブ セキュリティ

- シグネチャに合致する通信を遮断
  - Core Rule Set （攻撃パターンのシグネチャ）
  - App Rules（アプリ脆弱性のシグネチャ）

WAF 第 1 段階の検査：許可リスト

WAFで検査不要な通信の条件を定義し、該当する通信はWAFを迂回



VMware NSX®
Advanced Load Balancer™
Cloud Service

IP レピュテーション
ボット管理
シグネチャ

WAF を迂回する通信の例

- 静的コンテンツの URI パス宛の通信
- DAST スキャナや管理ネットワークの IP からの通信
- 通信の 10% を WAF で処理し、残りは迂回

許可リスト

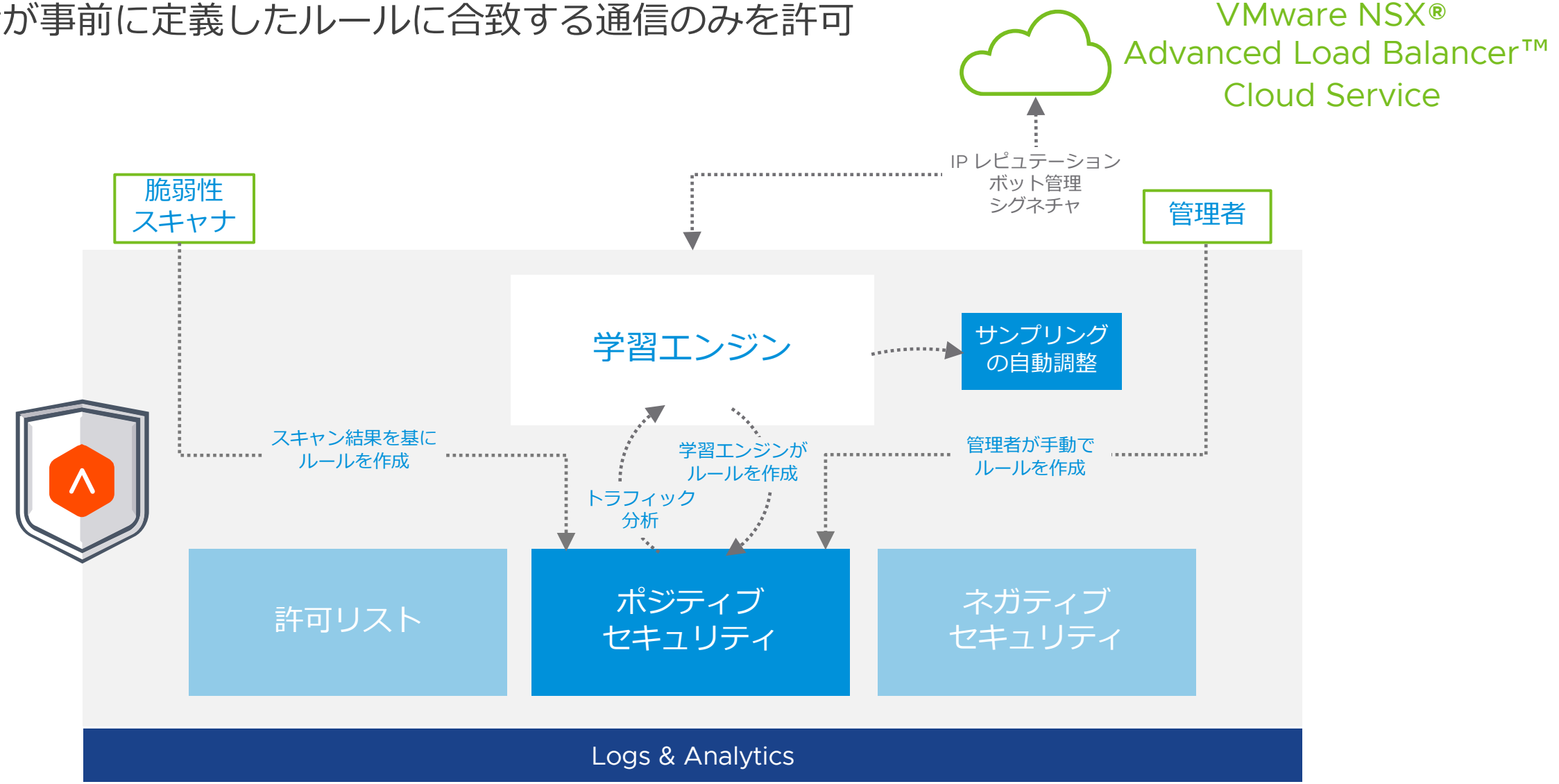
ポジティブ
セキュリティ

ネガティブ
セキュリティ

Logs & Analytics

WAF 第 2 段階の検査：ポジティブ セキュリティ

管理者が事前に定義したルールに合致する通信のみを許可



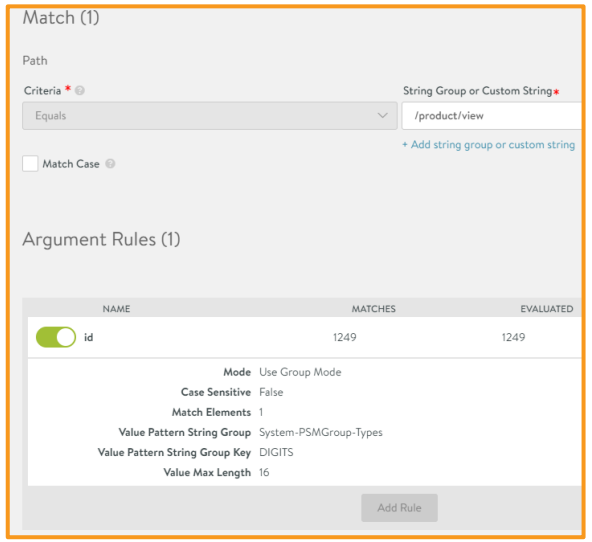
学習エンジン（ラーニング機能）の動作

正常な通信のデータを学習をすることで、ポジティブセキュリティのルールを自動的に作成

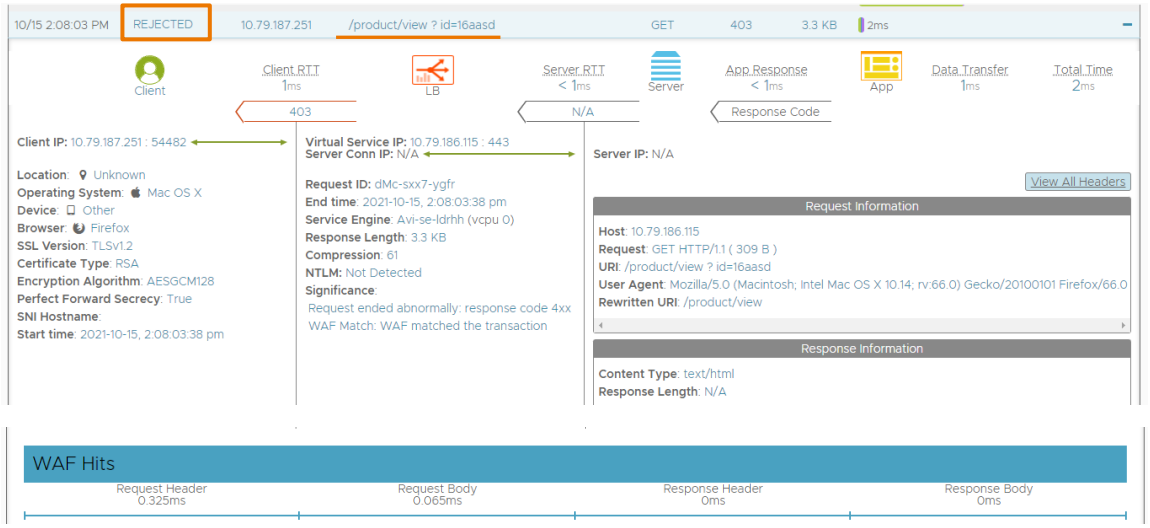
学習したデータを基にルール作成を開始

```
----- Promoted Rules -----
/wishlist/
   name
    Pattern: WORD
    Length : ^[0-9A-Za-z._-]*$
   type
    Pattern: WORD
    Length : ^[0-9A-Za-z._-]*$
   search
    Pattern: WORD
    Length : ^[0-9A-Za-z._-]*$
   id
    Pattern: FLAG
    Length : ^$
/product/view
   id
    Pattern: DIGITS
    Length : ^[0-9]*$
/category/view
   id
    Pattern: DIGITS
    Length : ^[0-9]*$
/faq
   userQuestion
    Pattern: FLAG
    Length : ^$
```

URI パス /product/view の id パラメータに数字 16 桁までの入力を許可するルールが自動的に作成された



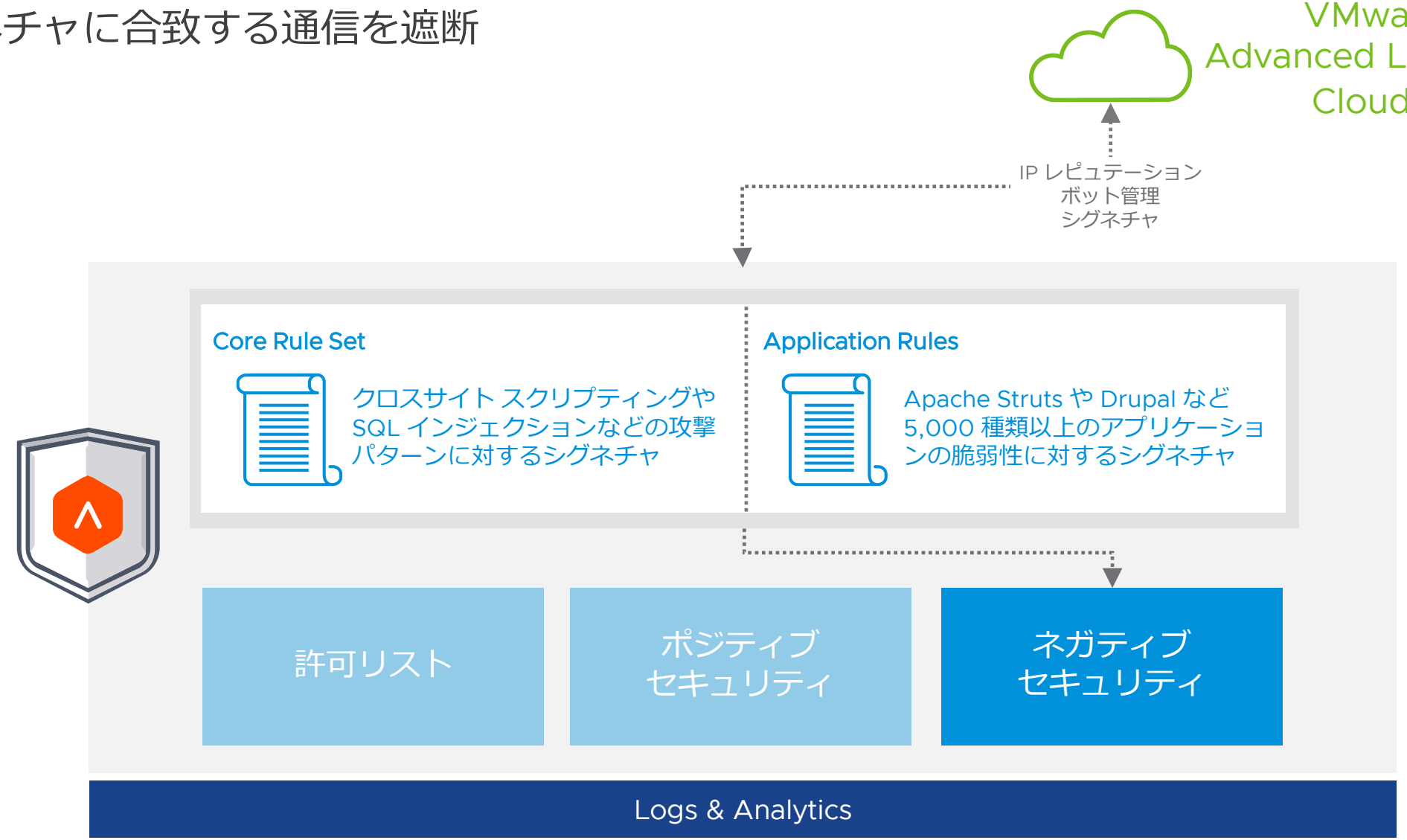
https://10.79.186.115/product/view?id=16aasd にアクセスした場合、ルールに違反しているため、通信は REJECTED（遮断）されることを確認



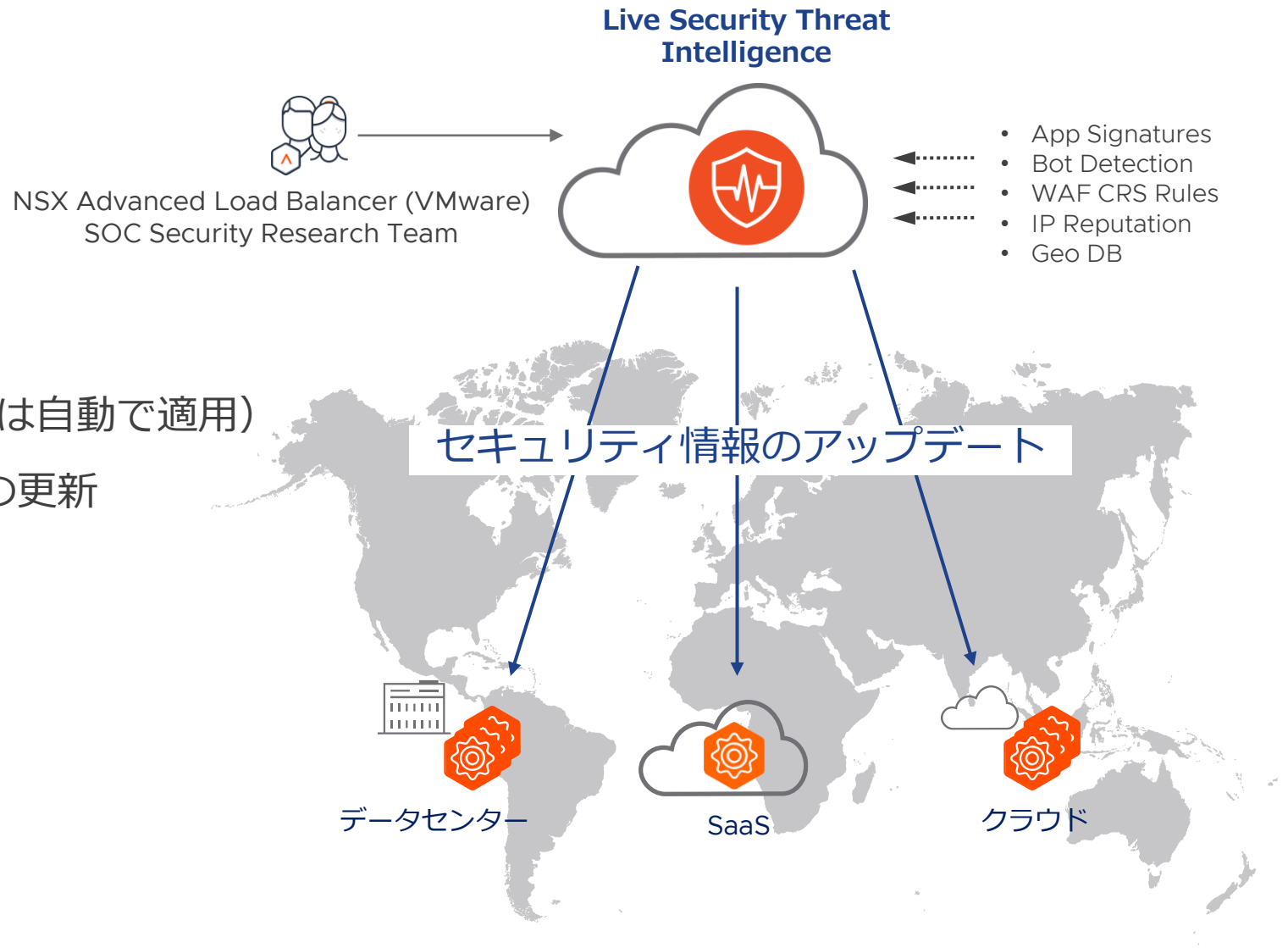
攻撃を検知したペイロード部分が表示されるため、ルールのどの部分に違反があったのかが明確

WAF 第 3 段階の検査：ネガティブ セキュリティ

シグネチャに合致する通信を遮断



Live Security Threat Intelligence
お客様の WAF の運用を支援するクラウドサービス



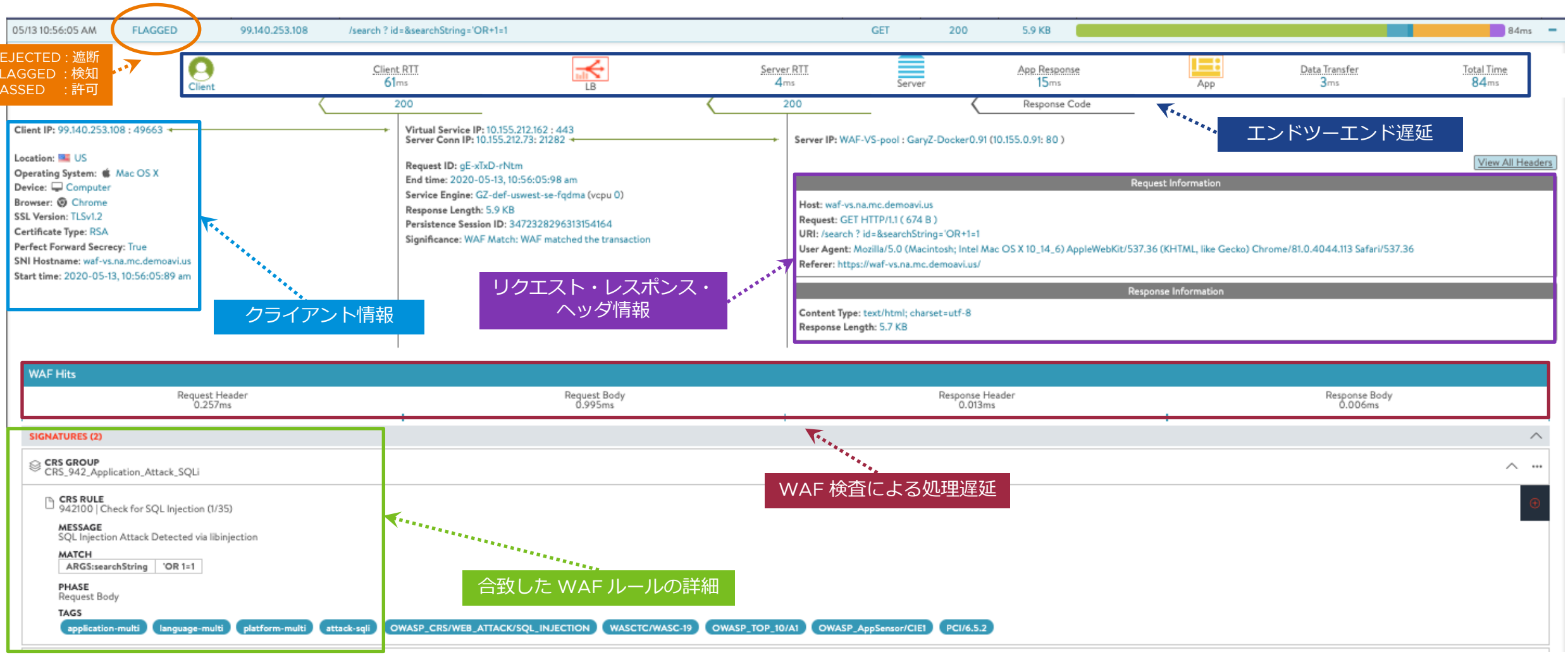
お客様の運用システムとの連携

- WAF シグネチャの更新（手動または自動で適用）
- IP レピュテーションデータベースの更新
- ボット管理データベースの更新

誤検知や検知漏れを削減できる
ログの分析・可視化

WAF トランザクション ログの可視化

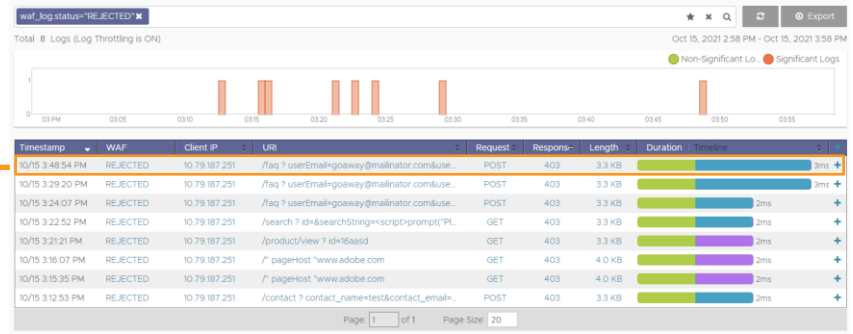
個々のリクエスト・レスポンス単位でユーザー体感速度や WAF の処理内容を把握することが可能



誤検知ログへの対応フロー

遮断したログを精査し、偽陽性である通信を選別して、ルールを修正

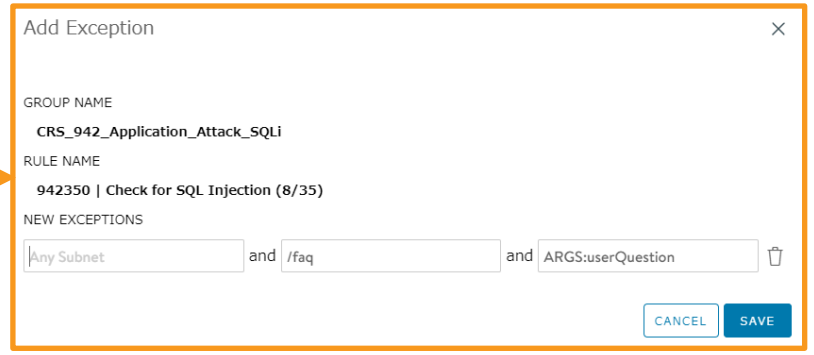
①REJECTED（遮断）したログの抽出



②誤検知したルールの詳細を確認

クリック

③例外ルールを設定



④例外ルールが設定されていることを確認

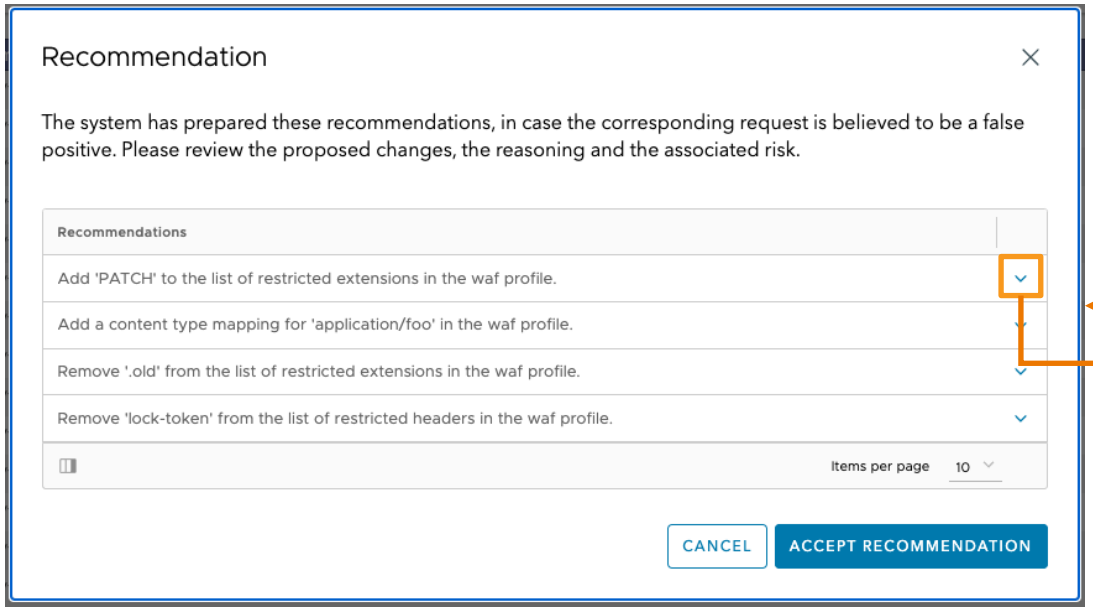
誤検知の是正を支援する Recommendation 機能

誤検知の可能性およびルールの修正案を表示し、必要に応じてワンクリックでルールを修正可能

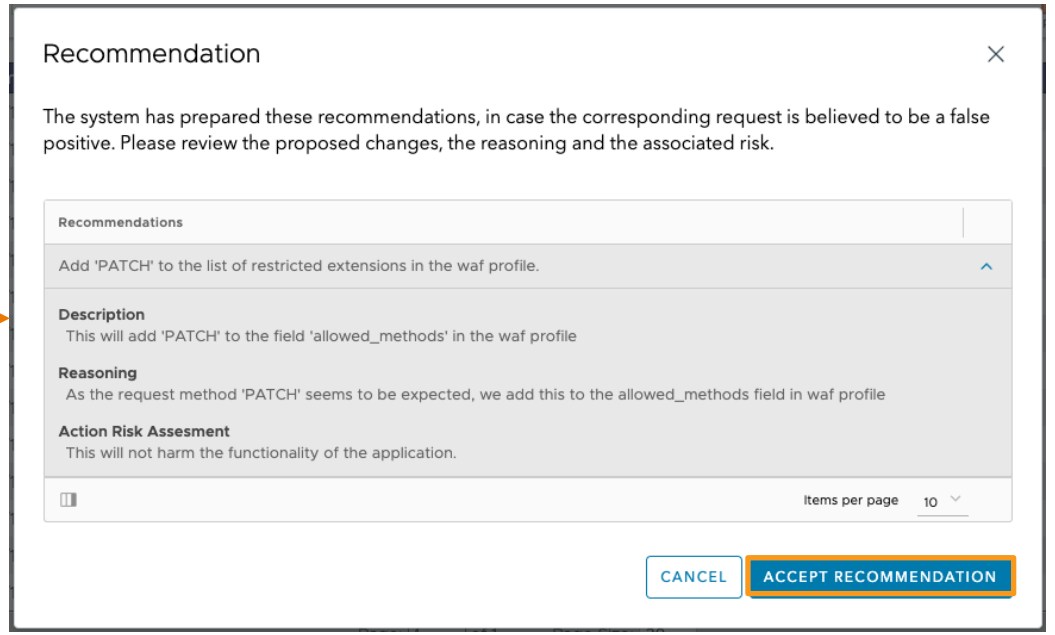
① REJECTED（遮断）したログの Recommendation をクリック

Timestamp	WAF	Client IP	URI	Request	Response	Length	Duration
12/16 9:03:51 PM	REJECTED	100.64.19.20	/upload...	POST	403	4.0 KB	2ms
12/16 9:03:49 PM	REJECTED	100.64.19.20	/	GET	403	4.0 KB	2ms
12/16 9:03:47 PM	REJECTED	100.64.19.20	/foo.old	GET	403	4.0 KB	2ms

② Recommendationの一覧を確認



③ Recommendationの詳細を確認し、誤検知であればルールの修正を実行



検知漏れログへの対応フロー

許可したログを分析し、偽陰性の可能性がある不審な通信を選別して、ルールを追加

- 分析機能 -

ブラウザ以外からのアクセスは何？

Top Browsers

Browser	# Logs	% of Logs
Mobile Safari	28850	37.47%
Firefox	14654	19.03%
IE	14565	18.92%
Chrome	14531	18.87%
Other	4386	5.7%

特定の国からのアクセスが多い？

Top Locations

Location	# Logs	% of Logs
	11173	14.51%
	7486	9.72%
	7428	9.65%
	7419	9.64%
JPN	7403	9.62%
	7386	9.59%
	7376	9.58%
	5678	7.38%
	4386	5.7%
	3810	4.95%

このURLパス宛のアクセスが多いのは何故？

URL Paths

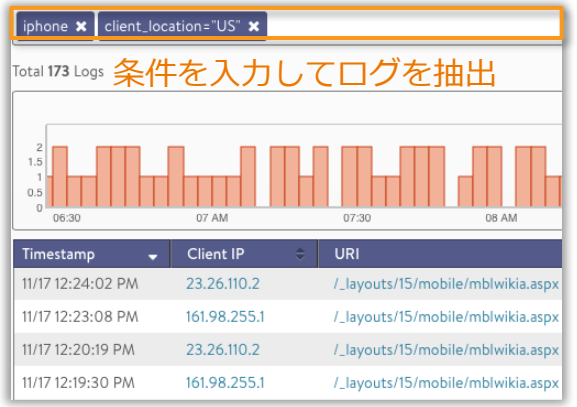
URL Path	# Logs	% of Logs
/	25935	36.9...
/login.php	14123	20.1...
/about.php	7621	10.8...
/index.php	7096	10.12%
/security.php	7030	10.0...
/instructions.php	6904	9.85%
/vulnerabilities/exec/	724	1.03%
/vulnerabilities/sqli/	664	0.95%
/index.action	12	0.02%
/login.action	6	0.01%

200以外のレスポンスコードは大丈夫？

Top Response Codes

Response code	# Logs	% of Logs
302 (検出)	40057	57.12%
200 (OK)	27961	39.87%
403 (禁止)	2083	2.97%
400 (要求が正しくありません)	13	0.02%
503 (サービスを利用できません)	7	0.01%
404 (見つかりません)	2	0%

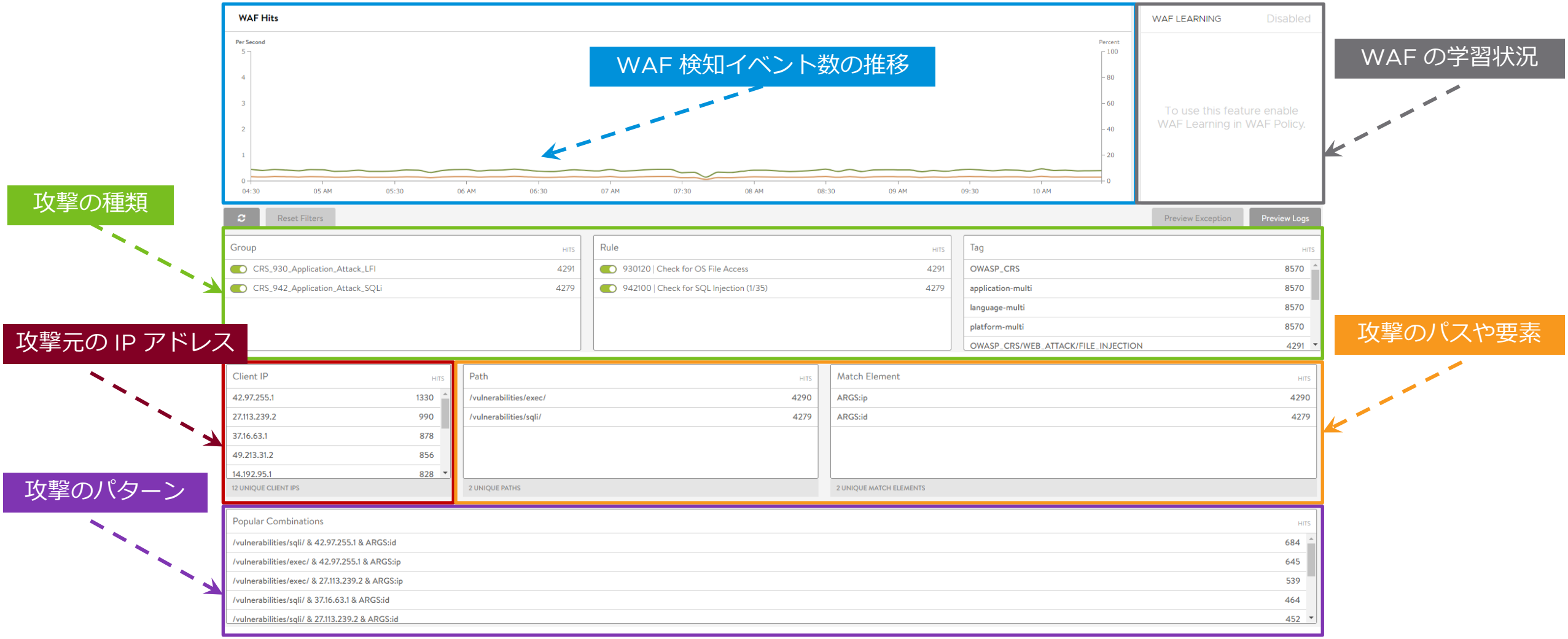
- 検索機能 -



ドリルダウンして詳細を確認

WAF の統計情報の表示

攻撃の種類や傾向を把握することでセキュリティのリスクを適切に管理

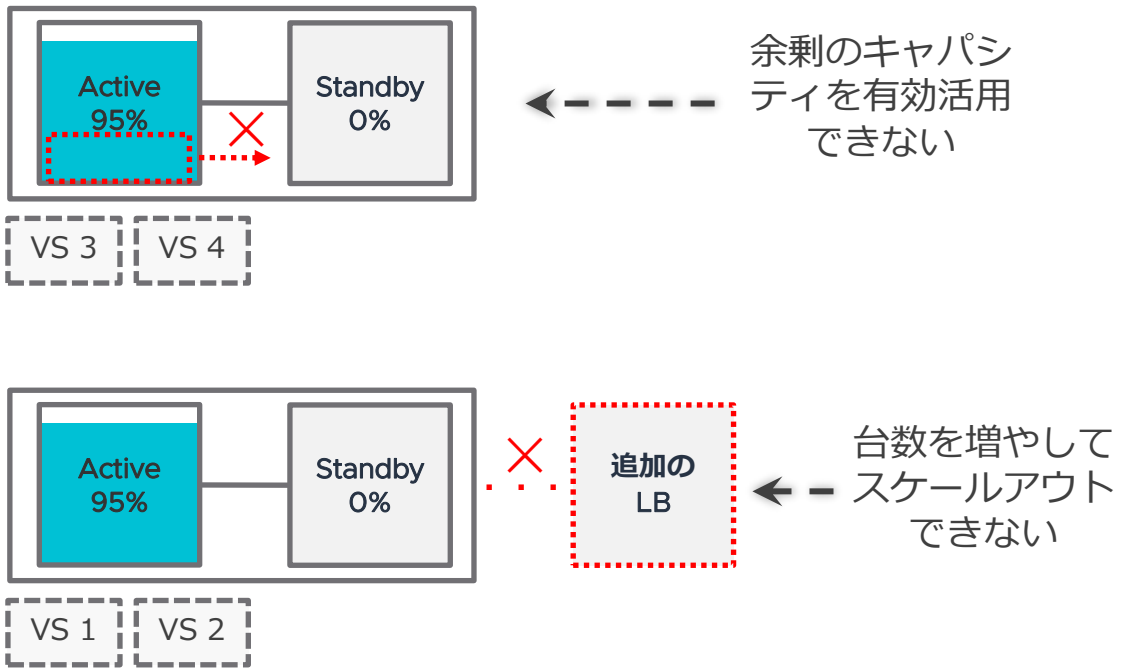


性能問題を克服できるスケールアウト型の拡張性

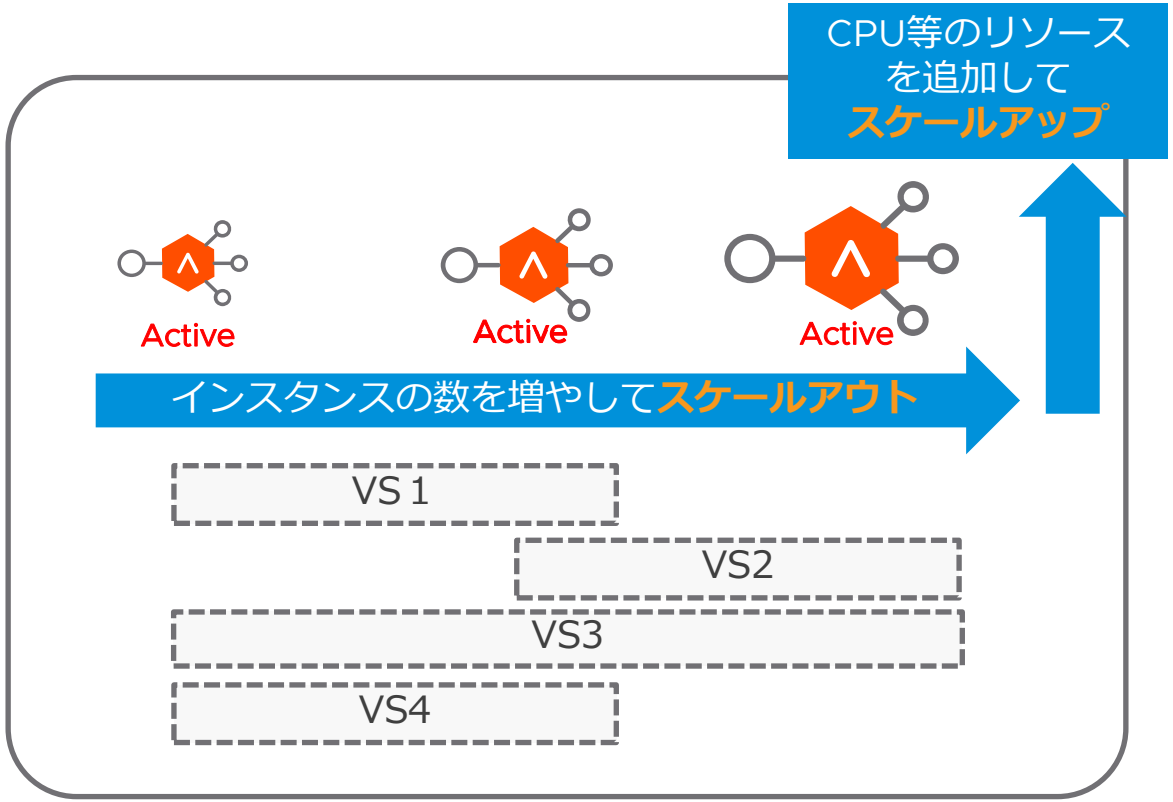
伸縮自在なスケーリング

処理負荷が高まった場合、自動またはオンデマンドで処理性能を引き上げることが可能

従来型のLB（Active-Standby構成）



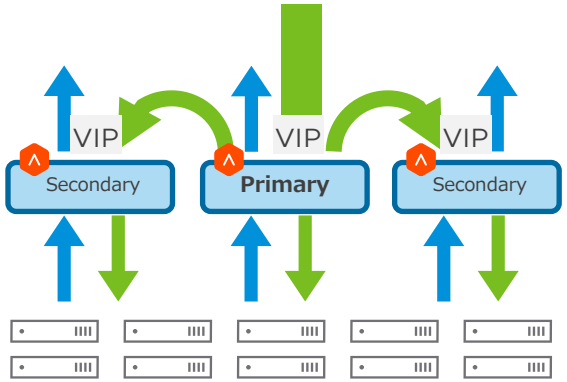
NSX ALB（ソフトウェアベースの Active-Active 構成）



VS Virtual Service (VIP)

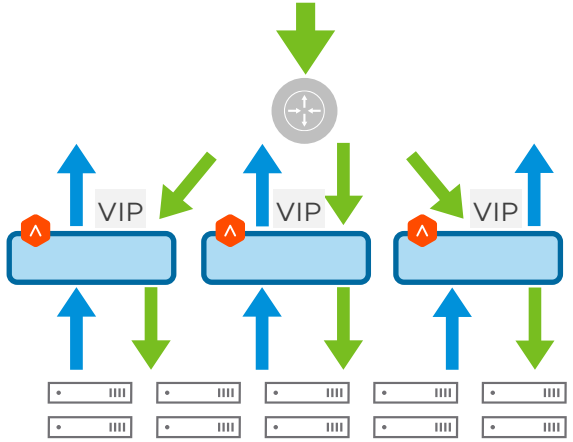
スケールアウトの技術詳細

Native (L2) スケールアウト



- 同じ仮想サービスが複数の SE で稼動する場合、1 つの SE が Primary SE となり、残りの SE はすべて Secondary SE となる
- Primary SE は VIP に対する ARP 返答を受け持ち、自らロードバランス処理を行うとともに、トラフィックの一部をSecondary SE にも転送する
- Secondary SE は Primary SE から受け取ったトラフィックに対してロードバランス処理を実行

ECMP (L3) スケールアウト



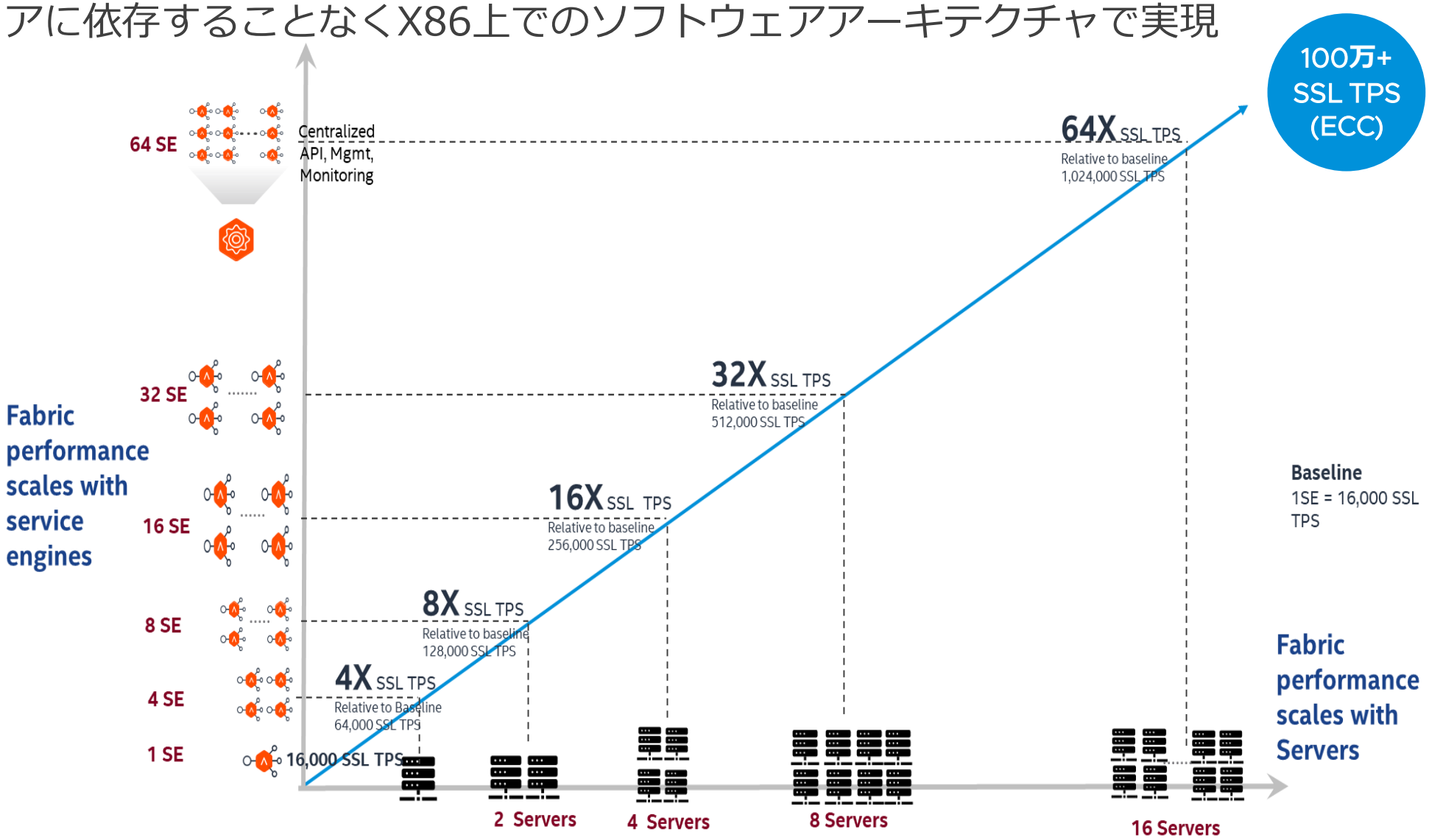
- すべての SE がVIPを Route Health Injection を利用し BGP 経由で配信
- ルータがクライアントからのフローをハッシュで SE に分散して転送
- SE はルータから受け取ったトラフィックに対してロードバランス処理を実行

驚異的なコストパフォーマンス

特定のハードウェアに依存することなくX86上でのソフトウェアアーキテクチャで実現

"Intel Xeon Processorを用いるだけで、通信影響なくリニアに1M SSL TPSまで容易にスケールさせることが可能
専用ロードバランサを用いた場合には、数十万ドルの費用がかかるものと思われます。"

Amit Pandey, ex-CEO of Avi, Head of NSX Services, VMware

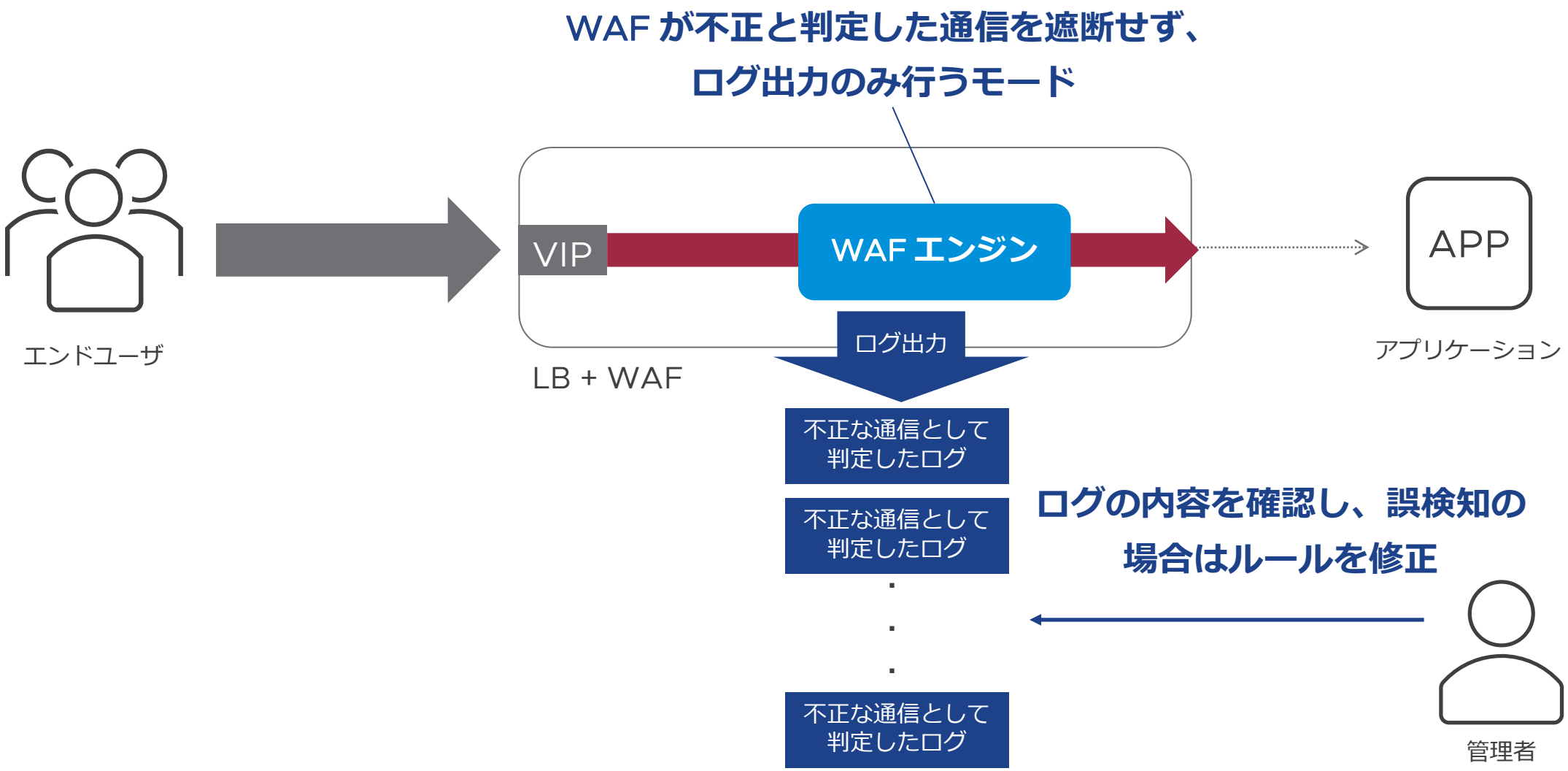


出典：VMware NSX Advanced Load Balancer powered by Intel Xeon Scalable processors distributed over 1 million SSL transactions per second
https://info.avinetworks.com/hubfs/Load-Balancer-Performance-Benchmark-Report.pdf

誤検知によるサービス影響を最小化するための運用方法

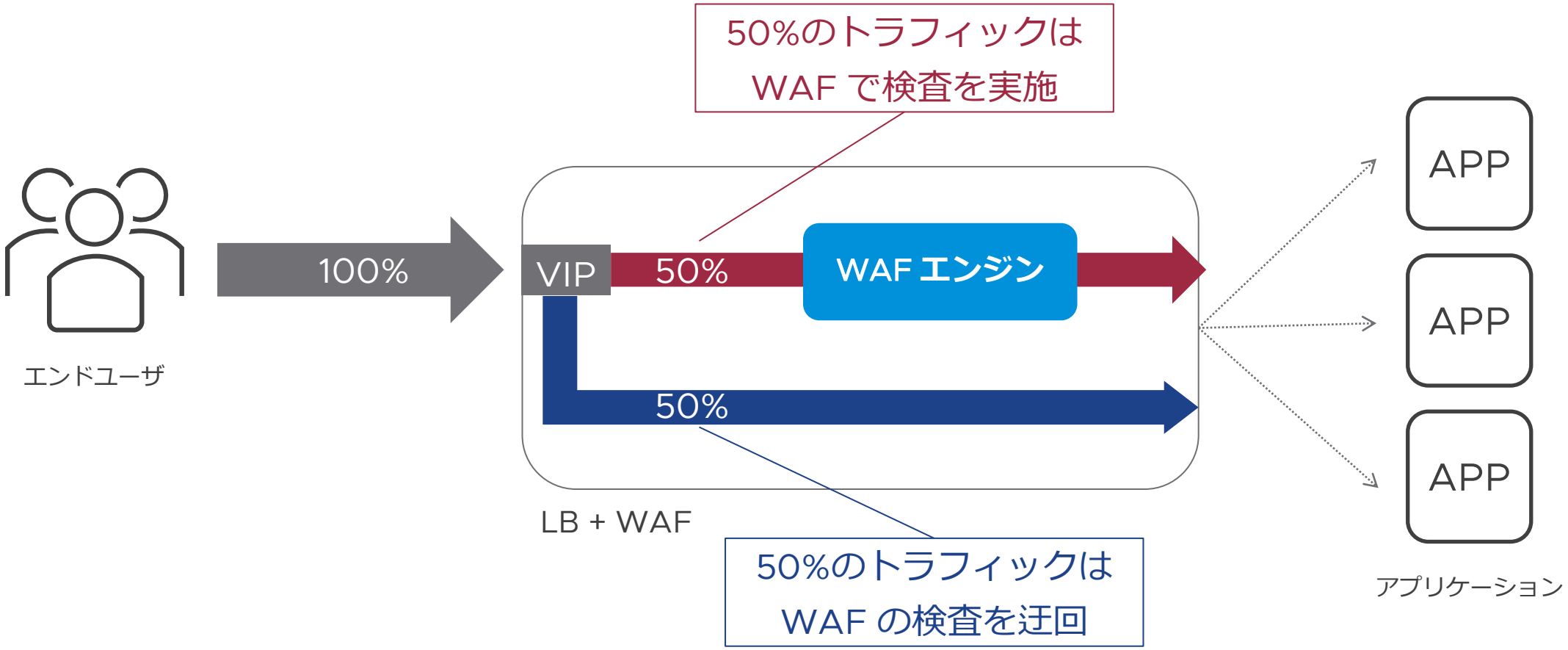
検知モードの適用

ルールの追加・変更するときは、最初から遮断モードにせず、一定期間は検知モードで試験運転



カナリアリリース

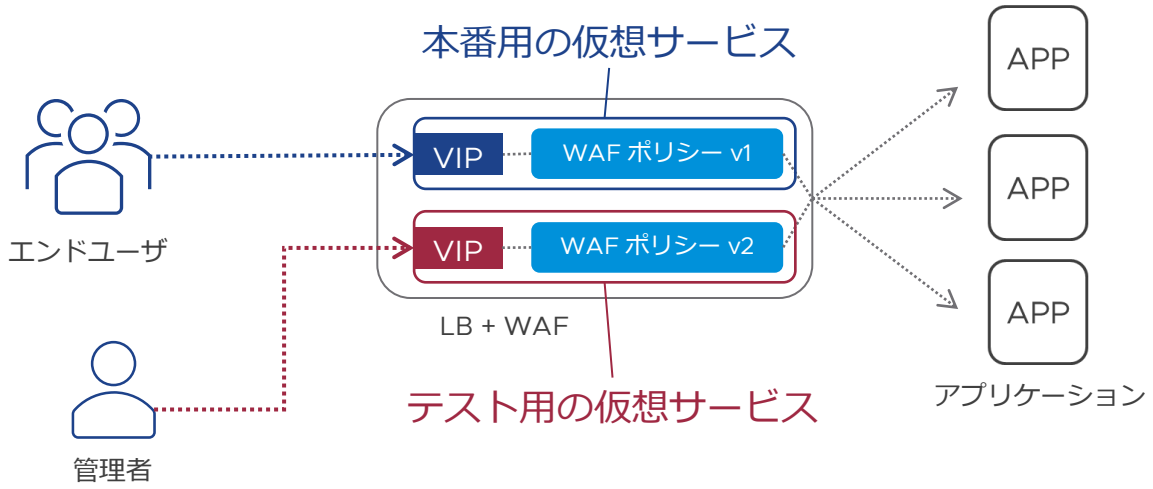
最初からすべての通信を WAF の検査対象とせず、一定割合の通信のみを WAF で検査



テスト環境の作成

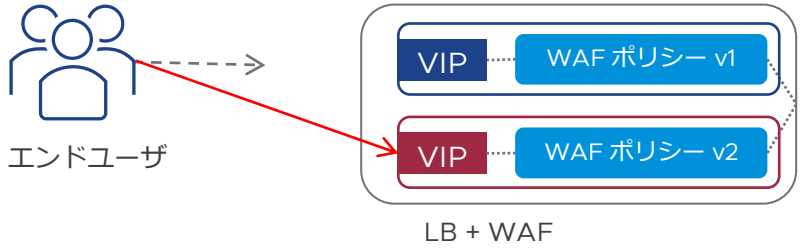
テスト用仮想サーバを作成し、本番環境に影響を与えずに、管理者がWAFのルールを事前に試験

本番環境に影響を与えない試験方法

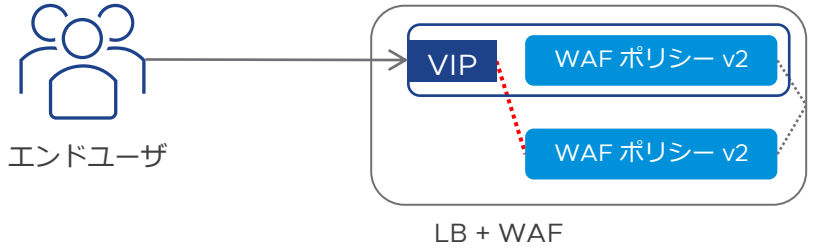


柔軟な移行オプション

DNSによる切替



WAFポリシーの差替



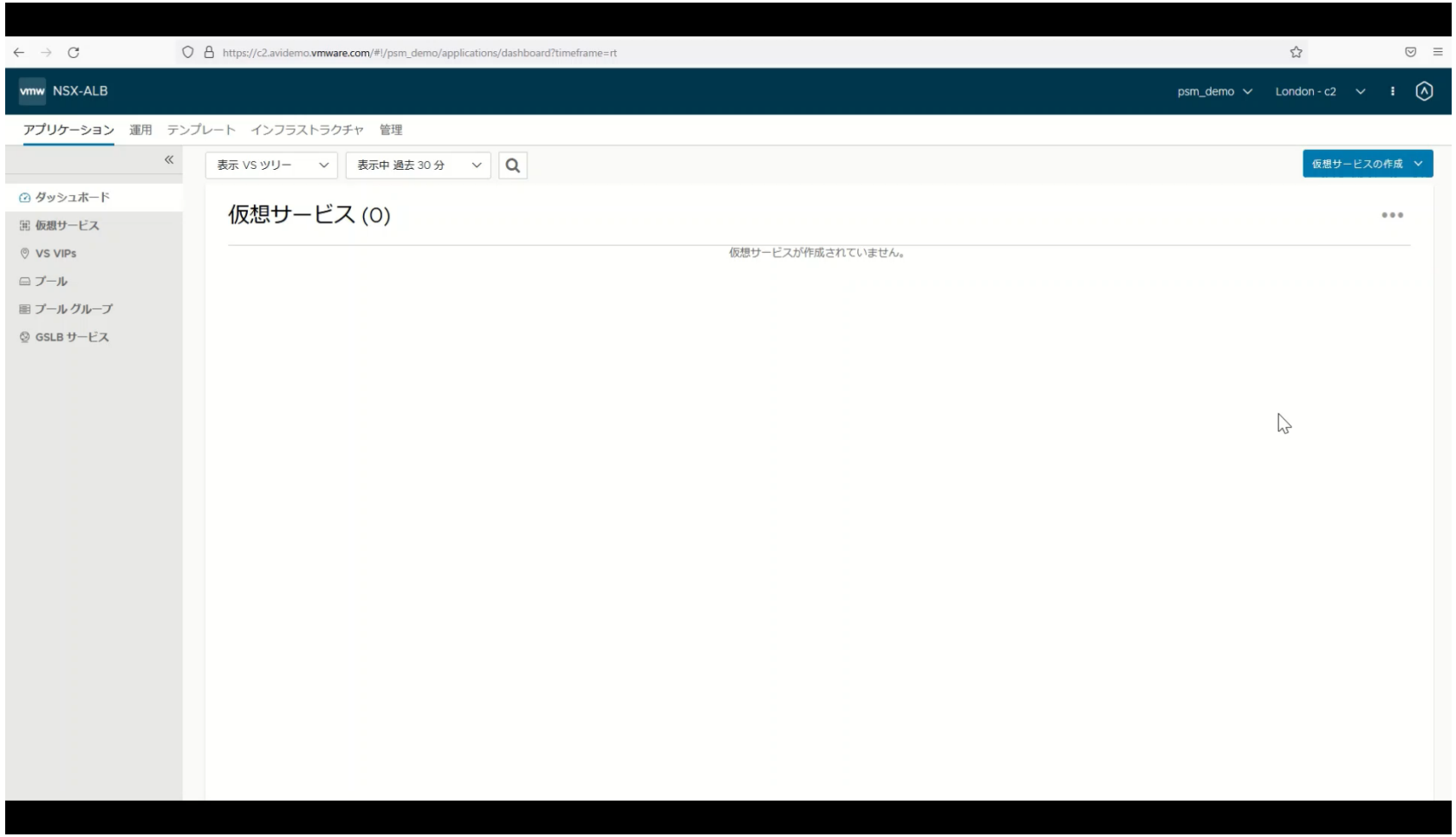
問題が発生した場合は、速やかに切り戻すことが可能

NSX Advanced Load Balancer WAFのデモ

Virtual Service の作成

Virtual Service の作成や LB の自動デプロイについて紹介

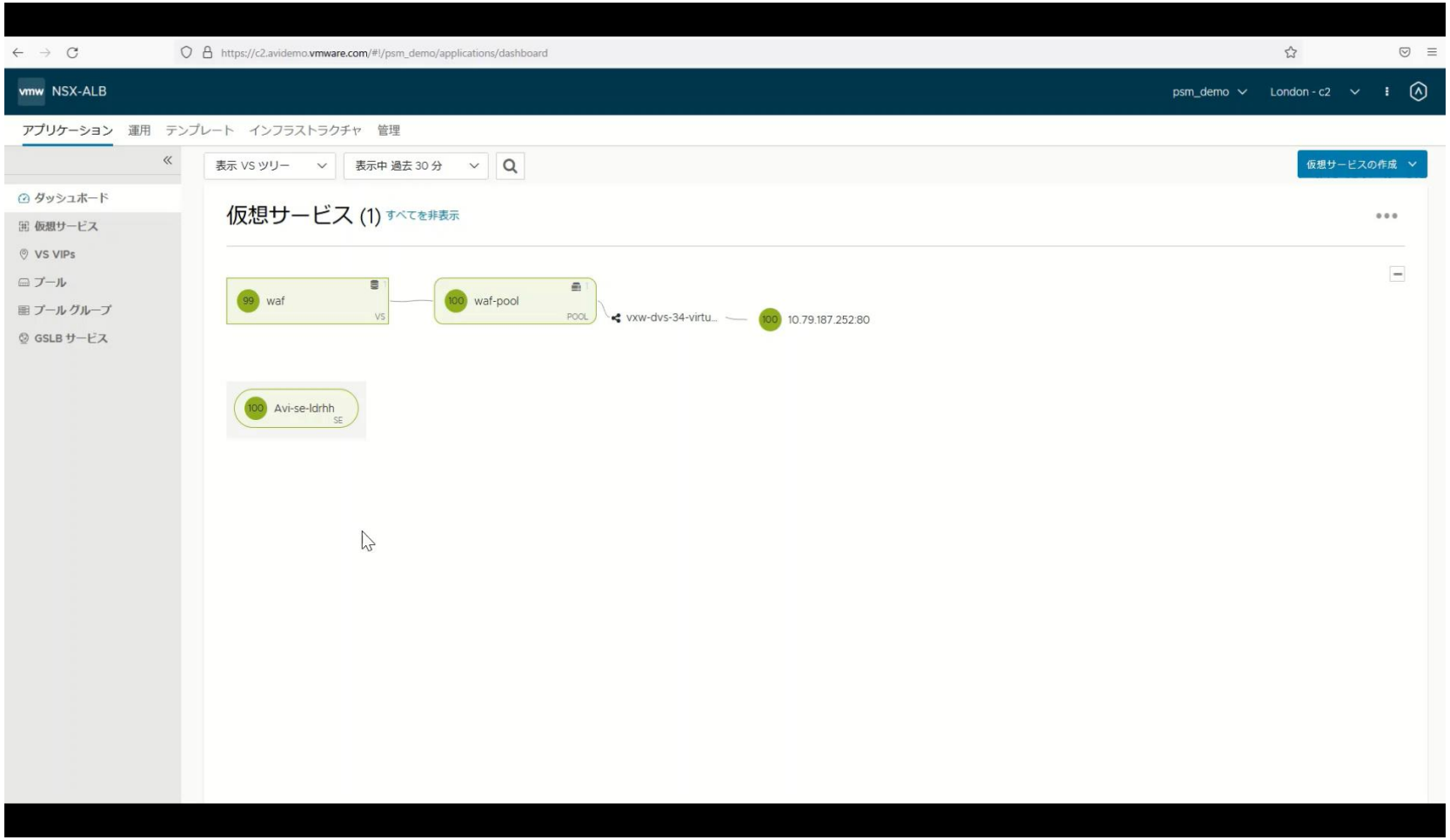
1. Virtual Service と Pool の作成
2. Virtual Service にアクセスして動作確認



WAF の設定①（ポジティブセキュリティ）

ラーニングの仕組みとポジティブセキュリティによる防御を紹介

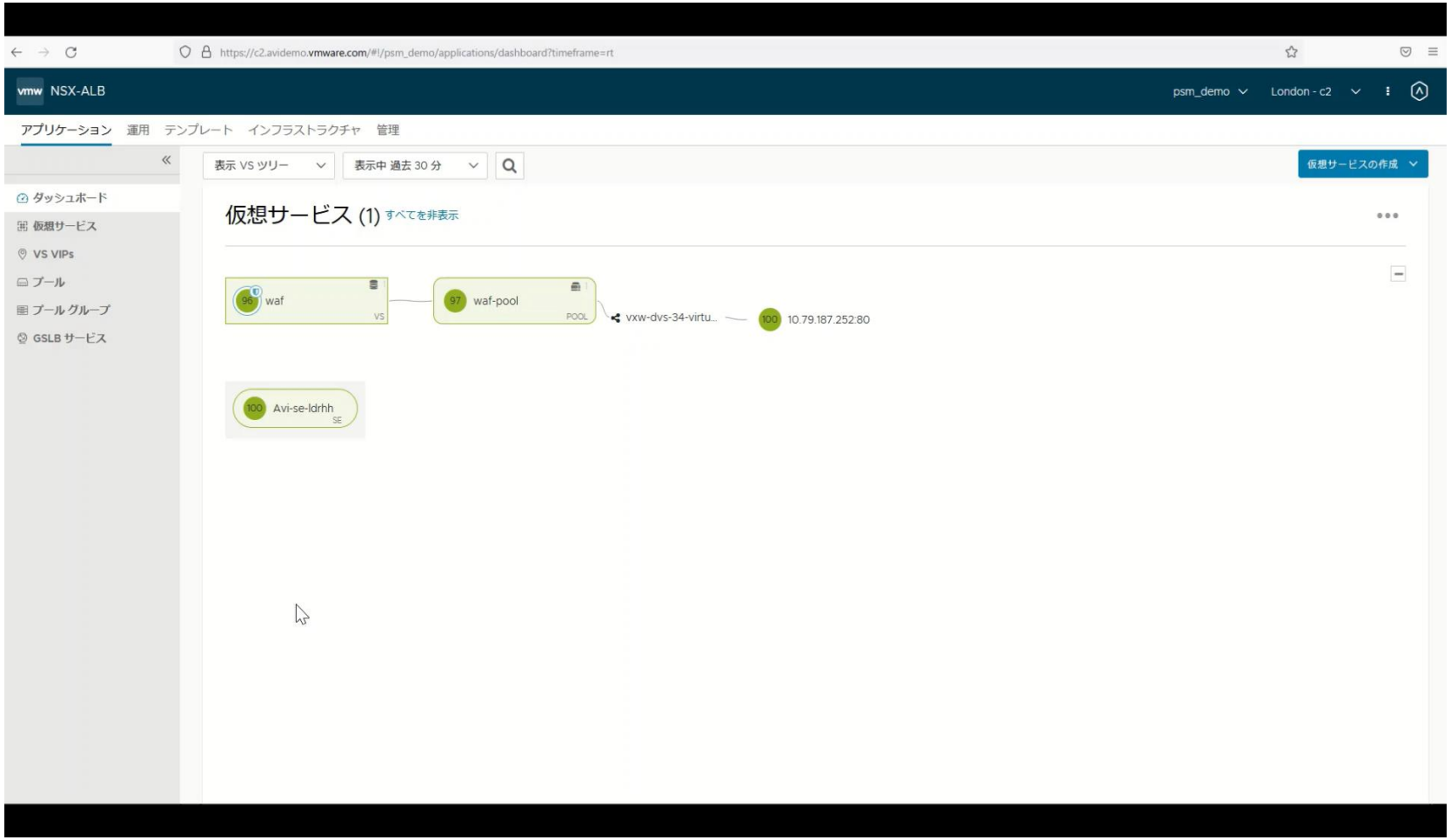
1. WAF のポリシーとプロファイルを作成
2. ラーニング機能を有効化し、トラフィックの自動学習を開始
3. ラーニング結果に基づきルールが自動的に作成されていることを確認
4. 検知モードでは攻撃が遮断されないことを確認
5. WAF を遮断モードに変更
6. ポジティブセキュリティのルールによって攻撃が遮断されることを確認



WAF の設定②（シグネチャ）

シグネチャによる防御と例外ルールの追加方法を紹介

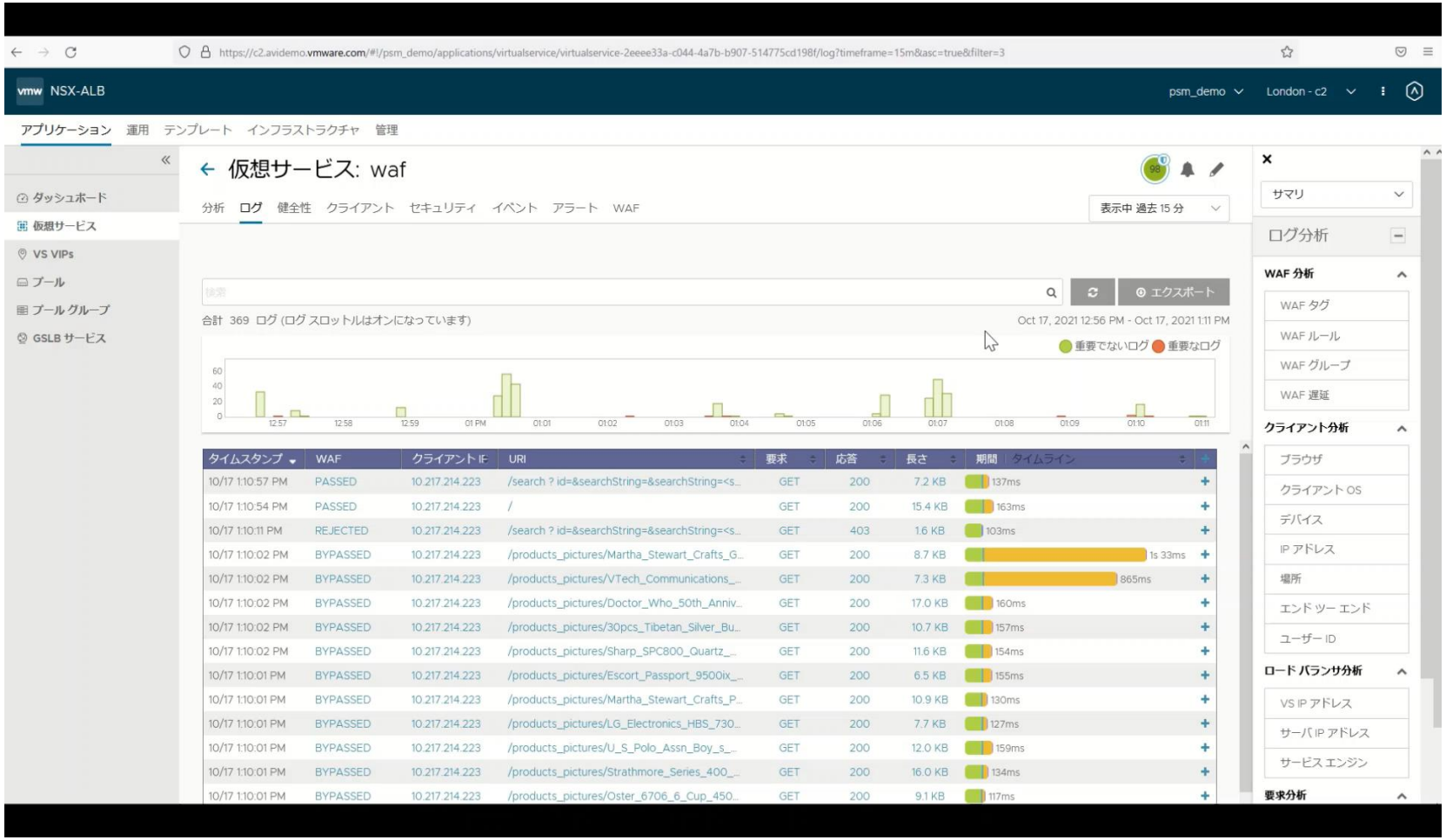
1. 検知モードでは攻撃が遮断されてないことを確認
2. WAF を遮断モードに変更
3. 攻撃がシグネチャによって遮断されることを確認
4. 該当シグネチャを例外ルールに追加
5. 攻撃がシグネチャで遮断されないことを確認



WAF の設定③（可視化）

WAF の可視化について紹介

1. WAF トランザクション単位の詳細を可視化
2. 分析と検索機能で必要なデータを抽出



WAF の選定で大事なポイント

攻撃を検知・遮断する製品能力だけでなく、運用者が WAF を使いこなせることが重要

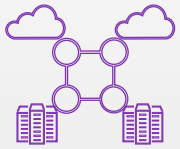
運用操作性

素早く簡単にデプロイでき、運用操作性が優れていること

可視化

脆弱性発見時の過去調査や誤検知・検知漏れに対応できること

マルチクラウド



オンプレ・クラウドを問わず、あらゆる環境で一貫性のあるセキュリティ ポリシーで運用できること

Blog: VMware のセキュリティ ソリューションを分かりすく解説

VMware Japan Blog には セキュリティ関連の Blog が多数掲載されております

NSX Advanced Load Balancer | NSX Advanced Load Balancer | セキュリティ | ネットワーク

VMware のWAF でApache Log4j の脆弱性対策してみた

kenyah@vmware.com
December 22, 2021

2021年12月、Apache Log4j に深刻な脆弱性（CVE-2021-44228/CVE-2021-45046）が見つかりました。この脆弱性を悪用した攻撃が全世界で猛威を振るっています。こちらのブログでは、NSX Advanced Load Balancer を活用したWeb のセキュリティ対策を具体的にご紹介いたします。

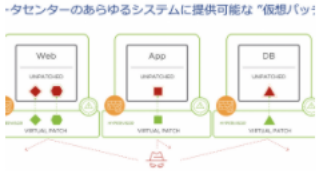
NSX Advanced Load Balancer による対策とは

Apache Log4j の脆弱性を悪用した攻撃を防ぐためには、NSX Advanced Load Balancer が提供するWAF(Web Application Firewall) と IP Reputation 機能が有効な対策になります。

WAF は、従来のファイアウォールや IPS/IDS では防ぐことができない、Web アプリケーションの脆弱性を悪用する攻撃に対処するための機能です。シグネチャと呼ばれる既知の攻撃パターンやアプリケーション脆弱性と照らし合わせ、該当する通信を検知・遮断します。WAF については、以前にもブログで詳しく紹介しましたので、こちらをご覧ください。

IP Reputation は、インターネット上の行動履歴などをもとに、発信元IP アドレスをスコアリングして、リスク評価する仕組みです。IP Reputation のデータベースと照らし合わせて、“評判の悪い”IP アドレスからの通信を遮断することが可能です。

https://blogs.vmware.com/vmware-japan/2022/12/waf-log4j.html



Network Security | NSX | NSX Advanced Load Balancer | NSX Advanced Load Balancer | NSX Data Center | セキュリティ | ネットワーク

VMware NSX による ”仮想パッチ” で脆弱性対策にアジリティを

Susumu Nagatoishi
December 16, 2021

2021 年 12 月 10 日に米国国立標準技術研究所（NIST）から公開され、Log4j または Log4Shell（以下、Log4Shell という）として知られるゼロデイ脆弱性（CVE-2021-44228）を悪用した攻撃の観測事例も出てくるなど、実際にサイバー攻撃の標的にされてきていることが明らかになってきました。CVE-2021-44228 には、現在 CVSS v3 リスクスコアの基本評価基準(Base Metrics)が 10（最大）と、最も深刻度が高い「緊急」という評価が割り当てられています。

VMware の脅威分析ユニット TAU（Threat Analysis Unit）は、本脆弱性を悪用した試みとなる活動を多く観測し、活動の監視および評価し続けています。

VMware 製品固有の影響については、VMwareセキュリティアドバイザリをご参照ください。

本脆弱性は、Apache log4jというオープンソースの Java プログラム用ロギング API と、DNS や LDAP を利用するための Java ライブラリである JNDI（Java Naming & Directory Interface）が大きく関係しています。
Lo4j は JNDI を Lookup する機能があり、書き込んだログの一部を自動で変数化します。これを悪用して、RCE（Remote Code Execution：リモートコード実行）を行えるようになり、その手順例は次のようになります。

https://blogs.vmware.com/vmware-japan/2022/12/vcn-virtual-patch.html

Thank You