WAFは本当に必要か？
Webアプリケーションへの攻撃を体験し、WAFの重要性を学ぶハンズオンセミナー

東京エレクトロン デバイス株式会社

自己紹介

熊谷 敦文（くまがい あつふみ）

2014-2020年
- F5社 BIG-IPのプリ/構築/サポート

2020-2022年
- F5社 NGINX 製品のプリ
- Cloudflare社 プリ/構築/サポート（現在弊社では取り扱い無し）

2022年～
- F5社 Distributed Cloud Servicesプリ/構築/サポート

あつふみ という名前でブログ書いてます

東京エレクトロンデバイス ブログ

https://cn.teldevice.co.jp/blog/
https://cn.teldevice.co.jp/blog/search/?q=F5XC
https://cn.teldevice.co.jp/blog/search/?q=NGINX

【最近の趣味】

アニメ（幅広いジャンルのアニメをBGM感覚で視聴しています）

ゲーム（Ark: Survival EvolvedやAnno1800をやってます）

ハンズオンセミナーのゴール

- Webアプリケーションを狙った代表的な攻撃手法を知る
- WAFの仕組みを理解する
- クラウドWAFを利用して、ハンズオン環境のアプリケーションサーバーを保護することができる
- WAFの運用方法（ポリシーチューニング）をイメージできる

ハンズオンセミナーの簡単な流れ

1. **Webアプリケーションを狙った代表的な攻撃手法**
   - 近年のWebアプリケーションに対する攻撃について
   - DDoS/SQLインジェクション/Webスキミング攻撃の解説

     SQLiとWebスキミングは、実際に攻撃をしてもらい、その攻撃の仕組みを解説
2. **WAF（クラウドWAF）について**
   - WAFとは
   - WAFの検知方法（シグネチャ、挙動）を説明
   - クラウドWAFとは（クラウドのメリット：手前でブロックする重要性などを説明）
3. **F5のWAF（WAAP）ソリューションについて**
4. **ハンズオントレーニング環境について**
5. **ハンズオントレーニング**
   - HTTP LB/App Firewall作成
   - リクエストログ/セキュリティイベントログの見方
   - チューニング方法
   - デモ：Client-Side Defense（Webスキミング対策）の設定
6. **補足**

ハンズオントレーニング当日までに実施していただくこと

ハンズオントレーニング当日までに実施していただくこと

- 弊社にユーザー情報のご連絡
  - 弊社側でF5 XC にログインできるユーザーを登録をします
    - お客様のお名前（英語表記）、メールアドレスをご連絡ください

      例）

      名前：taro tokyo

      E-mail：taro.tokyo@teldevice.co.jp
    - ハンズオントレーニング前日までに弊社から案内をいたします

- ユーザー初期設定
  - F5 XC コンソールにログインできるところまでご確認ください

初期設定の方法は、次スライド以降を確認ください。

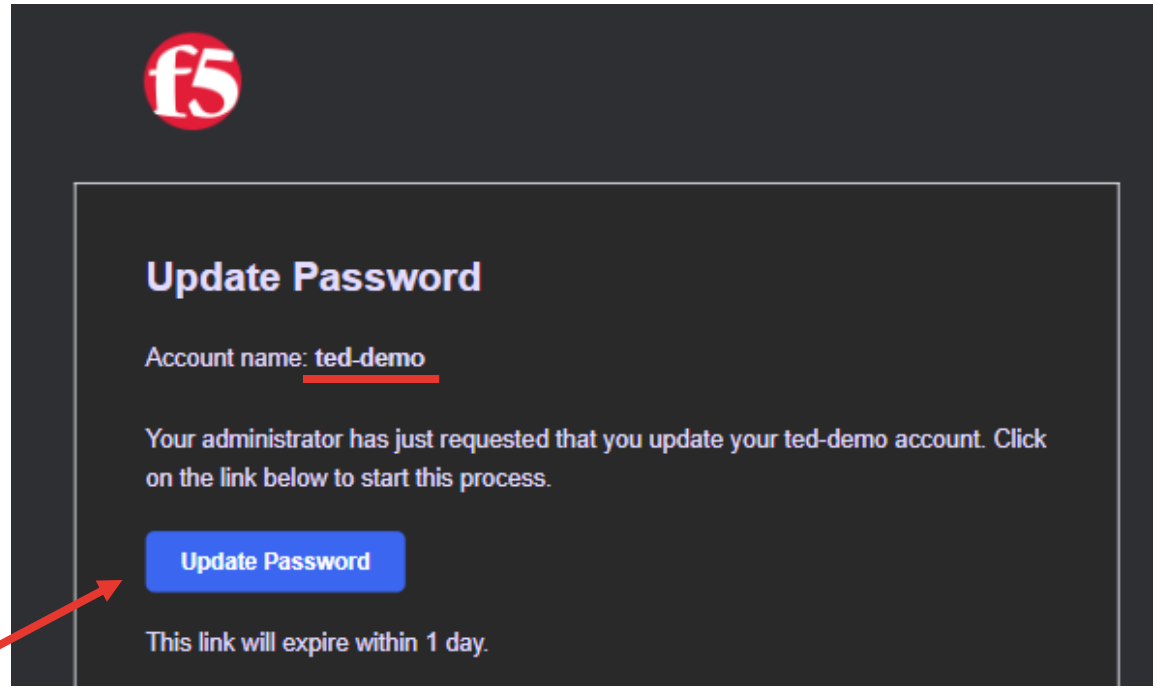
パスワード設定のメール確認

弊社がお客様の登録を実施しますと、
右図のようなメールが届きます。

件名：Update Your Account
差出人：
F5 Distributed Cloud <no-reply@cloud.f5.com>
※受信を確認できない場合は迷惑メールフォルダを
ご確認ください。

パスワード設定のメールが届きますので、
「Update Password」を押下してパスワードの設定をします。

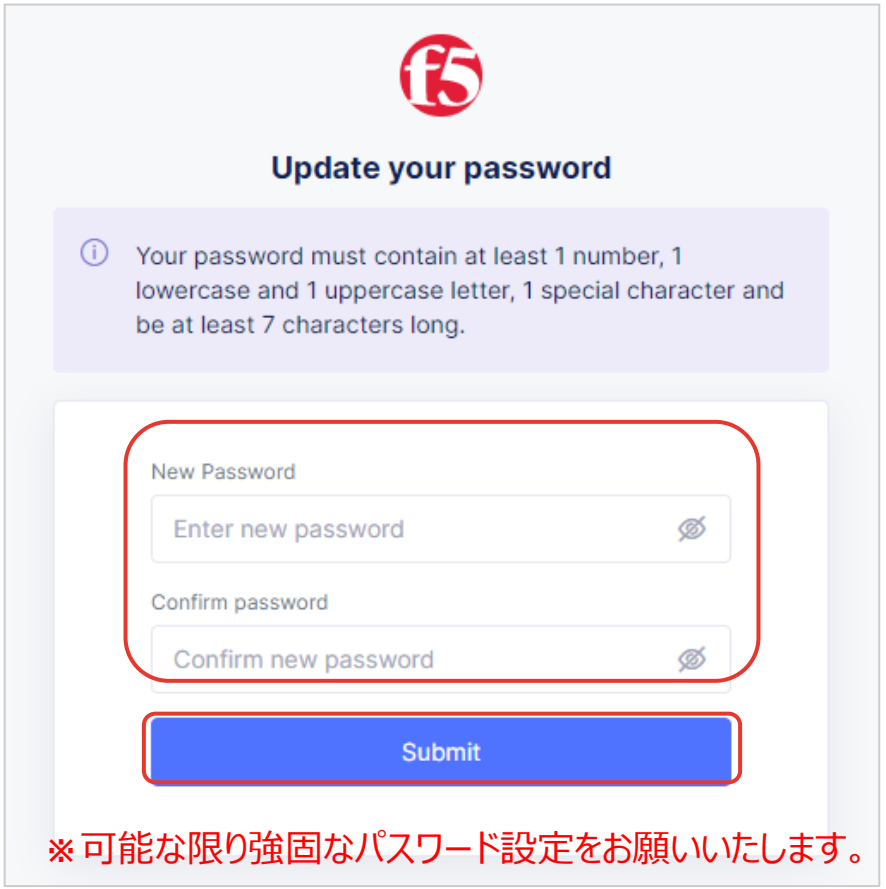
サンプルメール



ここをクリック

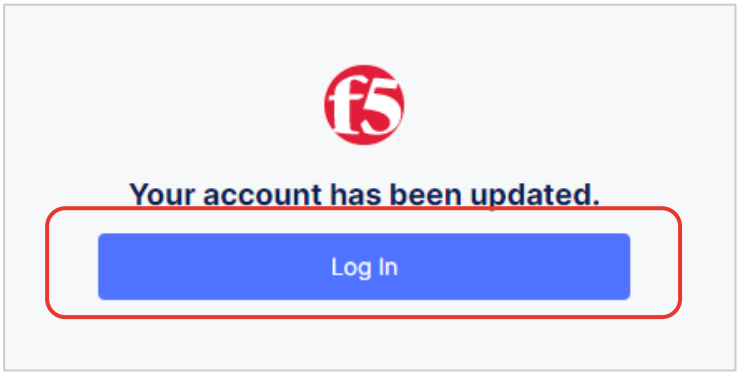
ログインパスワードの設定

「Update Password」をクリックすると、パスワード設定を
求められますので、設定をします
パスワードには、大文字/小文字/特殊文字を１つずつ含み、
かつ７文字以上である必要があります



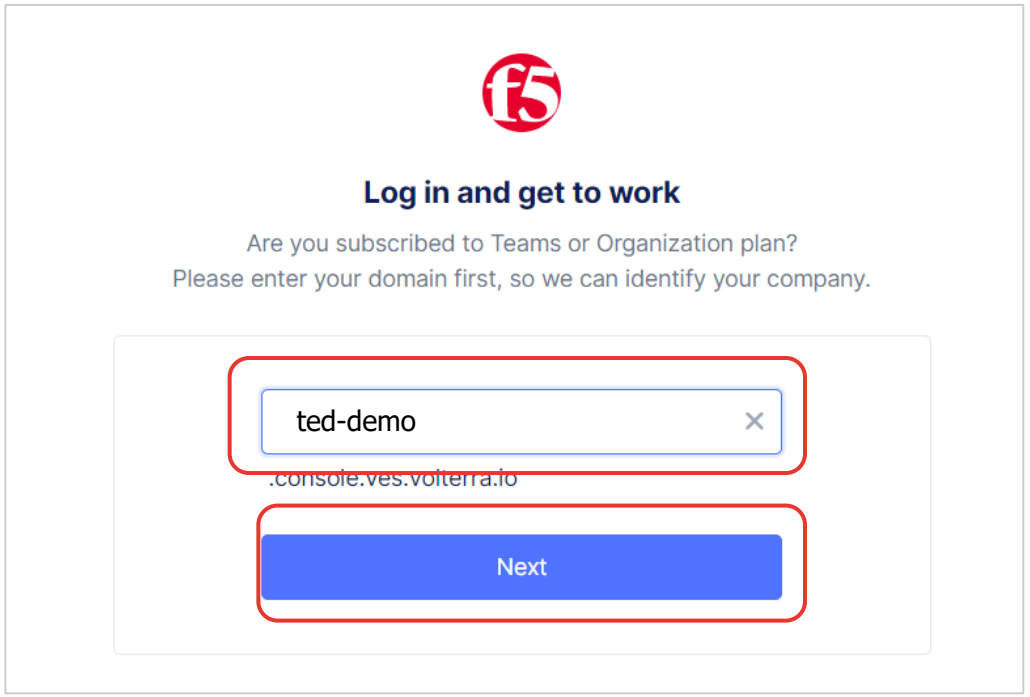
※可能な限り強固なパスワード設定をお願いいたします。

設定後、「Log In」を押下します

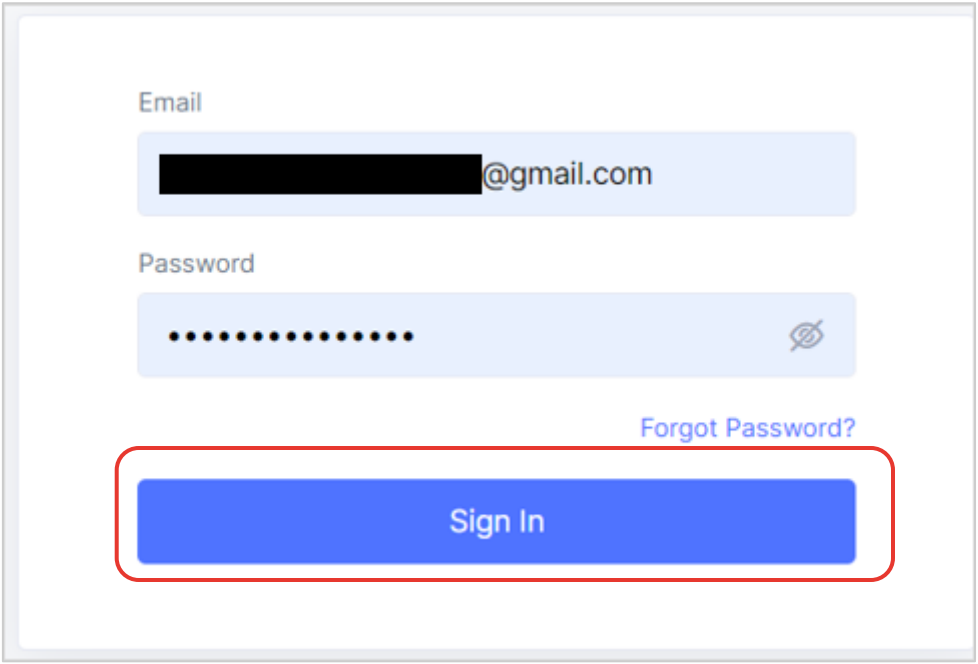


再ログイン

Log in をクリック後、テナント名を求められる画面に遷移した場合は、「ted-demo」と入力してNextを押下します

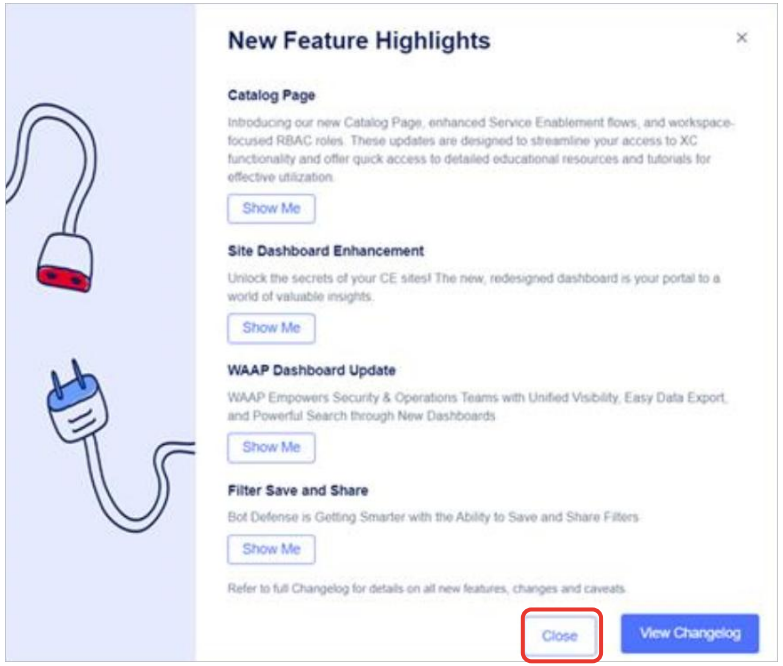


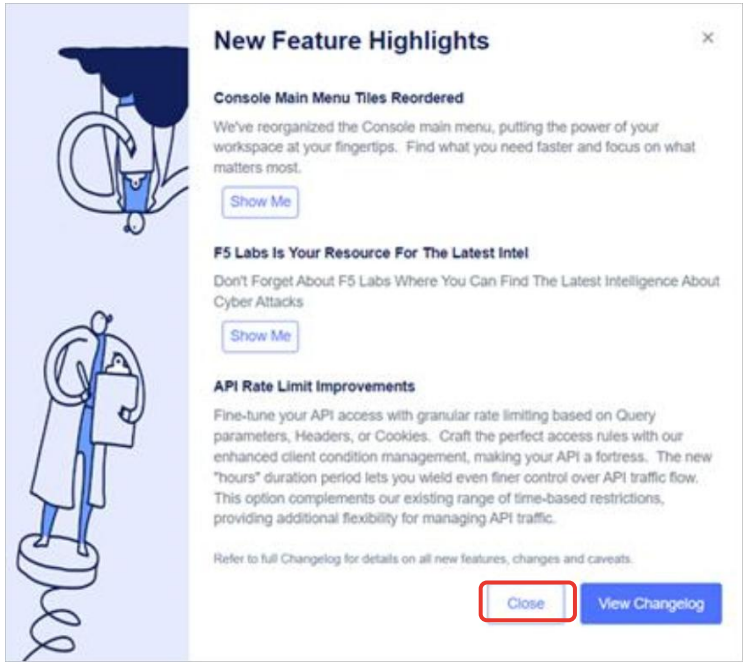
再度Emailアドレスとパスワードを入力し、Sign Inを押下します



New Feature Highlights

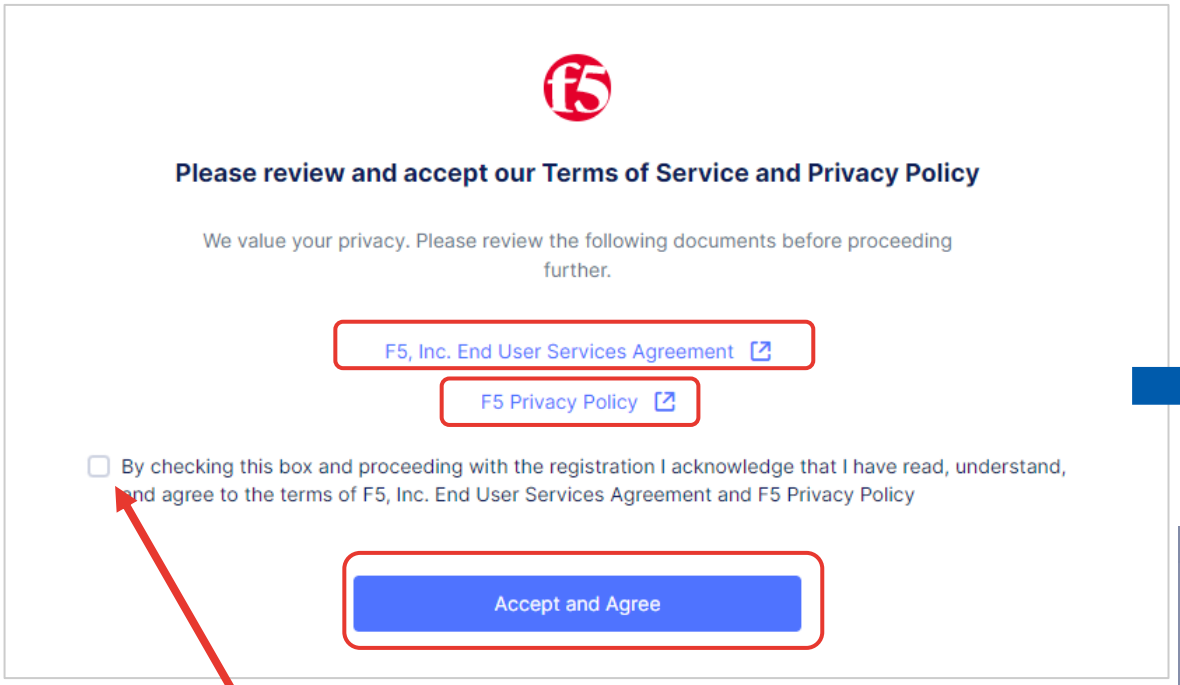
以下のようなNew Feature Highlights画面がでた場合は、[close]を押下します
表示されない場合は次のスライドに移動してください





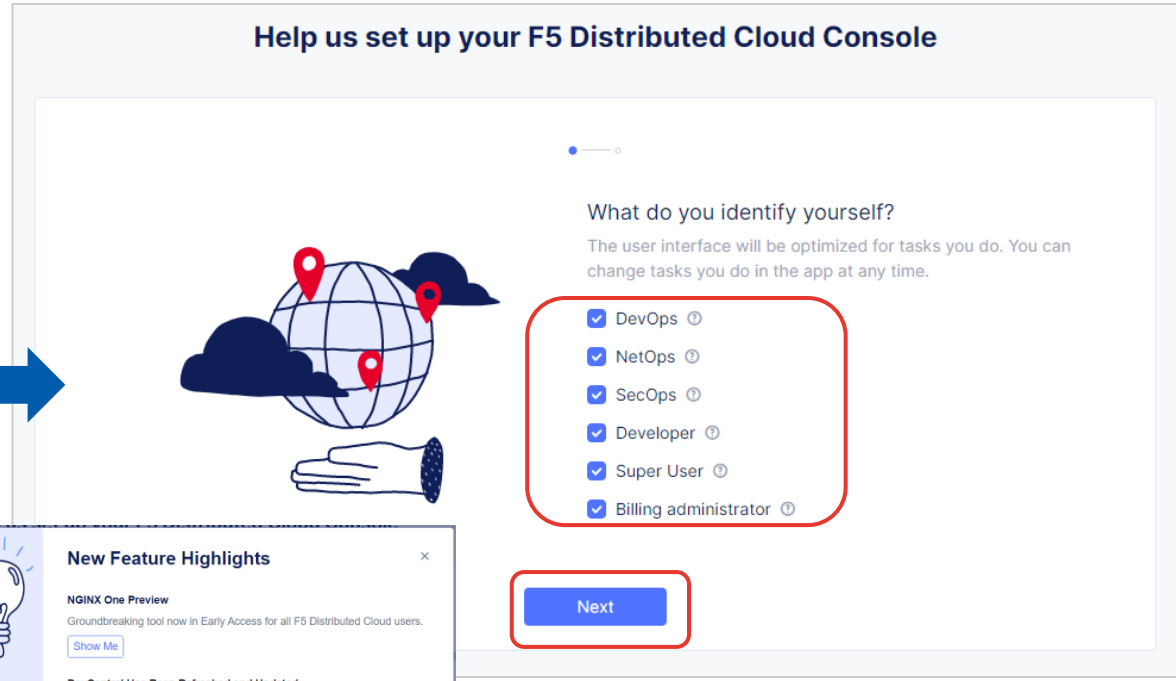
Services Agreement/ Privacy Policyの同意とIdentifyの設定

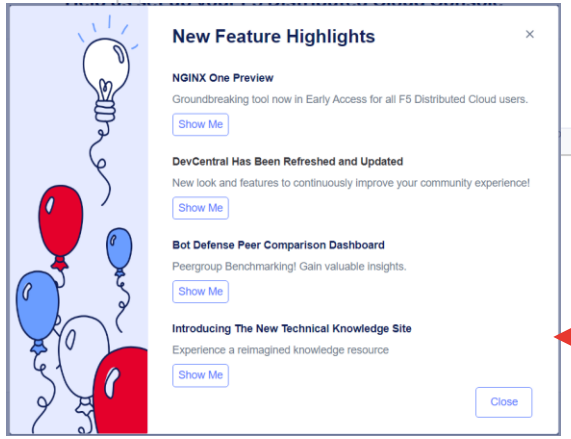
End User Services AgreementとPrivacy Policyの
内容を確認後、チェックボックスにチェックを入れ、
Accept and Agreeを押下します



ここをクリック

DevOps、NetOpsなど全ての項目にチェックをいれて、
Nextを押下します



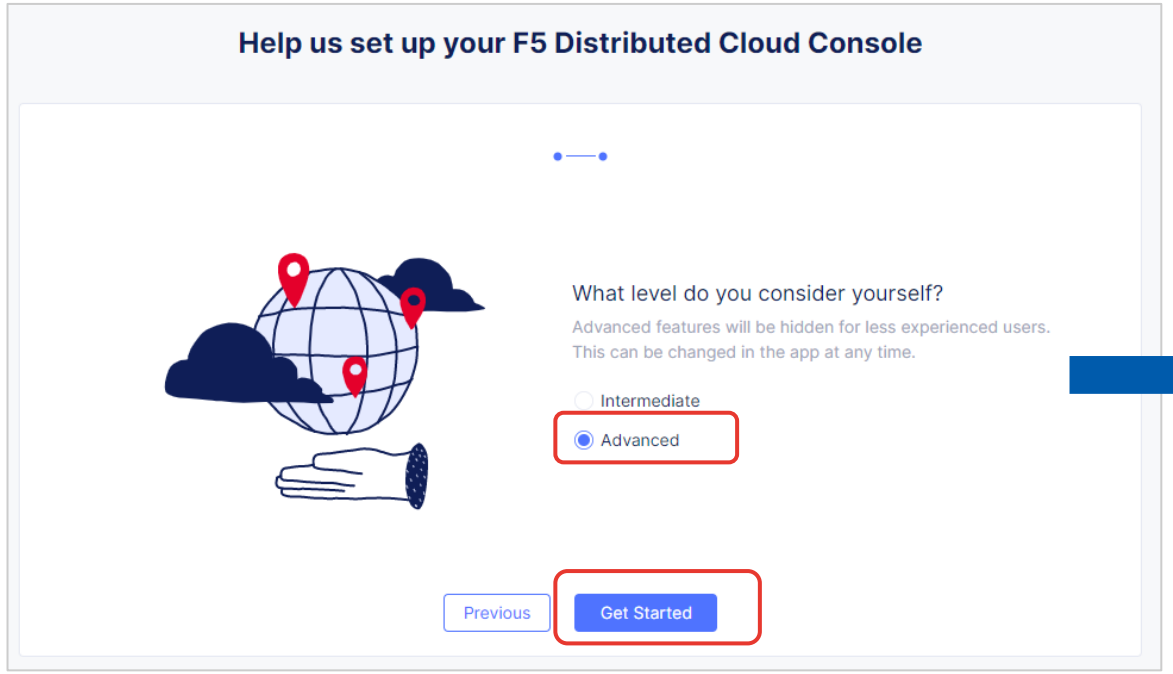


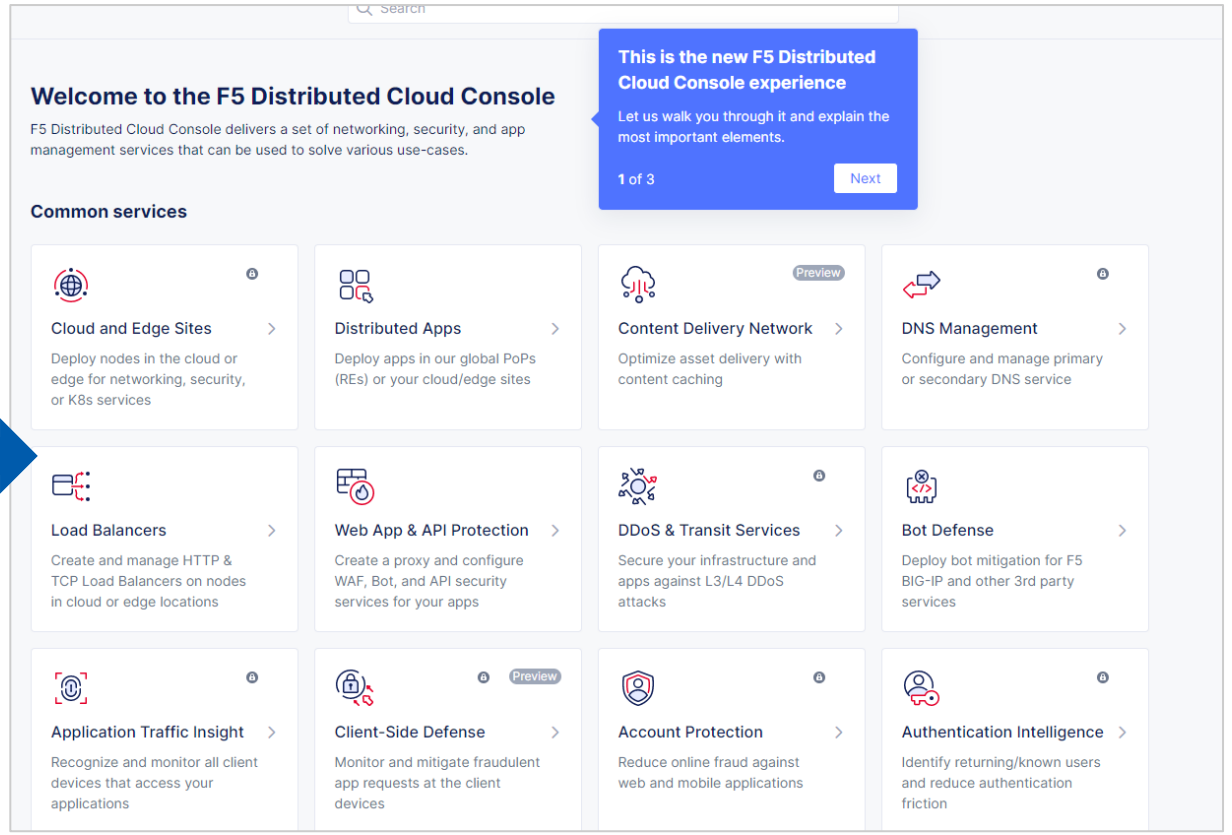
この画面が出たらClose

東京エレクトロン デバイス

Advancedの設定とコンソール画面のトップ画面の確認

Advancedにチェックを入れ、Get Startedを押下します





※コンソール画面のURL
https://ted-demo.console.ves.volterra.io/

事前準備はここまでとなります

Webアプリケーションを狙った代表的な攻撃手法について

近年のWebアプリケーションに対する代表的な攻撃について

本ハンズオンセミナーでは以下の3つをピックアップ

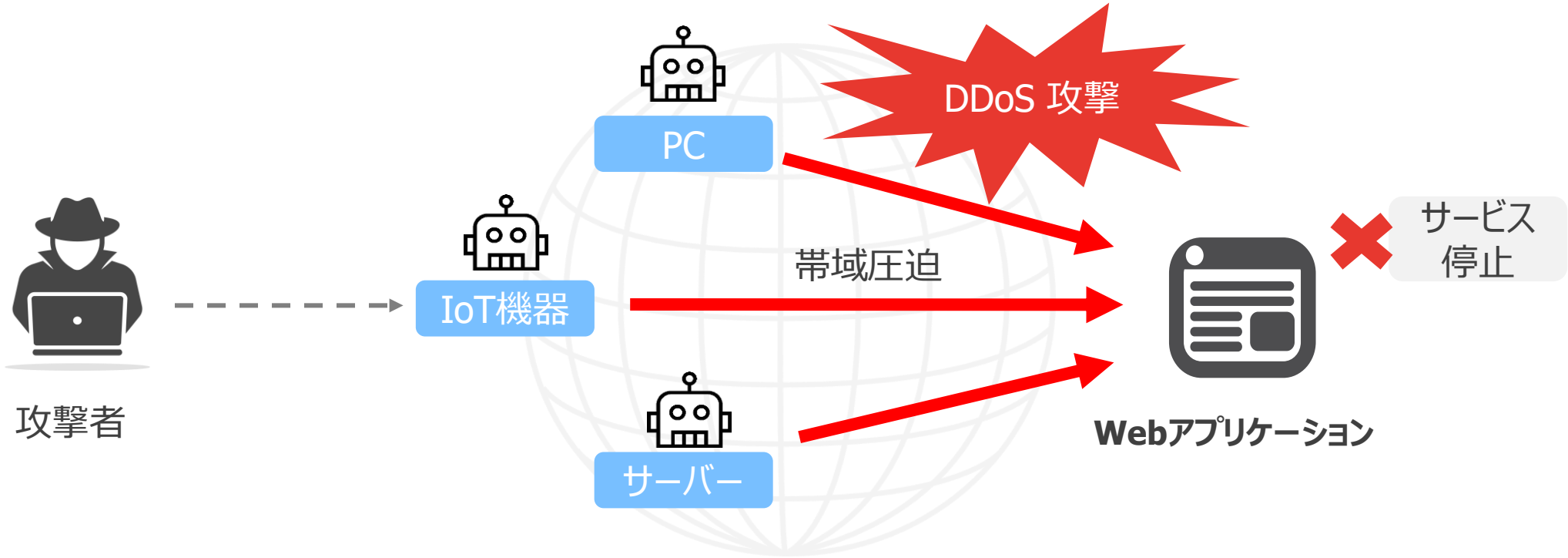
- **DDoS攻撃**
  - 大量のアクセスを複数のコンピューターから一斉に送りつけて、サーバーやネットワークをダウンさせる攻撃
  - 2024年の年末に話題となった攻撃

- **SQLインジェクション（SQLi）**
  - Webアプリの入力フォームなどに悪意のあるSQL文を埋め込むことで、データベースに不正アクセスする攻撃
  - OWASP Top 10※に毎回ランクインする代表的な攻撃
    ※Webアプリケーションにおける重大なセキュリティリスクを世界中から収集されたデータと専門家の意見を基に10項目にまとめたリスト

- **Webスキミング**
  - ECサイトなどに不正なJavaScriptを仕込み、購入フォームなどからクレジットカード情報や個人情報を盗み取る攻撃
  - 利用者や管理者が気づかない間に、機密情報が第三者に送信されてしまうことがほとんど

DDoS攻撃について

複数のコンピューターから一斉に大量のアクセスを送りつけて、サーバーやネットワークをダウンさせる攻撃



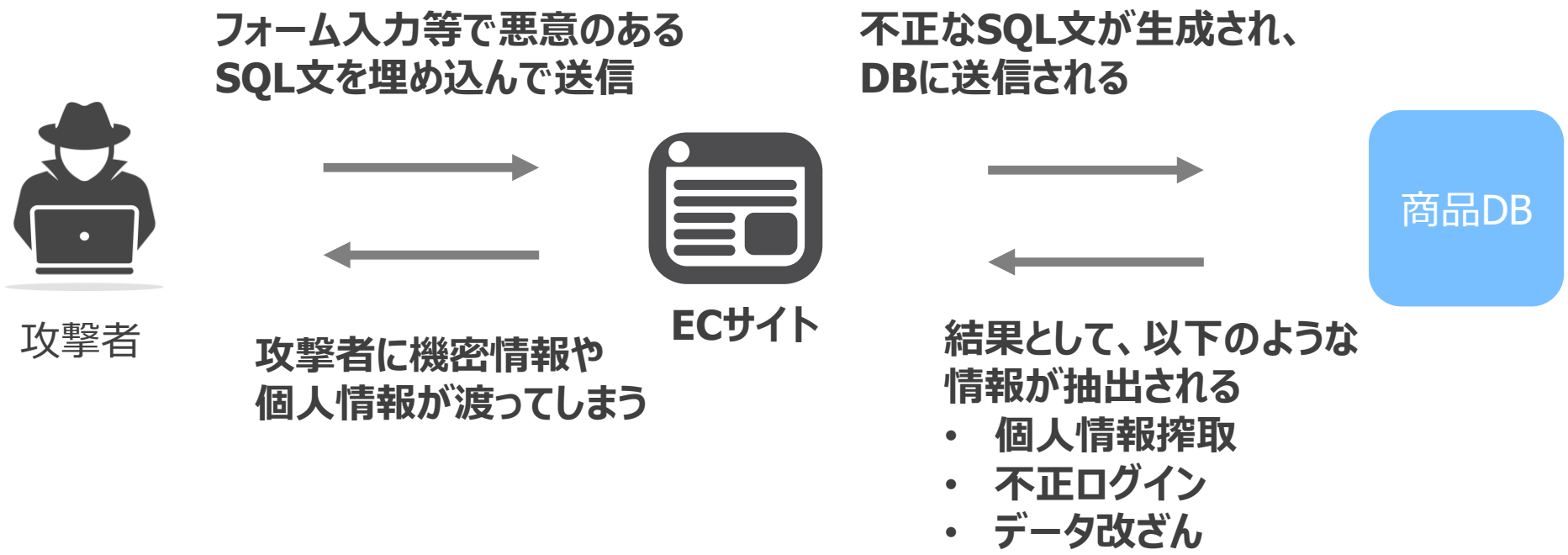
1. 攻撃者（ハッカー）が、世界中のウイルスに感染したパソコンやIoT機器（これを「ボット」と呼びます）を遠隔操作
2. それらの機器から一斉にターゲットのサーバーへアクセス
3. **サーバーのネットワークの帯域が圧迫され通信が遅くなる、または大量のアクセスに対してサーバーは対応しきれずサービスがダウン**

SQLについて

SQL（エスキューエル）とは、データベースと会話するための言葉（命令文）
Webアプリケーションや業務システムの裏側でよく使われており、ECサイトなどのログイン処理や商品検索などの命令文として使われる

SQLインジェクションについて

SQLインジェクションとは、Webアプリケーションに対して悪意のあるSQL文（データベースを操作する命令）を入力することで、個人情報を入手、不正ログイン、データベースを不正に操作することが可能な攻撃手法



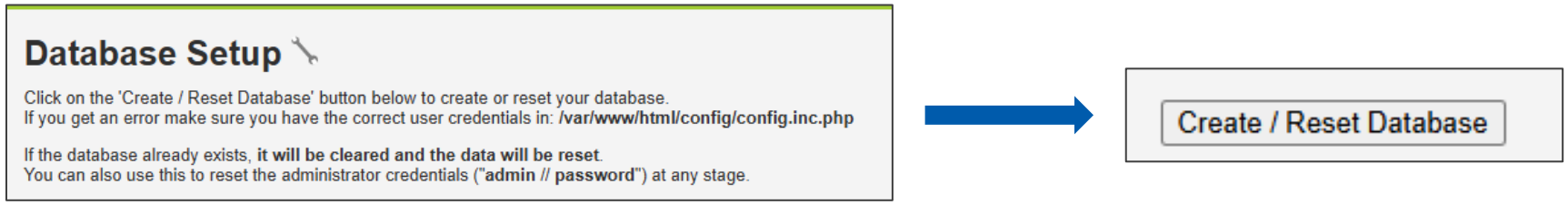
SQLインジェクションを実際に試す①

1. やられサーバーにアクセスする

https://handson-dvwa.xc.dev.tedlab.net
Username：admin
Password：password

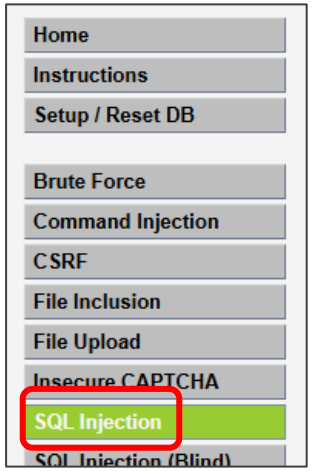
以下のような画面が出てきたらOK

もし「Database Setup」というような画面が出てきたら、画面最下部の「Create/Reset Database」ボタンをクリック

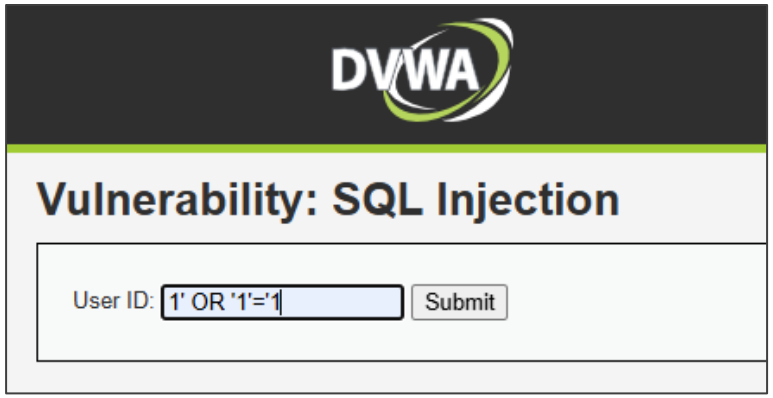


SQLインジェクションを実際に試す②

2. 左側のサイドメニューから、「SQL Injection」をクリック

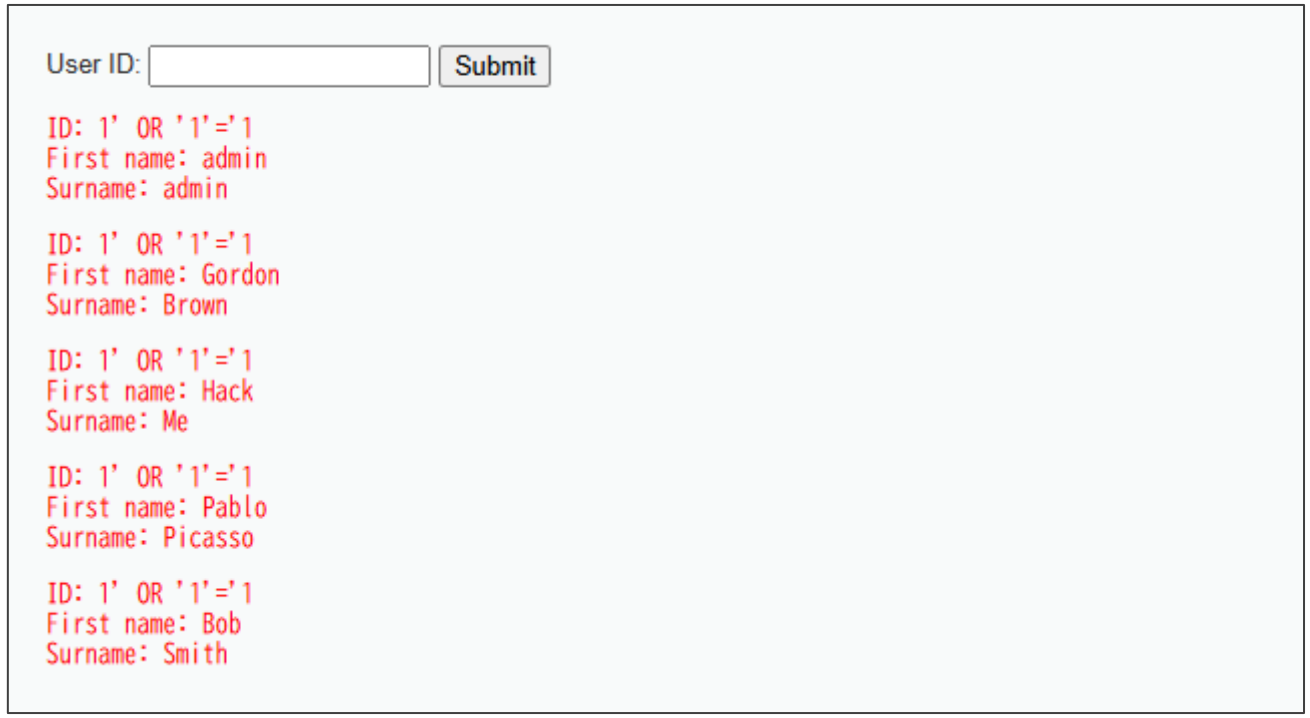


3. User IDフィールドに「1' OR '1'='1」と入力



SQLインジェクションを実際に試す③

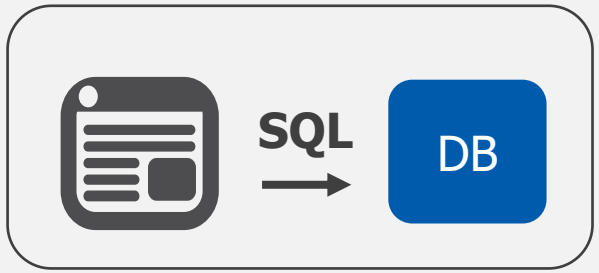
4. 以下のような結果が表示されたら、SQLインジェクションは成功です



SQLインジェクションの解説

User ID:に「1‘ OR ’1‘=‘1 」と入力すると次のようなSQL文がDBに実行されます

```
SELECT * FROM users WHERE id = '1' OR '1'='1';
```



usersテーブルの中から、WHEREの条件に一致する行（レコード）をすべて取得

WHERE句は「条件」を表します

ここの条件は「idが‘1’である」または「‘1’という文字列が‘1’と等しい」
つまり条件は常に満たす true（真） になる式

条件に、'1' OR '1'='1'を入れることで、DBに登録されているusersテーブルのすべての行はこの条件を満たしていることになる
結果的にテーブル内の全データ（ユーザー）が表示される

入力フォーム等にイレギュラーな文字列を意図的に入力することで、不正にデータを取得することが可能
入力する文字列によっては、DBのデータを書き換えることもできてしまう

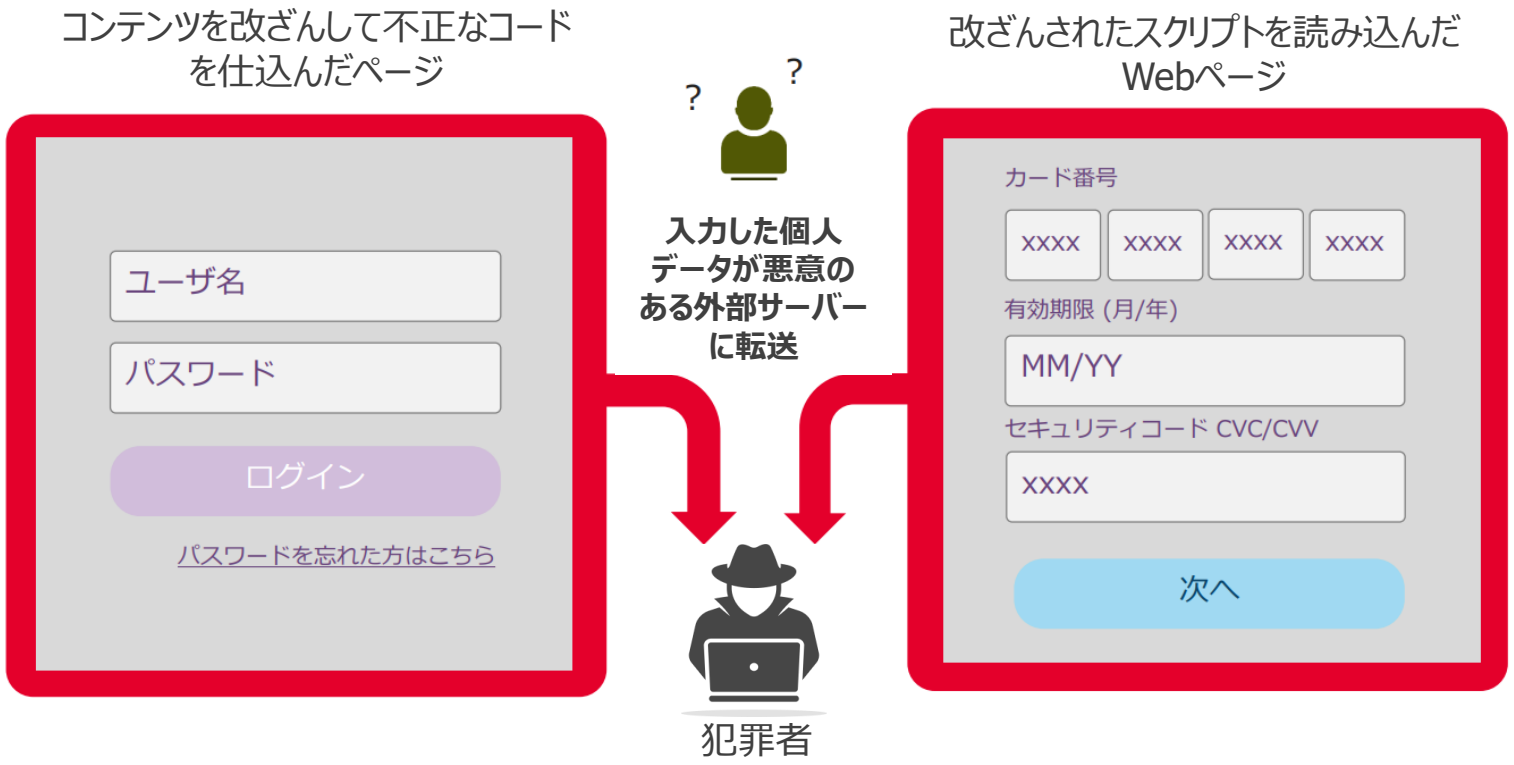
Webスキミングについて

Webサイトに悪意のあるスクリプト（JavaScriptなど）を仕込んで、ユーザーが入力した個人情報やクレジットカード情報などを盗み取る攻撃手法

Webスキミング攻撃は、ユーザーのブラウザ上で実行されるため、一般的なWAFで対策することは困難

以下の点から、攻撃の発生に気が付く事が困難

- **利用者は正規のURLにアクセス出来ている**
- **サイト運営者・管理者側としてもサイト自体は正常に動作出来ている**
- **スクリプトは、ユーザーのブラウザに読み込まれて実行されるため、サーバーには不審な挙動が残りにくい**



Webスキミングを体験

1. 以下のWebアプリのアクセスし、「名前」、「メールアドレス」、「パスワード」を入力し、送信ボタンをクリックします
   注意！　実際の名前やメールアドレス、パスワードは入力しないでください

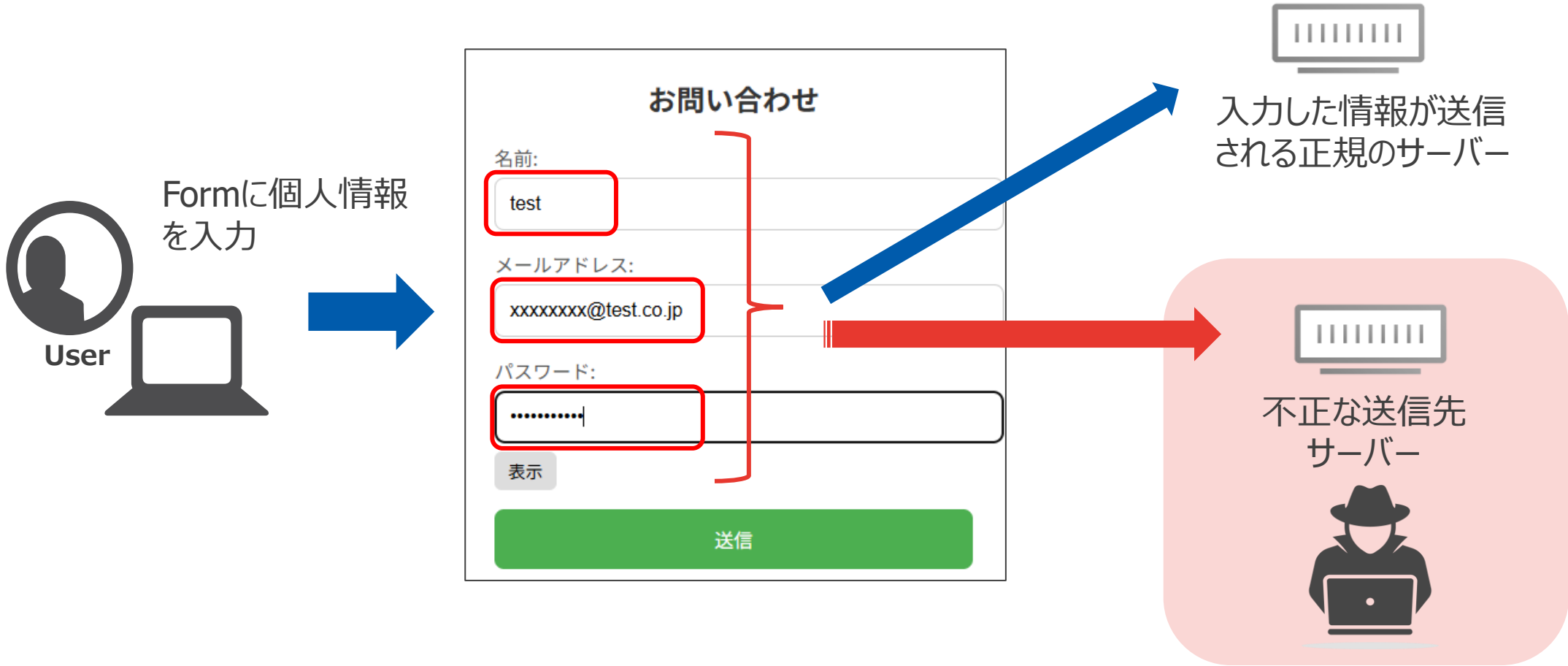
https://handson-csd.xc.dev.tedlab.net/

2. 「送信完了！」というポップアップがでたらOK

Webスキミングの解説①

一見なにか発生したように感じられなかったかと思いますが、**これがWebスキミングの特徴になります**

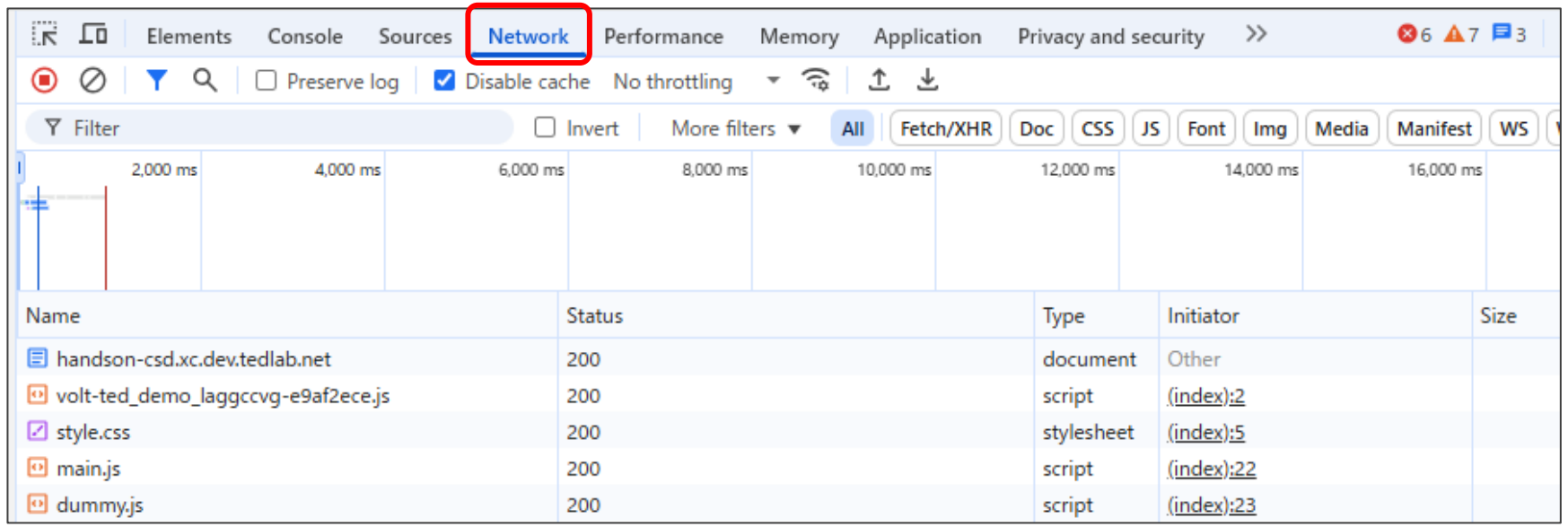
実際は先ほど皆様が入力した情報が不正に外部のサーバーにも飛ばされています



Webスキミングの解説②

ブラウザの開発者モードでJS(Java Script)を確認する

1. ブラウザを立ち上げmキーボードのf12キー押す（Google Chrome/Microsoft Edge/Firefox）
2. Network（ネットワーク）タブをクリックします（以下はGoogle Chromeの画面）



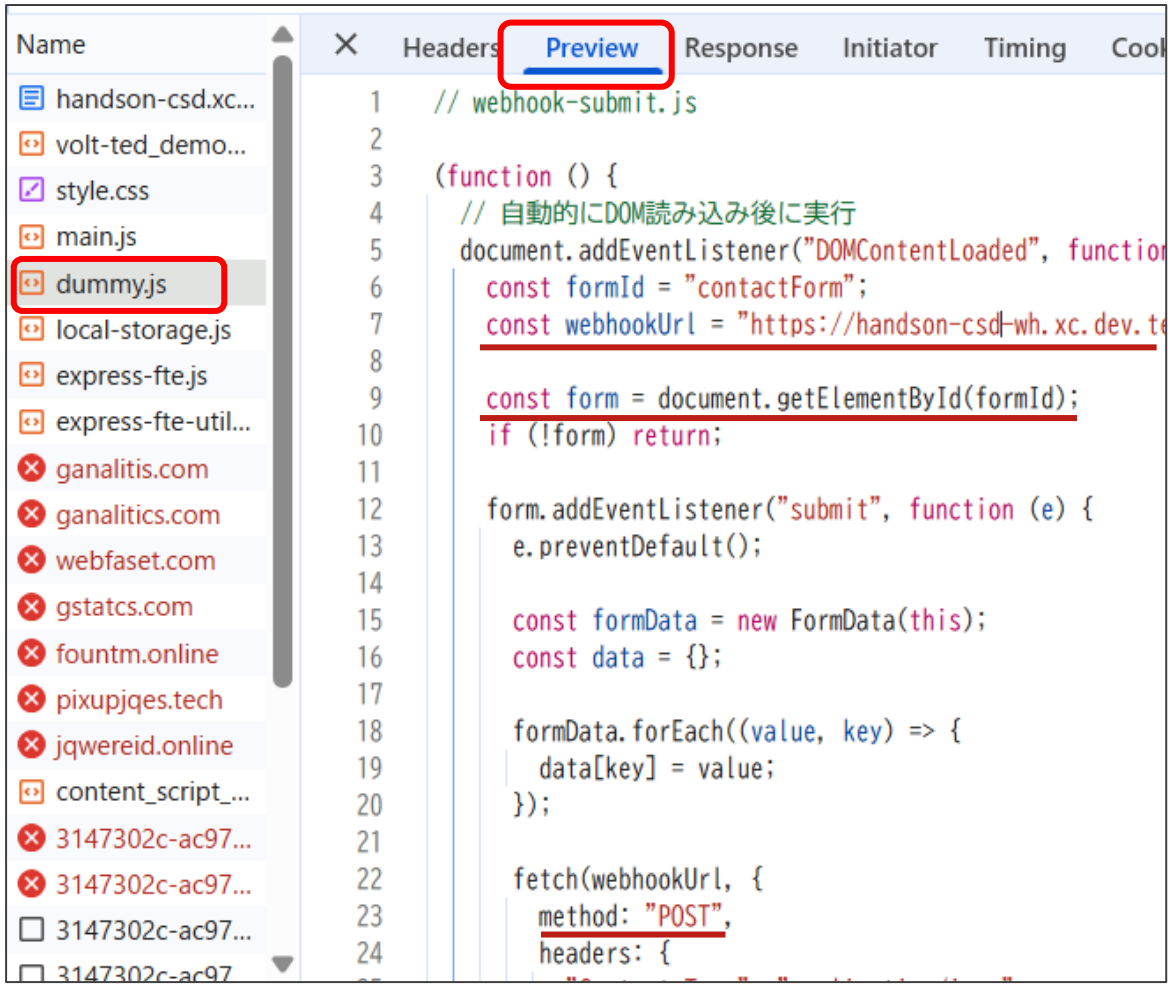
Webスキミングの解説③

以下の手順は、Google Chromeの場合の手順となります。

3. 以下のページにアクセスします
   https://handson-csd.xc.dev.tedlab.net/

4. 開発者モードの画面のName欄にあるdummy.jsをクリックし、右側タブのPreviewをクリックすると、スクリプトの内容を確認できます

5. スクリプトの内容を簡単に説明すると、Form入力した情報を取得し、POSTでwebhookUrlに送信しているという内容になります



今回は分かりやすいように、JS名を怪しい名前にしたり、送信先のURLをそのまま記載しています

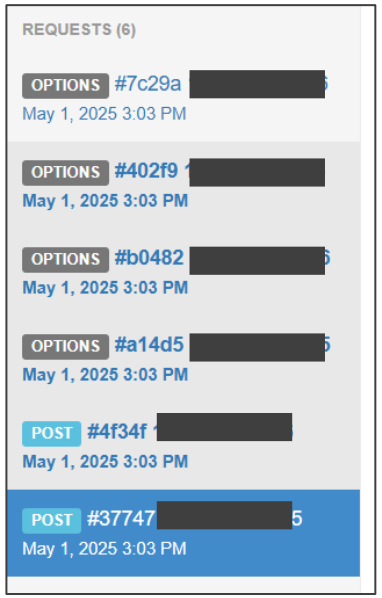
実際はJS名を最もらしい名前にしたり、送信先のURLを難読化・類似ドメインにするだけでも、ユーザーからは気づかれにくくなります

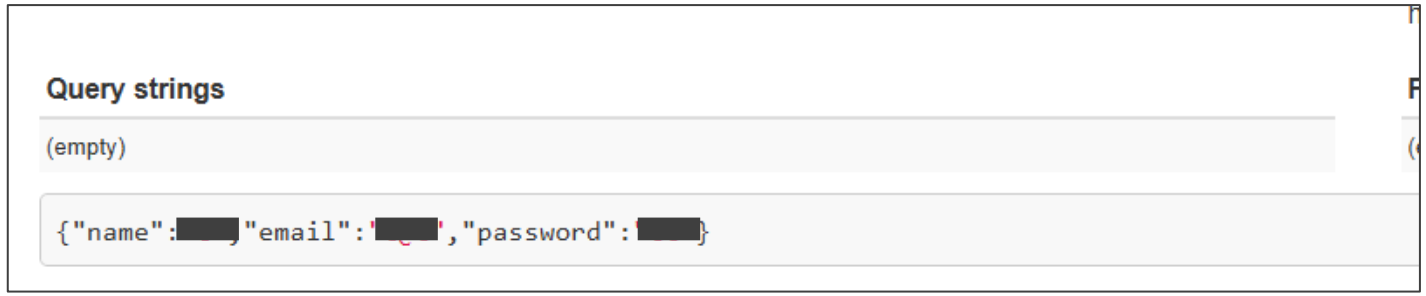
Webスキミングの解説④

次に実際に送信された情報を確認してみる

1. 以下のページにアクセスします（当日にお知らせします）
   https://handson-csd-wh.xc.dev.tedlab.net/#!/xxxxxxxxxxxxxxxxxxxxxxxx

2. 画面左のREQUESTからPOSTを選択し、画面下のQuery Stringsをみることで、Formで送信したデータをみることができます





Webアプリケーションを狙った代表的な攻撃手法のまとめ

DDoS攻撃

大量の通信を実施して、ネットワークの帯域やサーバーのリソースを枯渇させて、サービスを停止させる

SQLインジェクション

フォームにイレギュラーな文字列を意図的に入力することで、不正にデータを取得・改ざんするような攻撃

Webスキミング

Webサイトに悪意のあるスクリプト（JavaScriptなど）を仕込んで、ユーザーが入力した個人情報やクレジットカード情報などを盗み取る攻撃手法

次はこれらの攻撃からアプリケーションを守る、WAF（クラウドWAF）について紹介

WAF（クラウドWAF）について

WAFとは

WAF: Web Application Firewall（ウェブアプリケーションファイアウォール）の略

ウェブアプリケーションを狙った攻撃から守るために、通信を監視・制御して不正なリクエストをブロックするソリューション

WAFの検知方法（どうやって攻撃を見つけるのか）は、主に以下の2つになります

① シグネチャベース（パターンマッチング）

既知の攻撃パターン（SQLiやXSSなど）に一致するかどうかをチェックする

例えば、**クライアントからのリクエストに「'id=1' OR '1'='1'」が含まれていたら、攻撃と判断しブロックする**という動きです

このような攻撃パターンをそれぞれ**シグネチャ**と呼び、その**シグネチャに一致したらブロック**する方法です

SQLインジェクションなどの攻撃は、主にシグネチャベースで検知・ブロックします

② 挙動ベース（アノマリベース）

通常のリクエスト頻度や特定IPからのアクセス傾向など、普段とは異なったアクセス傾向から、攻撃と判断してブロックする方法です

最近では、AI（機械学習）を利用した検知・ブロック方法があります

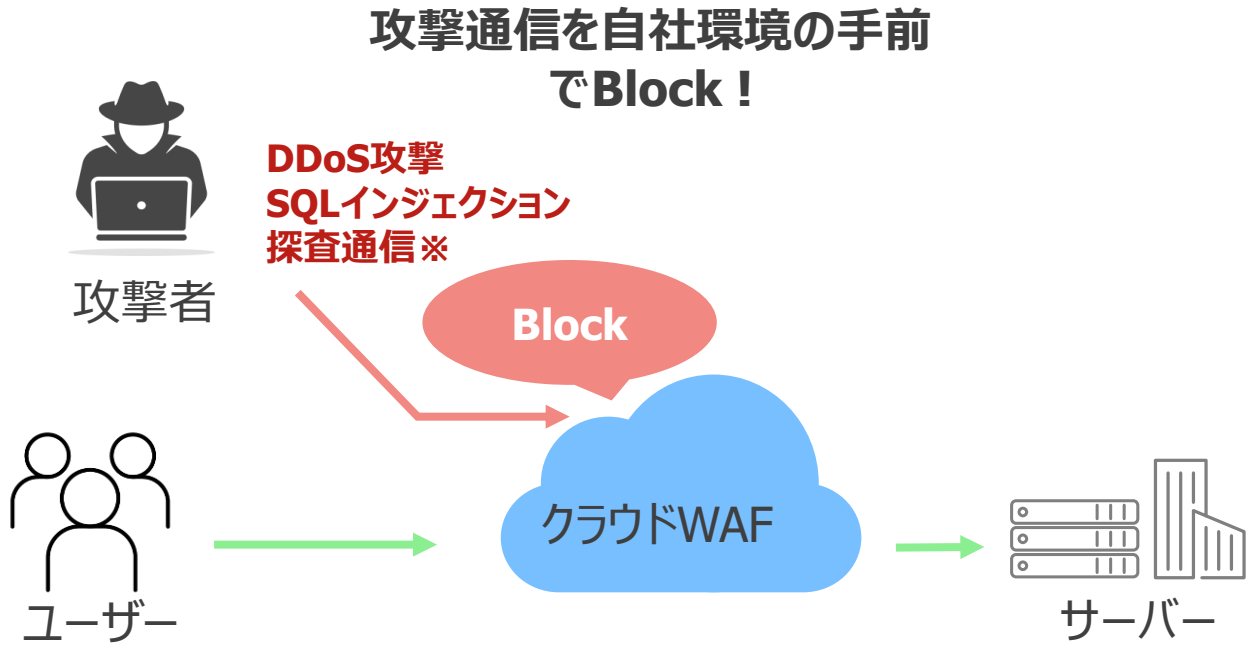
クラウドWAFとは

クラウドWAF：
クラウドサービスとして提供されるWAF

また、**多くのクラウドWAFにはDDoS保護機能が標準で組み込まれているか、追加機能として提供されています**

クラウドWAFのメリット

- **自社の環境に攻撃通信が届かない**
  - DDoS攻撃対策として効果的
  - 探査通信をさせない※
  - 攻撃を防ぐために自社環境のサーバーやネットワークのリソースも消費しない
- **運用時の負担が少ない**
  - シグネチャはメーカーが管理・更新
- 導入までの期間が短くコストも抑えられる



※攻撃対象となるシステムのOS、ソフトウェア、IPや開いているPortなどの情報を収集する通信
取得した情報は攻撃するための準備や脆弱性の特定に利用される

F5 XC WAF（WAAP）ソリューションについて

WAAP(Web Application and API Protection)とは

2017年に提唱された次世代のWebセキュリティ概念

Webアプリケーションやモバイルアプリで API利用の増加に伴い、近年高度化するサイバー攻撃に対して
従来型のWAFだけでは対策は不十分
→APIの保護も考慮したセキュリティ対策が必要

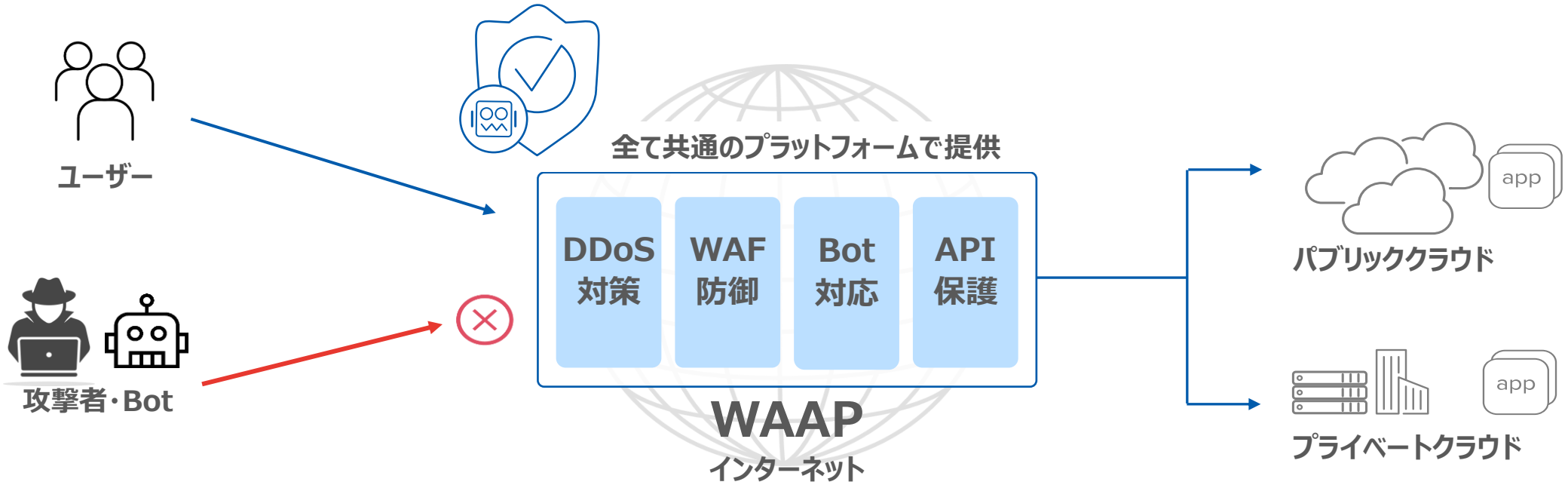
WAAPをWAF市場の進化として定義しており、XC WAAPにおいては以下の機能をコアとする

DDoS対策	ネットワーク及びアプリケーションレベルのリソース保護
次世代WAF	アタックシグネチャ自動更新、クライアントの振舞い学習
Bot対応	Bot及びツールの検証。振舞いを把握し悪意のあるBotを検知
API保護	APIディスカバリとリクエスト毎の異常確認

※Gartnerは４項目の細かい機能を明示していません。上記一部はF5の実装です。

F5 XC WAF（WAAP）ソリューションについて

- クラウド型のWAF（WAAP）ソリューション
- **標準でL3/L4、L7のセキュリティ機能を適用可能**
  - **DDoS攻撃やSQLインジェクション攻撃などの攻撃から保護**
- **Webスキミング対策機能を提供**



※RE（Regional Edge）は、F5 XCのグローバルに分散配置されたPoP（Point of Presence）を指します

F5 XC WAF（WAAP）ソリューションの特徴

攻撃は自社サイトに届く前に無効化

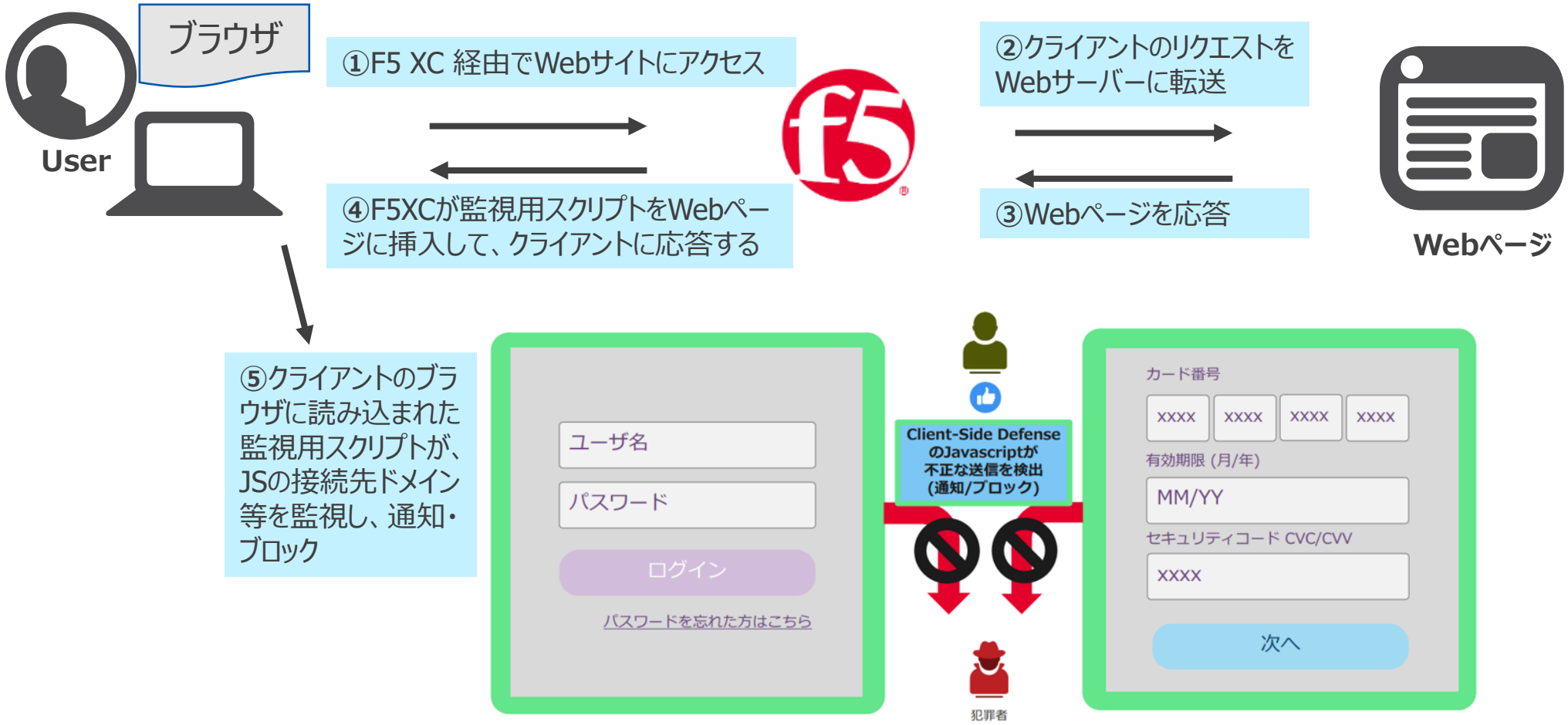
サイバー攻撃を防御するオンプレ製品はリソースに限界があり、クラウド製品でも高額課金の恐れがある

- 世界中にある多数のPoP拠点からWAAP機能を提供
- 攻撃をアクセス元の最寄りのデータセンターでブロック
- L3-L7のあらゆる攻撃に対応、DDoS攻撃保護機能は自動で有効

Webスキミング対策：Client-Side Defense（CSD）検出・保護の仕組み

F5 XC CSDがクライアントのJavaScriptの挙動を監視することで、不正な通信をブロックさせる機能

F5 XC CSDは、クライアントにWebページのコンテンツを応答するときに、監視用のスクリプトを埋め込むことで、本対策を実現



ハンズオン実施前の事前知識

- F5 XC WAF（WAAP）の基本的な考え方について

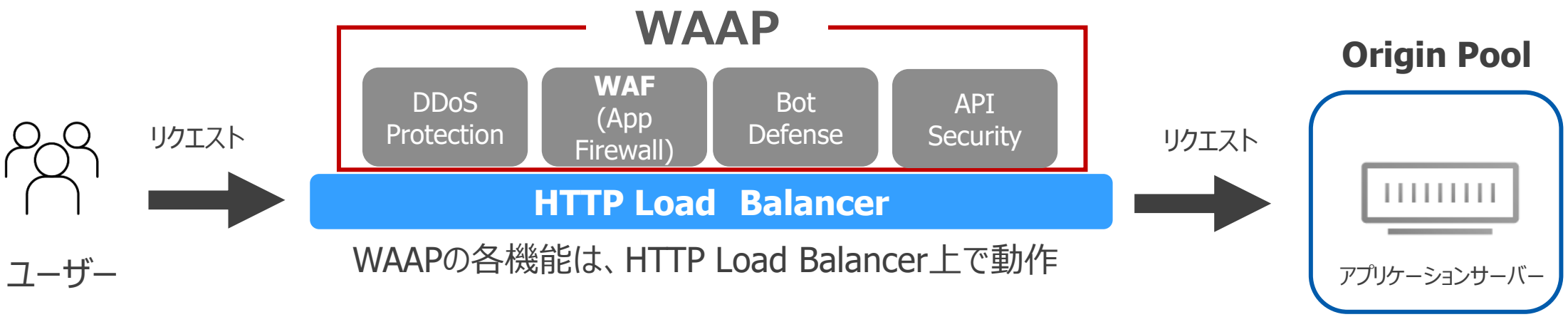
F5 XC WAF（WAAP）ソリューションの構成要素

次の3つのオブジェクトが必要になります

1. **HTTP Load Balancer**
   - クライアントからのリクエストを受付けるための設定
   - HTTP Load Balancer上でWAF(WAAP)機能が動作

2. **Origin Pool**
   - アプリケーションサーバーがOrigin Poolに相当します
   - HTTP Load Balancer は、受け取ったリクエストを実際に処理を行うサーバーへ転送します

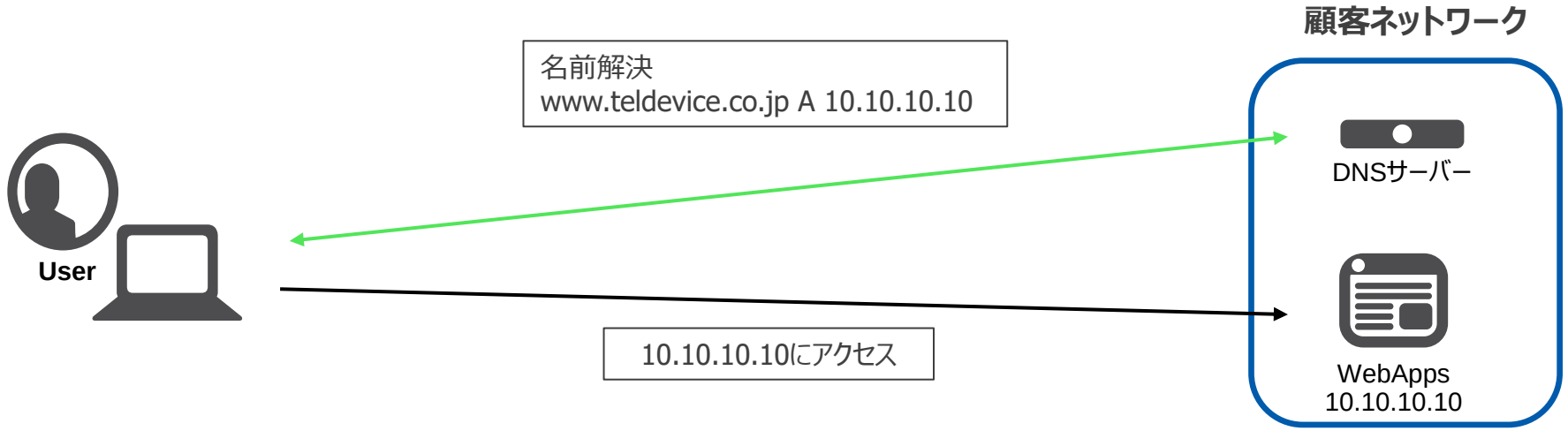
3. **App Firewall**
   - WAFのセキュリテイポリシーに相当します



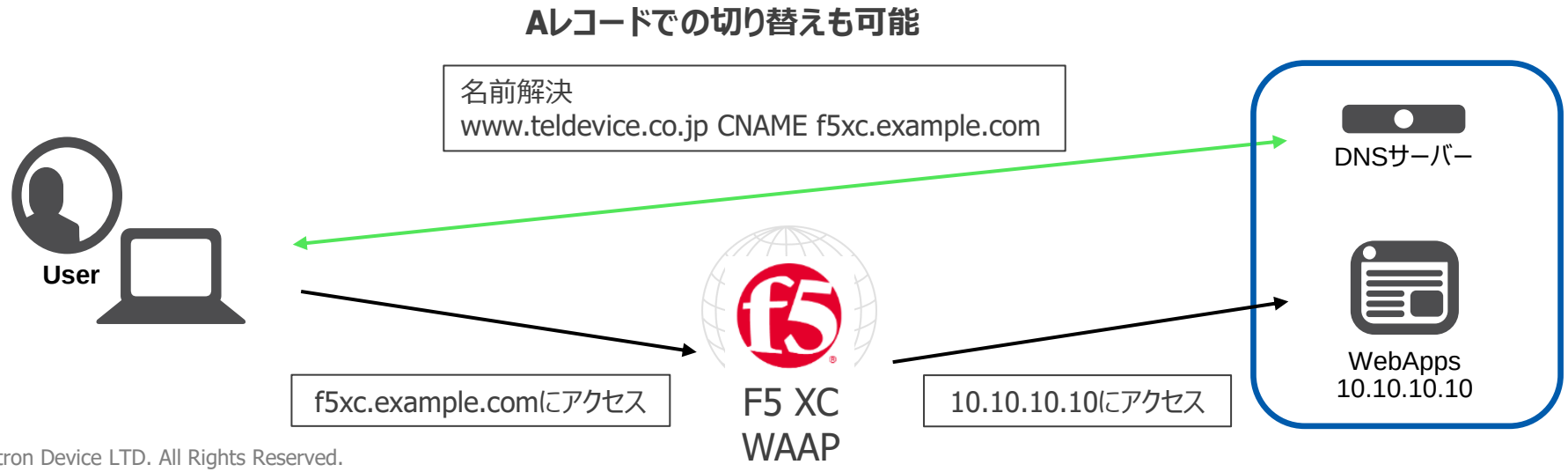
F5 XC WAF（WAAP）ソリューションの導入方法

DNSの切り替えによる導入

導入前



導入後



F5 XC WAF(WAAP)のDNSレコード登録について

基本的には、該当ドメインのDNSレコードの値を変更することでF5XC WAFを適用いたします

なお、F5 XCは権威DNSのサービスも提供しており、F5XC DNSに権限移譲しているドメインであれば、F5XC WAF(HTTP LB)を作成すると、自動的にAレコードが払い出されます（下図の②のパターン）

① 権限移譲していないDomainの場合

以下の2つのパターンがあります

1. Aレコードを利用
   F5 XC から払いだされるIPを、お客様管理のDNSサーバーに
   Aレコードとして登録します（オプション）

2. CNAMEを利用
   Load Balancer作成後に、CNAMEレコードが払い出されます
   そのCNAMEレコードをお客様管理のDNSサーバーに登録します

1. www.demo.teldevice.co.jp IN A xxx.xxx.xxx.xxx

or

2. www.demo.teldevice.co.jp IN CNAME ves-io-xxx.ac.vh.ves.io

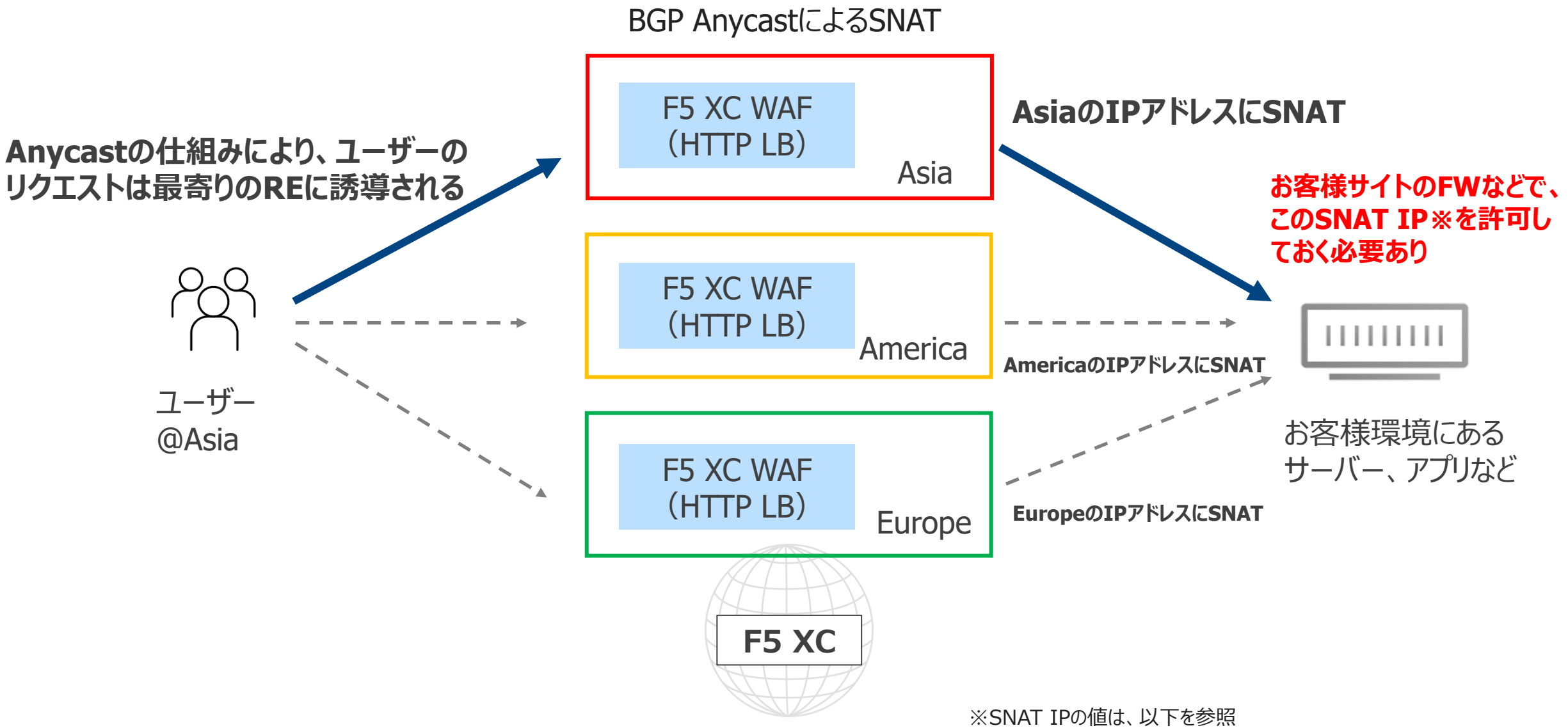
②権限移譲しているDomainの場合

Load Balancer作成時に該当のドメイン名を割り当てることで、
自動的にAレコードが払い出されます

F5 XC DNSは、
demo.teldevice.co.jp.ドメインを管理

ハンズオントレーニングでは②の方法を利用

クライアント通信のSource NAT（SNAT）について



※SNAT IPの値は、以下を参照
https://docs.cloud.f5.com/docs/reference/network-cloud-ref

HTTP Load BalancerのSSL/TLS証明書登録

WAFを適用するにはHTTPS（暗号化）通信をHTTP Load Balancerで終端（復号）する必要があります

終端するには、HTTPS Load BalancerにSSL/TLS証明書の登録をする必要がありますが、F5 XC には次の3つのパターンがあります

1. **お客様持ち込みのSSL/TLS証明書を利用**
   お客様のオリジンサーバーでご利用されている証明書/鍵を登録します

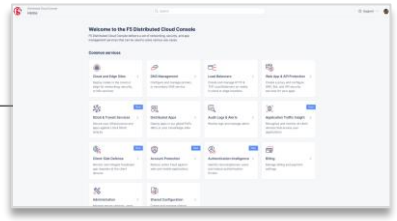
2. **F5 XC 自動発行の証明書を利用（ACMEレコードを手動で登録）**
   HTTP Load Balancerを作成後に払い出されるACMEレコードをDNSサーバーに登録することで、
   Let's Encryptの証明書が発行され、適用されます

3. **F5 XC 自動発行の証明書を利用**
   F5 XC DNSに権限移譲しているドメインの場合、Load Balancer作成時に自動でLet's Encryptの証明書が発行され、適用されます

ハンズオントレーニングでは3つめの方法を利用

ハンズオントレーニング環境について

ハンズオントレーニング環境



F5 XC コンソール画面

F5 XC 上にF5 XC WAF（HTTP LB）をデプロイ

F5XC WAFで保護するサーバー

AWS

F5 XC

DDoS 対策

WAF 防御

Bot 対応

API 保護

HTTP Load Balancer

リクエスト

HTTPS

攻撃

検証ユーザー

HTTP

dvwa container x 10
Port:8081 ~ 8090

Ubuntu 22.04

クライアントのアクセス先
<任意のホスト名>.xc.dev.tedlab.net

http://dvwa.app.dev.tedlab.net:80xx

ハンズオントレーニング環境

No.	ユーザー名	ドメイン名	dvwaコンテナのポート
1		<任意のホスト名>.xc.dev.tedlab.net	8081
2		<任意のホスト名>.xc.dev.tedlab.net	8082
3		<任意のホスト名>.xc.dev.tedlab.net	8083
4		<任意のホスト名>.xc.dev.tedlab.net	8084
5		<任意のホスト名>.xc.dev.tedlab.net	8085
6		<任意のホスト名>.xc.dev.tedlab.net	8081
7		<任意のホスト名>.xc.dev.tedlab.net	8082
8		<任意のホスト名>.xc.dev.tedlab.net	8083
9		<任意のホスト名>.xc.dev.tedlab.net	8084
10		<任意のホスト名>.xc.dev.tedlab.net	8085

<任意のホスト名>には、ご自身の名前等をご入力ください。
例）taro.xc.dev.tedlab.net

ハンズオントレーニング

ハンズオントレーニングの流れ

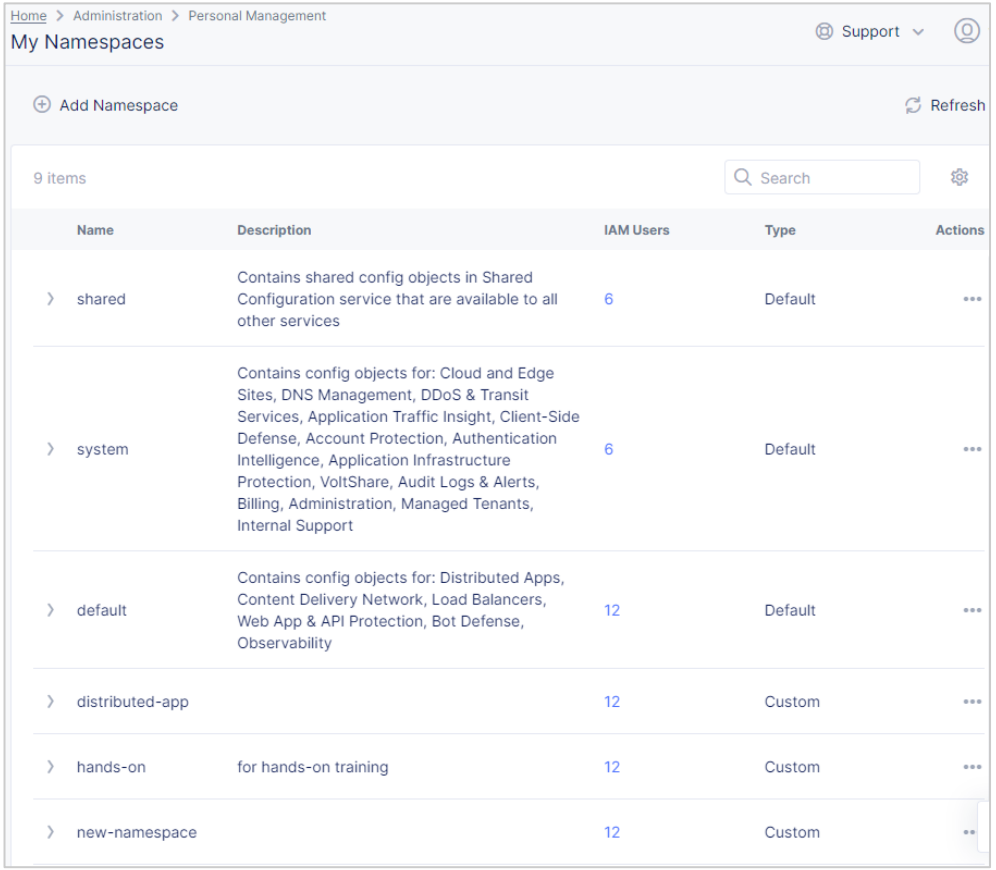
- Namespace の説明
- Health Checkの作成
- Origin Poolの作成
- HTTP Load Balancer の作成
- Routes の説明/設定
- WAF Policy の作成/チューニング
- デモ：Webスキミング対策（Client-Side Defense）について

ハンズオントレーニング

- Namespace の説明
  ユーザー・チーム・サービス単位でリソースを分けて管理するための論理的なスペースです

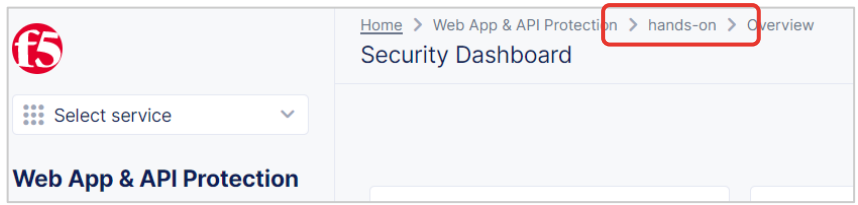
Namespaces の説明

- テナント制御が可能
- ユーザー単位でアクセスできるNamespaceを制御

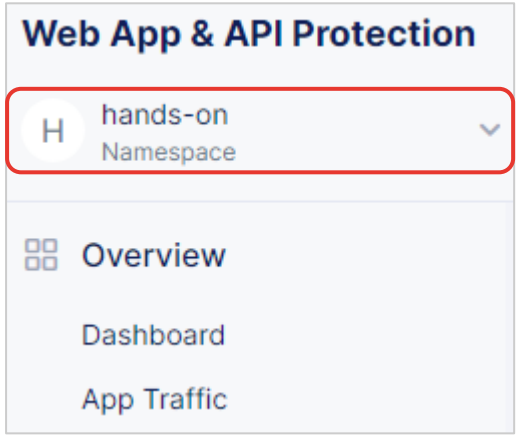


本トレーニングでは hands-on を使用します

ログインして設定投入時には Namespace が hands-on になっていることが確認できます

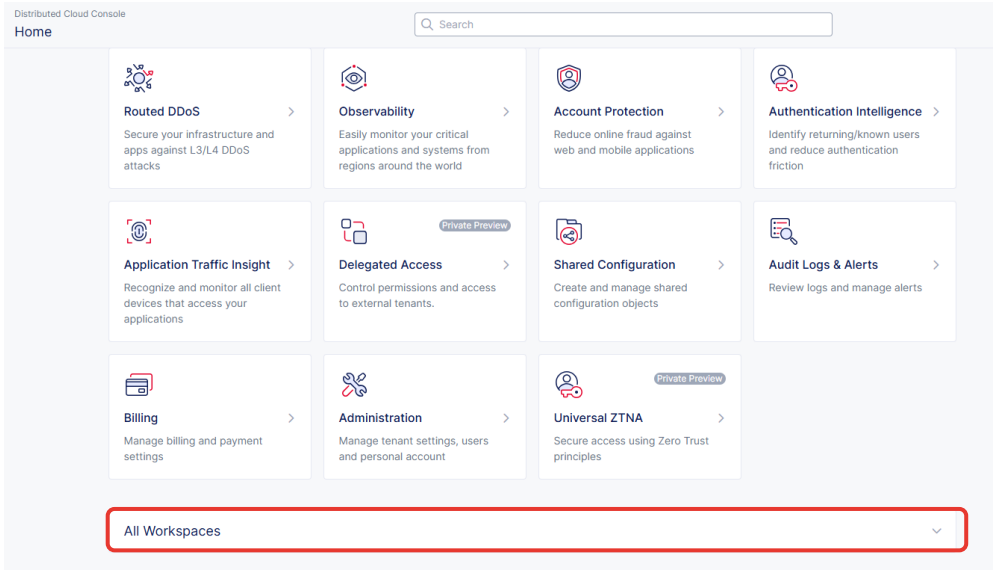


管理者としてログインされた場合には右のように選択が可能になります。

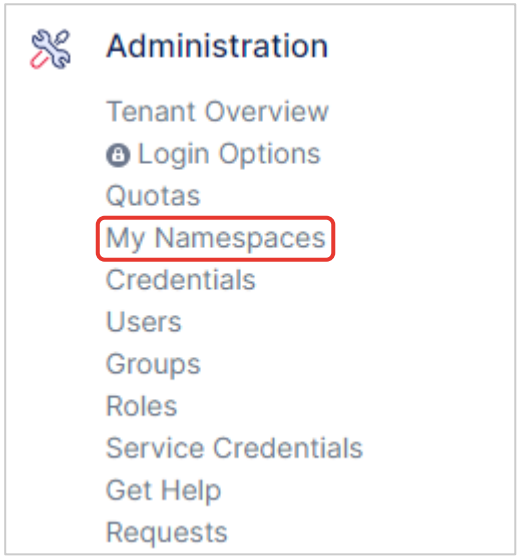


Namespace の作成①（参考）

Home メニューの All Workspaces を展開します

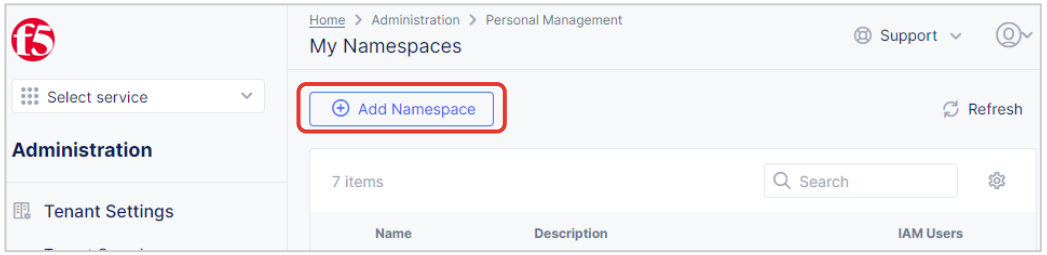


Administration の My Namespaces を選択します

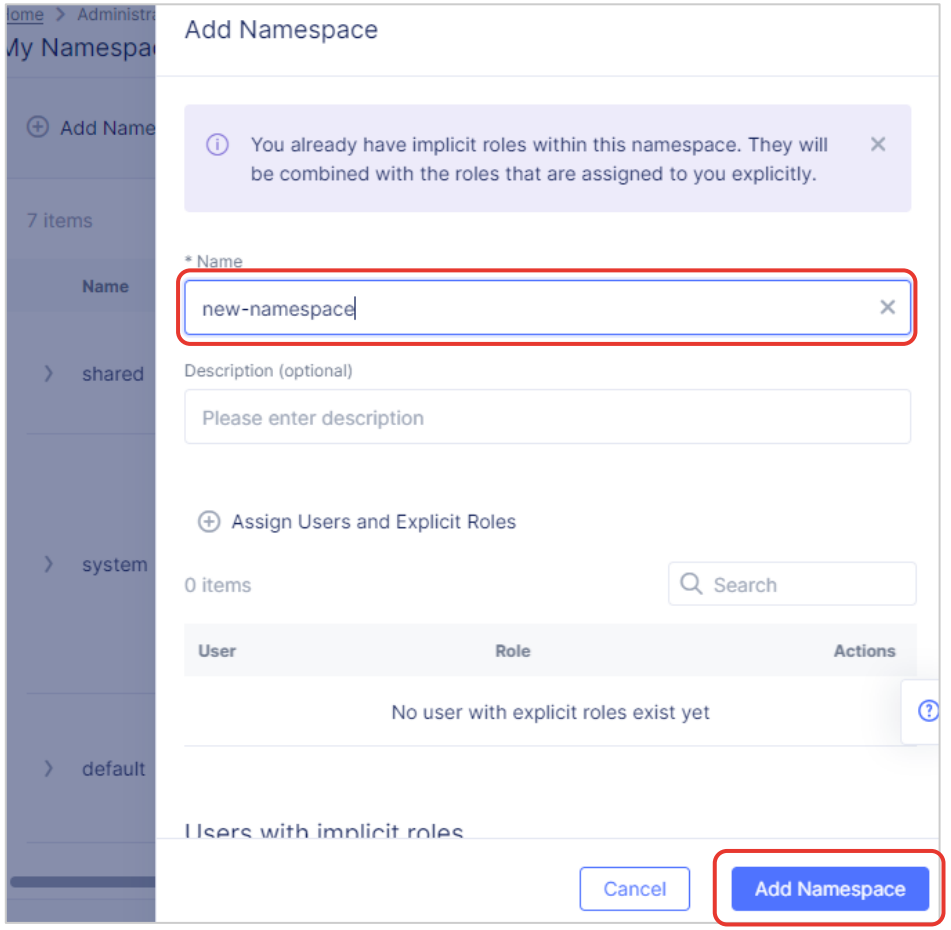


Namespace の作成②（参考）

Add Namespace を押下します



Name に Namespace の名称を入力し、
Add Namespace を押下します



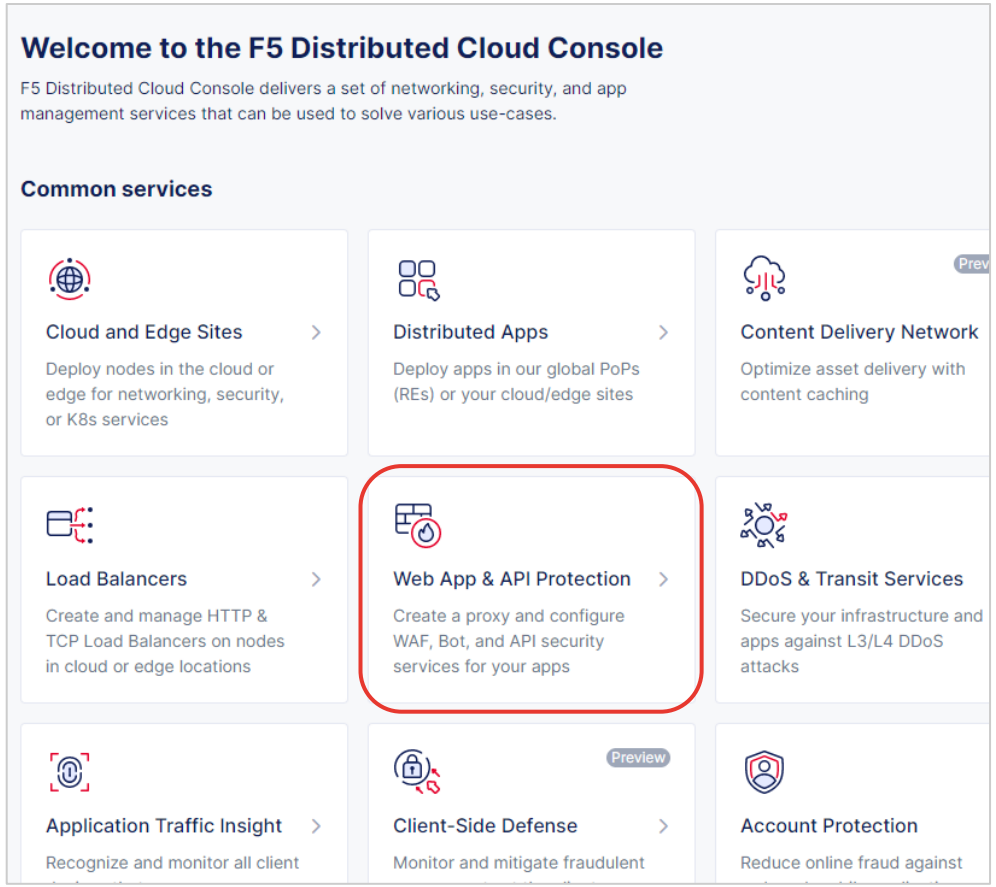
ハンズオントレーニング

- Health Checkの作成

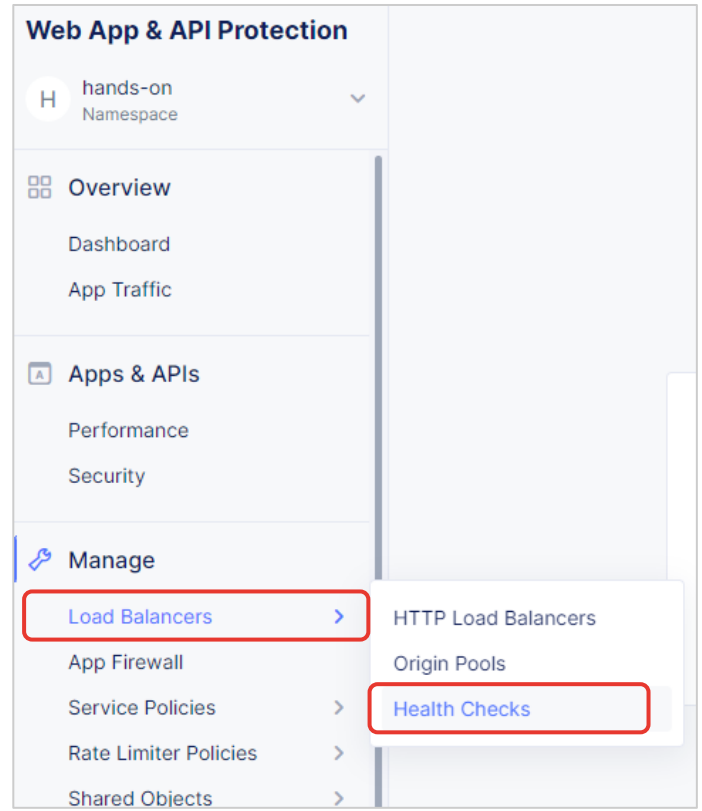
  Health Checkとは、サーバーやアプリケーションが正常に動作しているかどうかを確認するためのものです
  Health Checkで正常に動作できていないと判断した場合は、そのサーバーに対してリクエストを送信しません

Health Check の作成①

Home メニューから Web App & API Protection を選択します

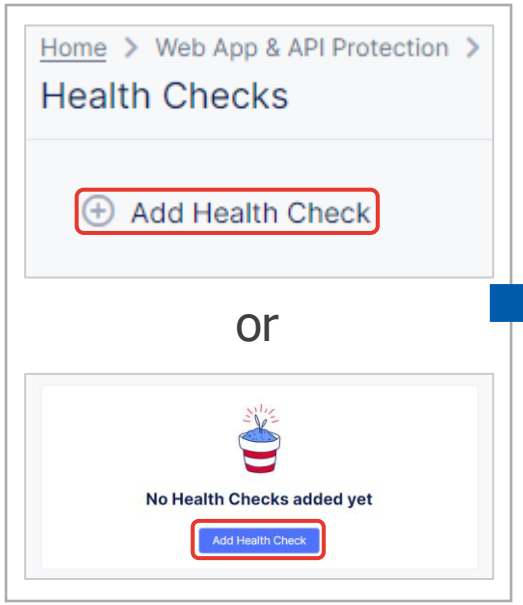


左ペインから
[Manage] – [Load Balancers] – [Health Checks]
を選択します

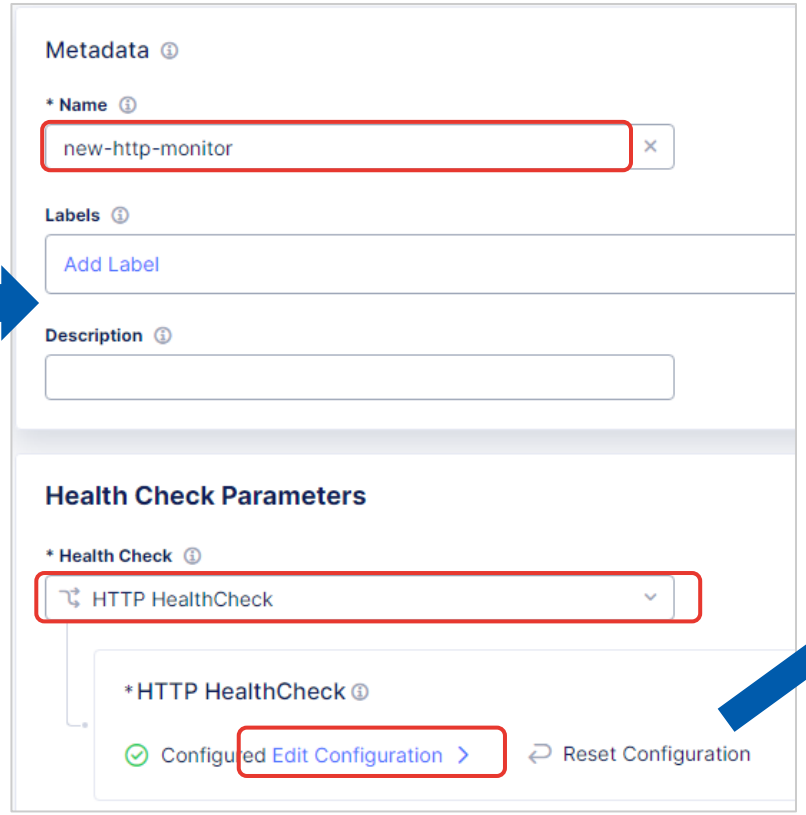


Health Check の作成②

Add Health Check を押下します



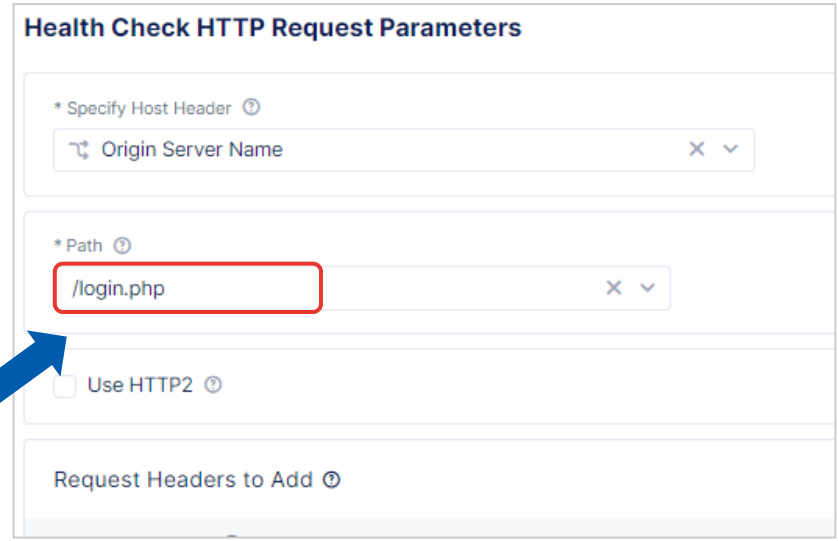
各パラメータを指定します



- Name
  - ➢ 任意（識別できる名称）
- Health Check
  - ➢ HTTP HealthCheck

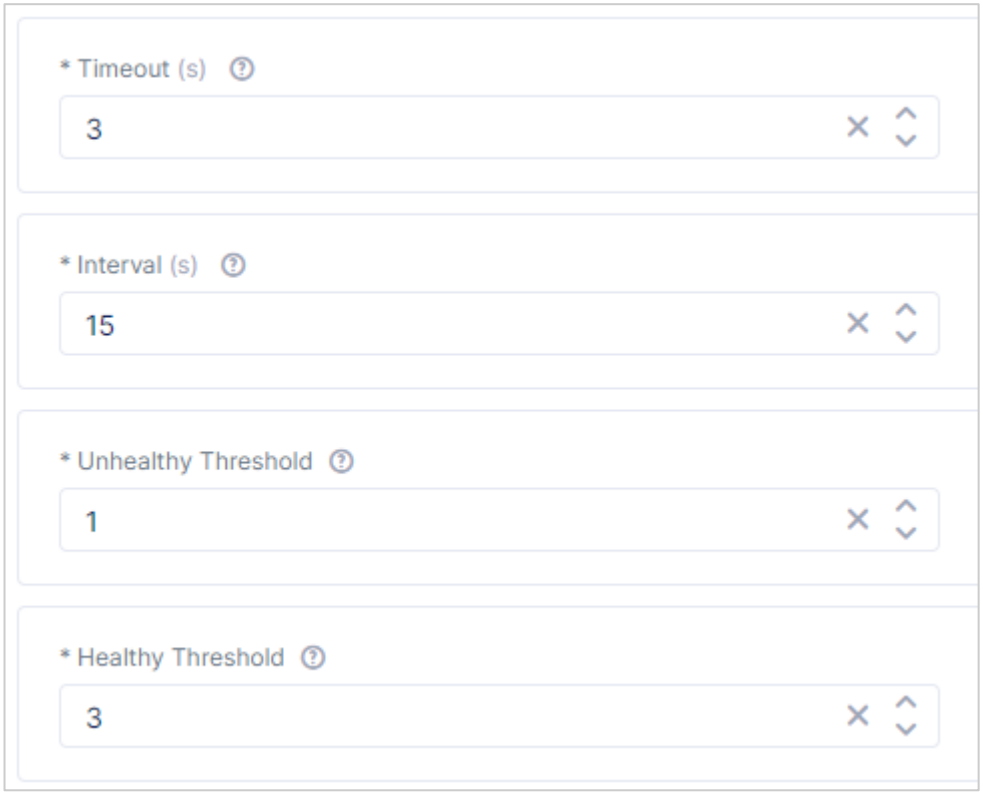
Path を指定して Apply を押下します

- Path
  - ➢ /login.php



パラメータを指定後 Add Health Check を押下して保存します

Health Check のパラメータ（参考）



- Timeout
  Health Check を試行してから正常な応答を待つ時間
- Interval
  Health Check を試行する間隔
- Unhealthy Threshold
  DOWNをマークするまでの Health Check の失敗回数
- Healthy Threshold
  UPをマークするまでの Health Check の成功回数

Interval 15 / Healthy Threshold 3 の場合、Origin Server が実際に UP してから 30～45 秒後に UP がマークされる（通信が復旧する）動作になります。

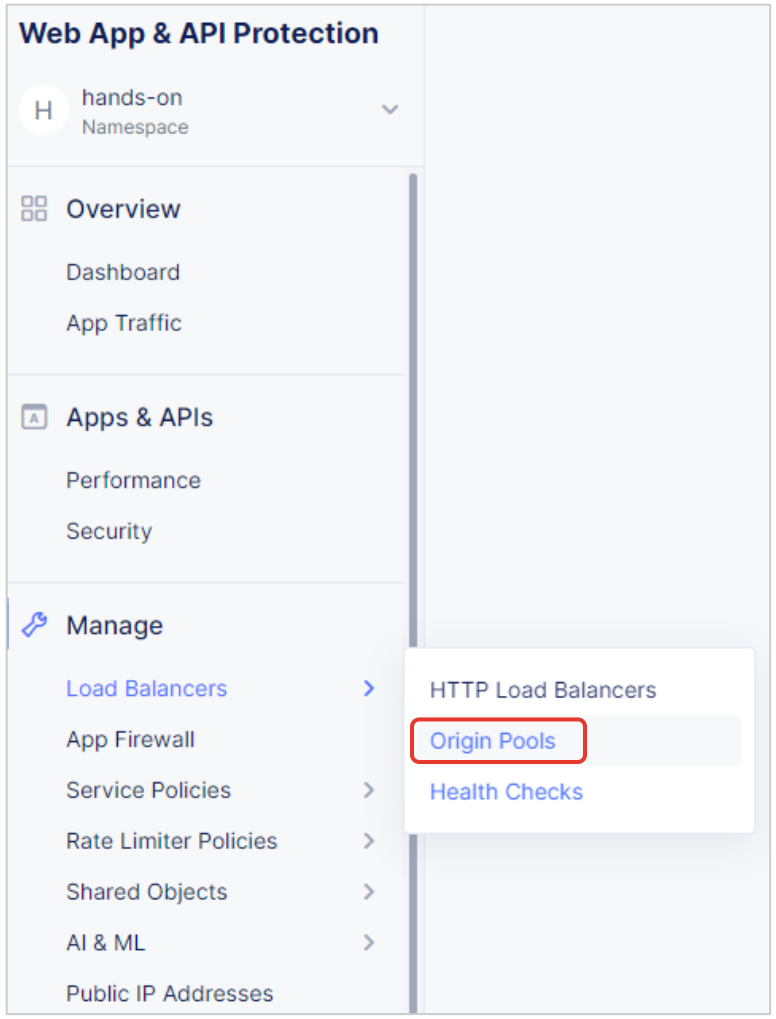
Health Check を設定すると、各リージョンから Origin Server への Health Check が行われます
また、リージョン単位で見ても複数の送信元から Health Check が行われます

ハンズオントレーニング

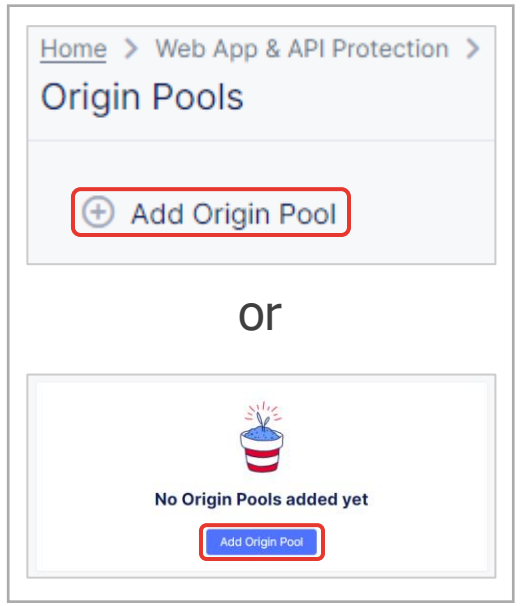
- Origin Poolの作成
  Origin Poolは、クラウドWAFがリクエストを受け取った後の転送先となるサーバーになります
  複数のサーバーを設定することも可能です

Pool の作成①

左ペインから
[Manage] – [Load Balancers] – [Origin Pools] を選択します

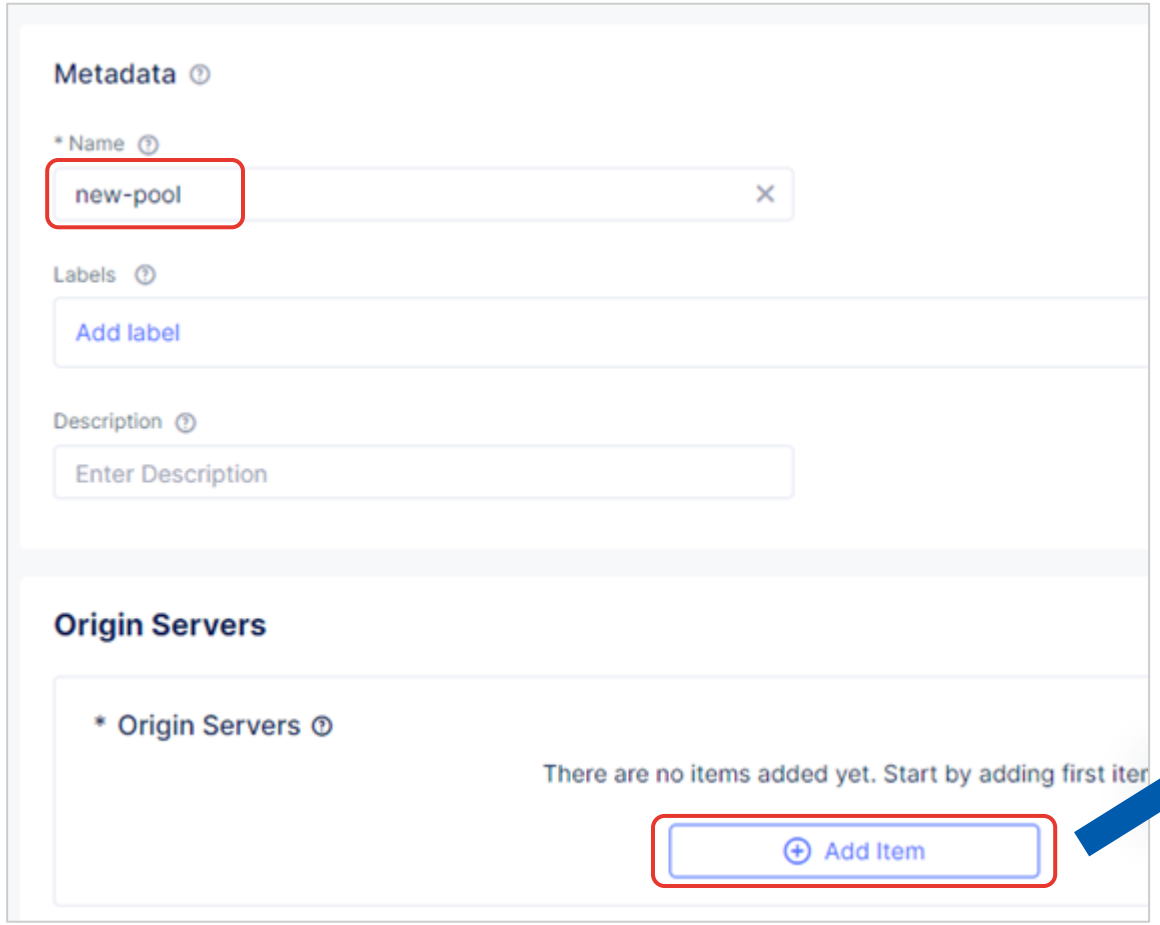


Add Origin Pool を押下します



Pool の作成②

各パラメータを入力し、Origin Servers の作成を行います



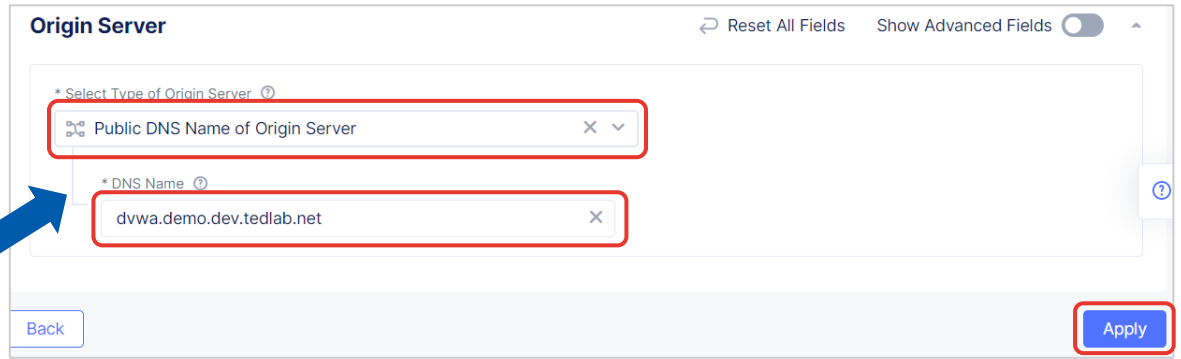
- Name
  - 任意（識別できる名称）
- Origin Servers

  Add Item を押下して Origin Server の設定画面に入ります
  - Select Type of Origin Server:

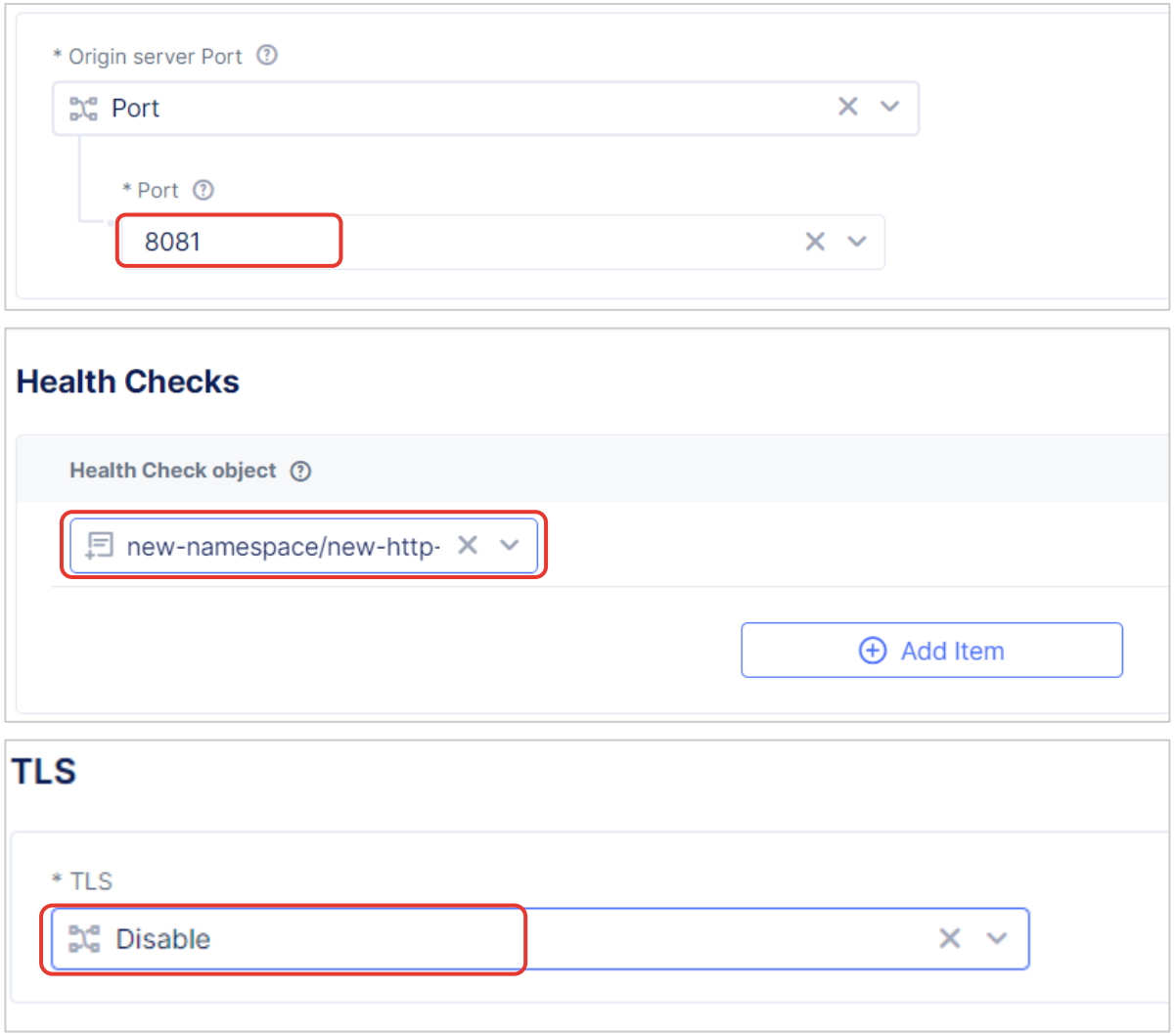
    Public DNS Name of Origin Server
  - DNS Name:

    dvwa.demo.dev.tedlab.net

  Apply を押下して Origin Server の作成を完了します



Pool の作成③



主要パラメータを指定します

- Port
  - 80XX
    （ご自身に割り当てられたdvwaのport番号）
- Health Checks
  - 作成した Health Check
- TLS
  - Disable

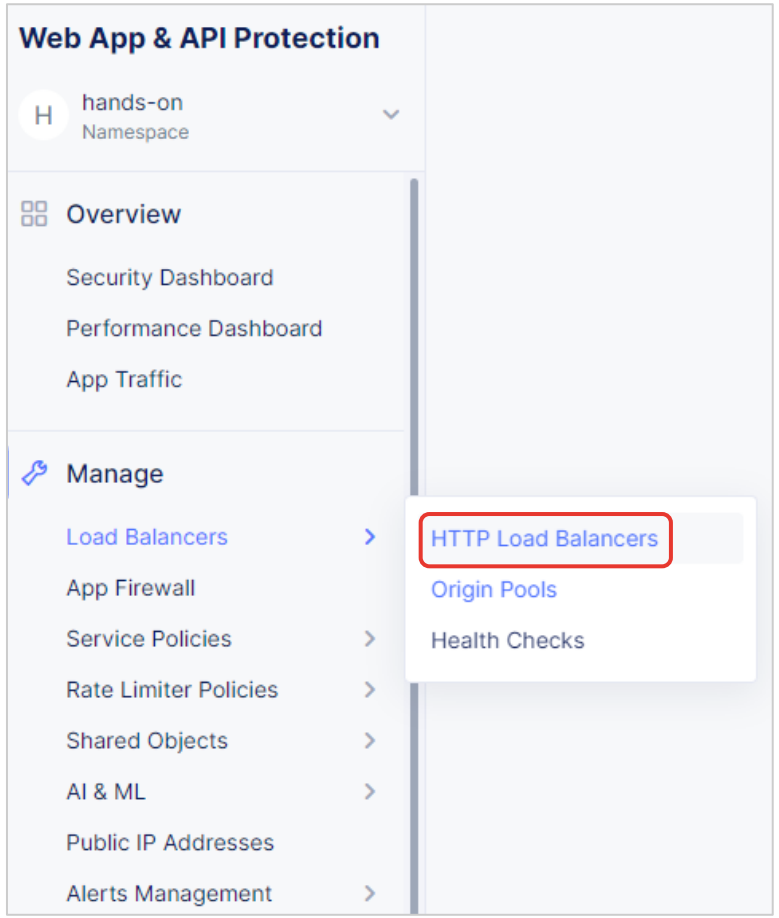
Add Origin Pool を押下して Pool の作成を完了します

ハンズオントレーニング

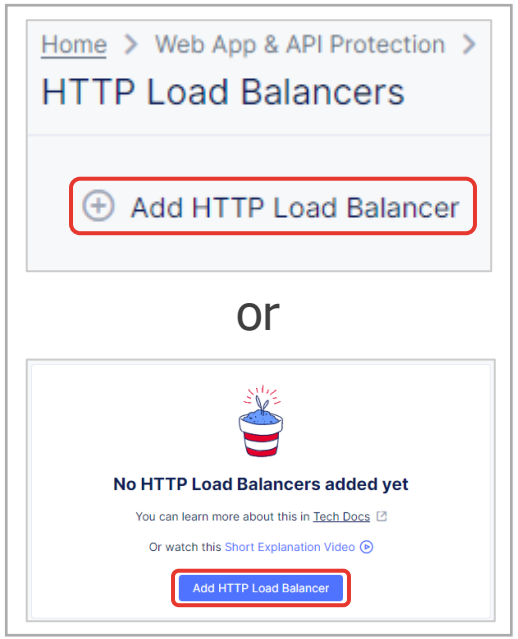
- HTTP Load Balancer の作成
  クライアントからのリクエストを受付けるための設定になりますHTTP Load Balancer上でWAF(WAAP)機能が動作します
  なお、受け取ったリクエストはOrigin Poolに転送されます

HTTP Load Balancer の作成①

左ペインから
[Manage] – [Load Balancers] – [HTTP Load Balancers]
を選択します

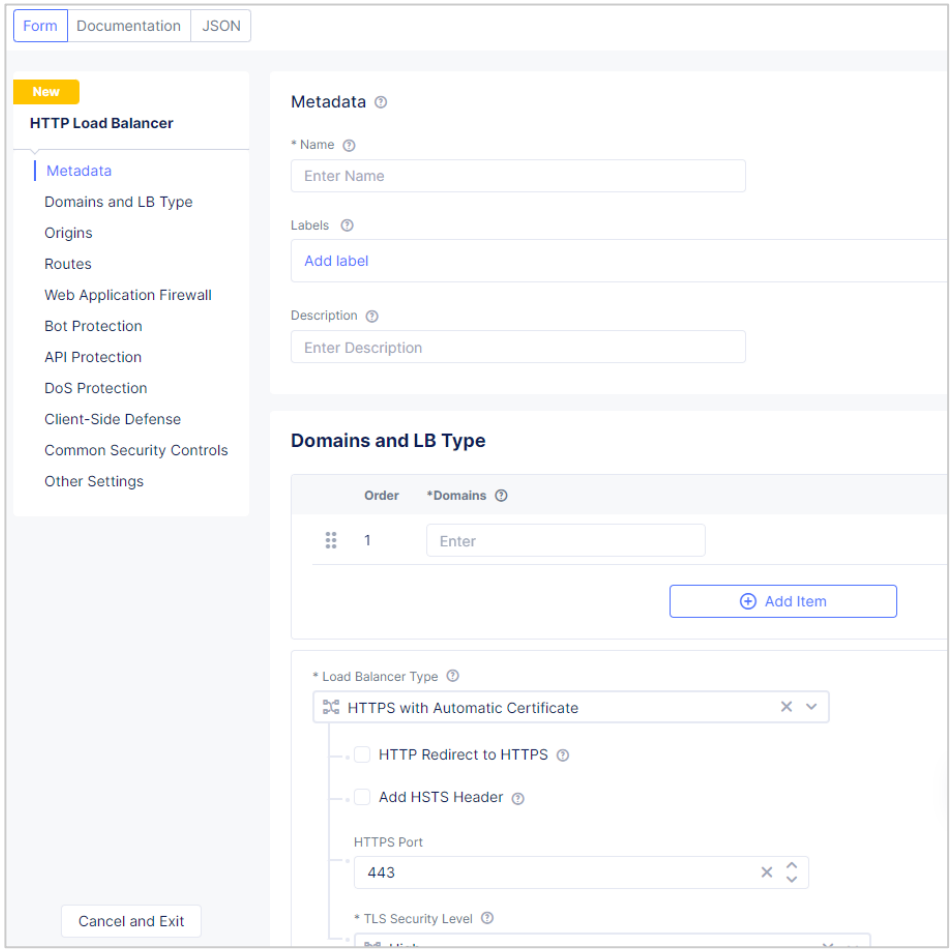


Add HTTP Load Balancer を押下します



HTTP Load Balancer の作成②

Routes, Web Application Firewall 以外の基本的なパラメータを設定します
パラメータを入力後、 Save and Exit を押下して保存します

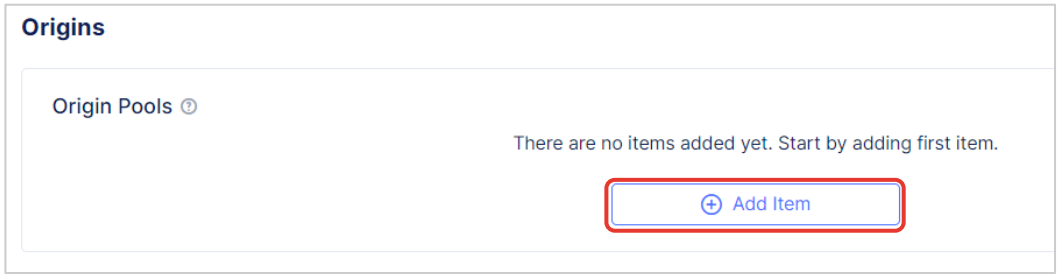


- Name
  - HTTP Load Balancer の名称を指定します
    - 任意（識別できる名称）
- Domains
  - FQDN を指定します
    - <host>.xc.dev.tedlab.net
- Load Balancer Type
  - HTTPSの場合、証明書を自動生成するか手動インポートかを選択できます
    - HTTPS with Automatic Certificate
- HTTPS Port
  - Load Balancer で受け付けるポート番号を指定します
    - 443（デフォルト）
- Origin Pools
  - 作成した Pool
- Routes
- Web Application Firewall

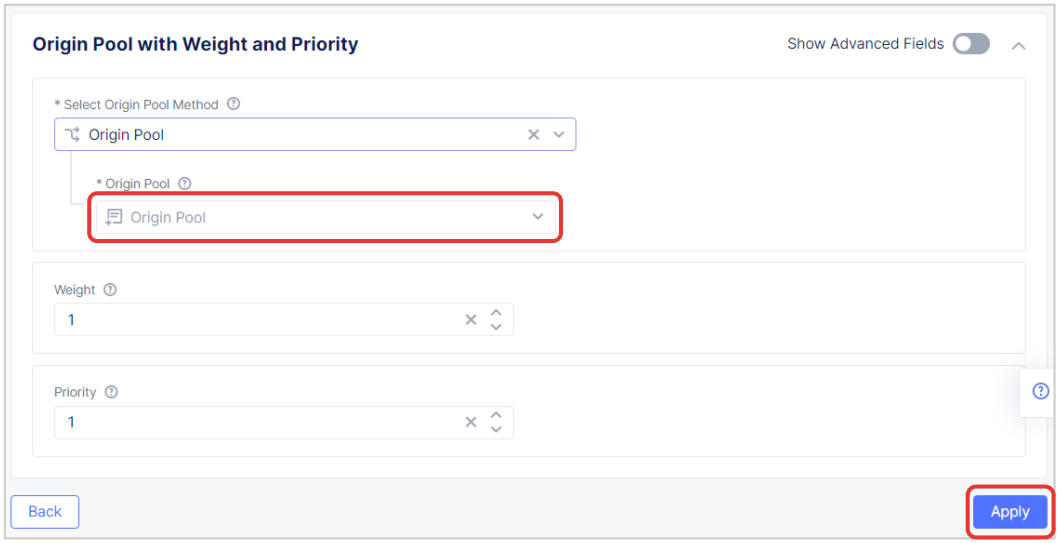
後述します

HTTP Load Balancer の作成③　ーPool の指定ー

HTTP Load Balancer 作成画面で、Origins から
Origin Pools の Add Item を押下します



Origin Pool のプルダウンから作成した Pool を選択し、
Apply を押下します



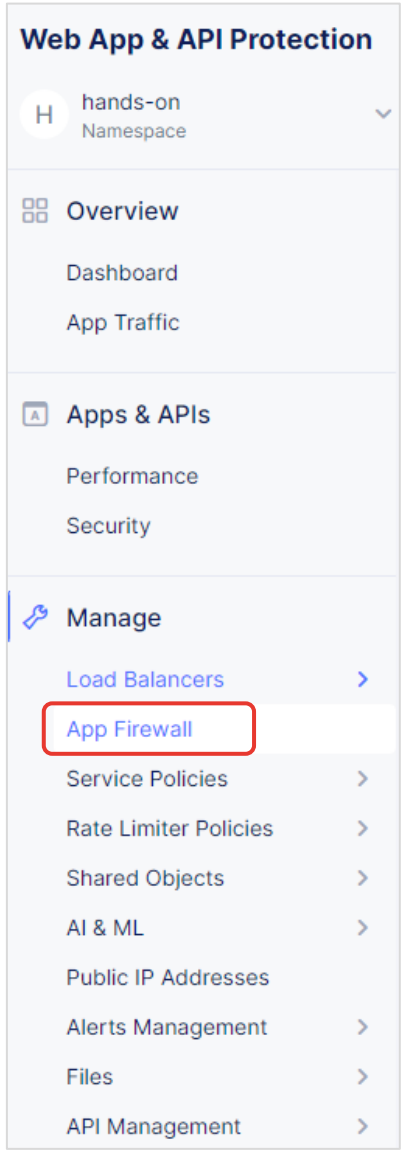
HTTP Load Balancer の設定画面に戻ったら
Add HTTP Load Balancer を押下して設定を保存してください

WAF Policy の作成/チューニング

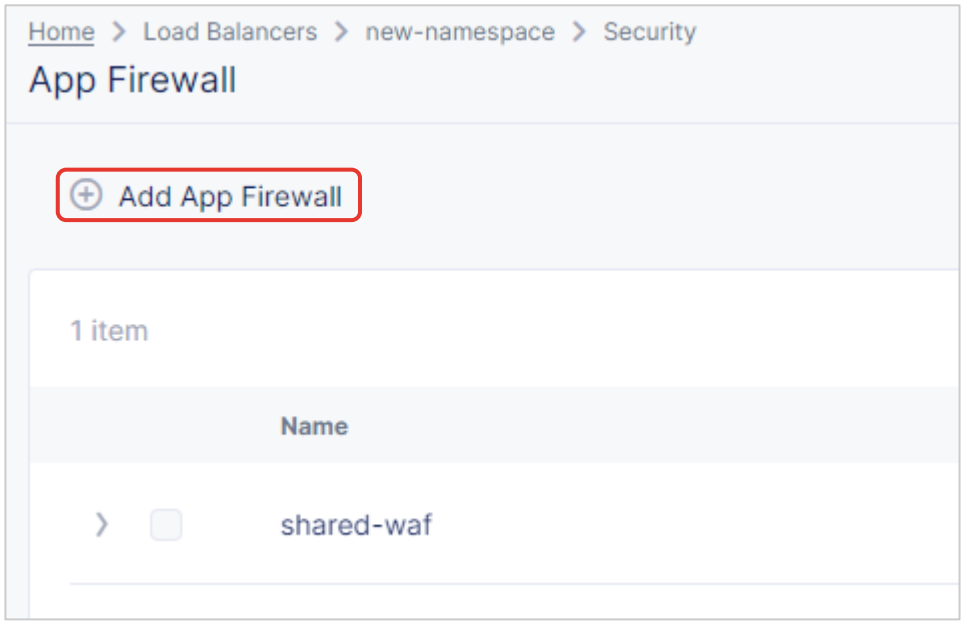
- 作成
- 適用
- 攻撃の検知
- チューニング

App Firewall の作成①　ーWAF 設定画面に入るー

左ペインから [App Firewall] を選択します

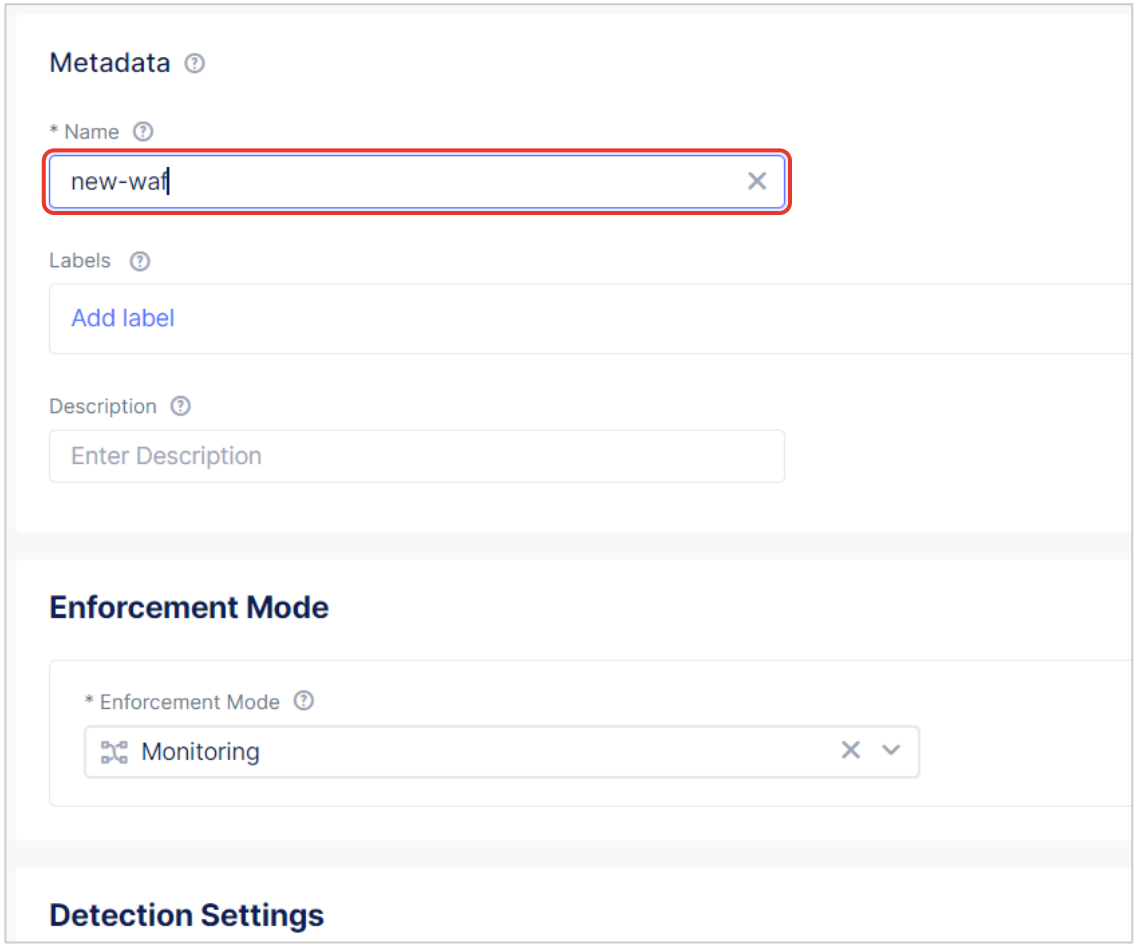


Add App Firewall をクリックします



App Firewall の作成②　ーパラメータを指定するー

※ここでは Name のみを指定して保存します

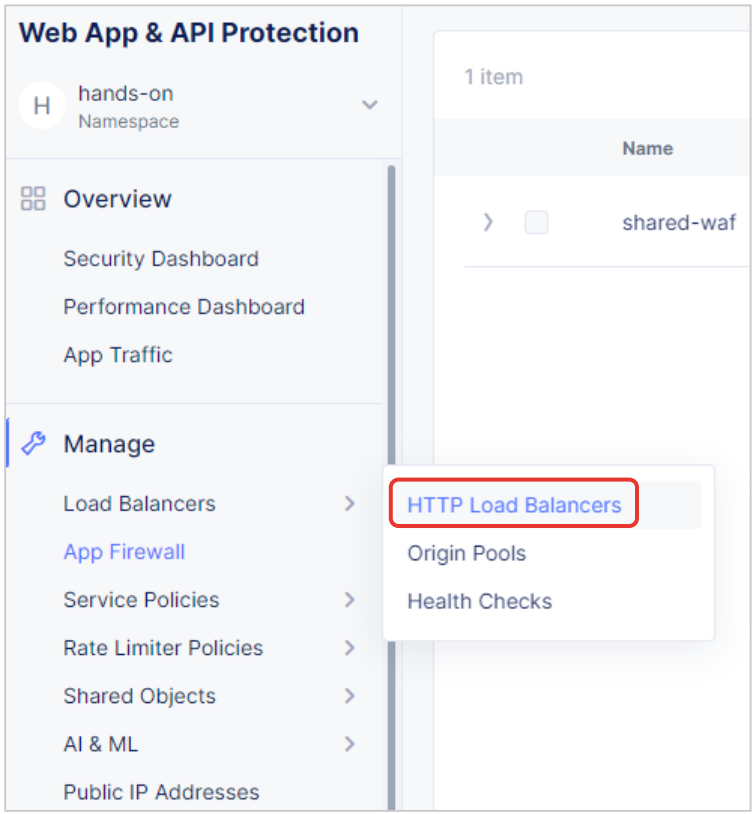


- Name
  - 任意（識別できる名称）

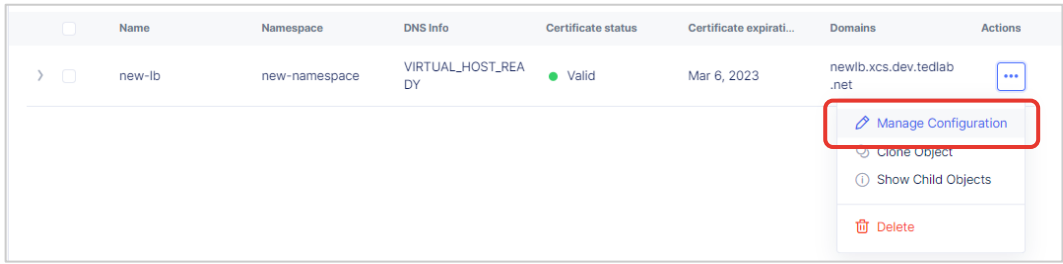
画面下部の Add App Firewall を押下して設定を保存します

App Firewall の適用①　－Load Balancer の編集画面に入る－

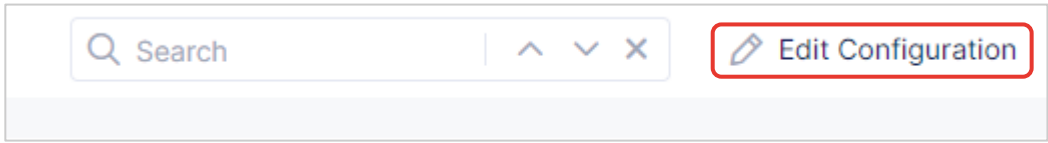
左ペインから
[Manage] – [Load Balancers] – [HTTP Load Balancers]
を選択します



作成した Load Balancer の Manage Configuration を選択します



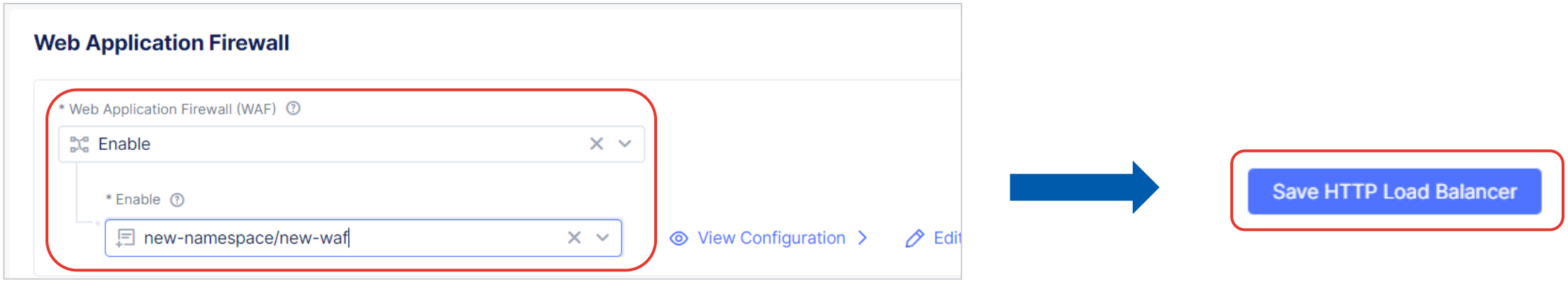
右上、Edit Configuration を選択します



App Firewall の適用②　ーLoad Balancer に WAF を適用するー

Web Application Firewall を Enable にして作成した App Firewall を選択します
プルダウンから作成した App Firewall を選択してください

選択したら Save HTTP Load Balancer を押下して設定を保存します



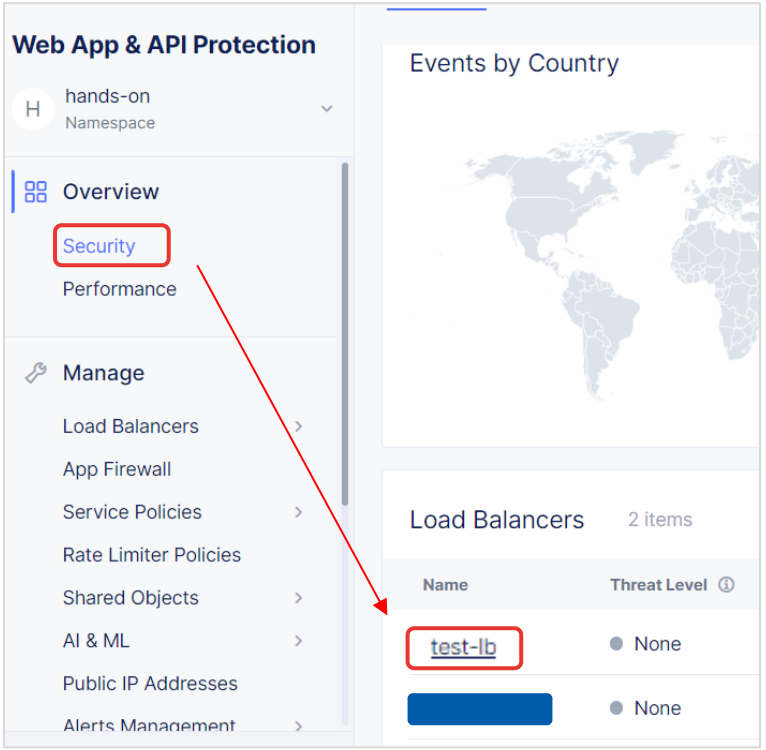
攻撃をしてみよう

右のURLにアクセスしてリクエストを送信してみましょう

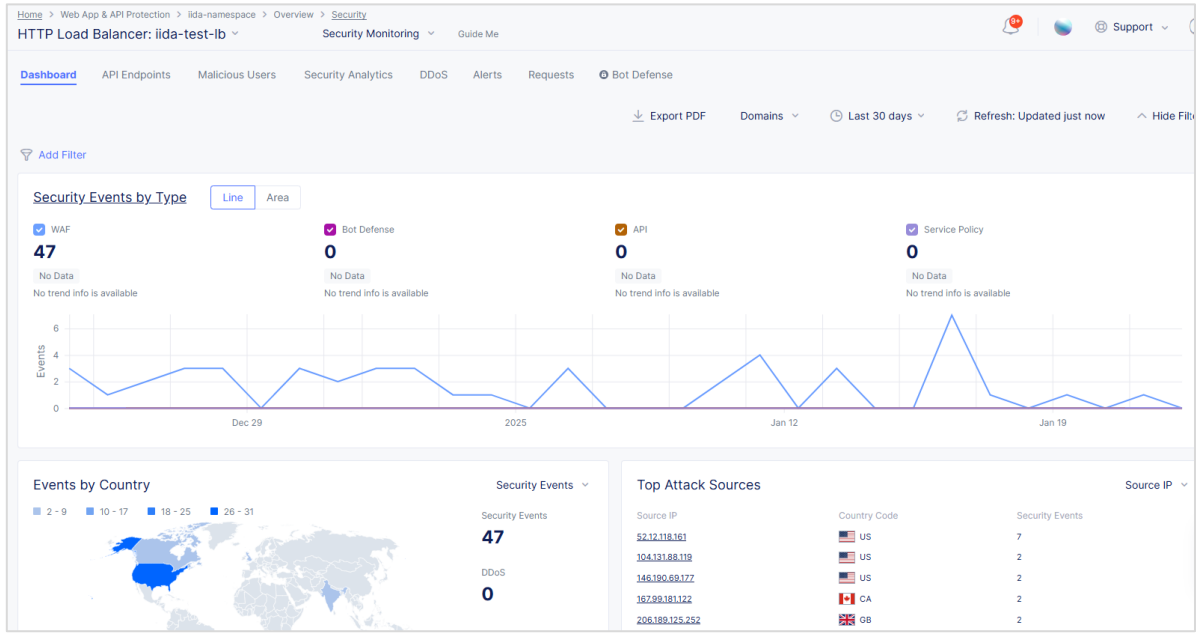
※ SQL インジェクションに該当します
※ <host> を変更して作成した Load Balancer にアクセスしてください

https://<host>.xc.dev.tedlab.net/vulnerabilities/sqli/?id=+id%3D%271%27+or+%271%27+%3D+%271%27

左ペイン [Overview] – [Security] を選択します
dashboard画面を下までスクロールし、作成した
HTTP Load Balancer を選択します

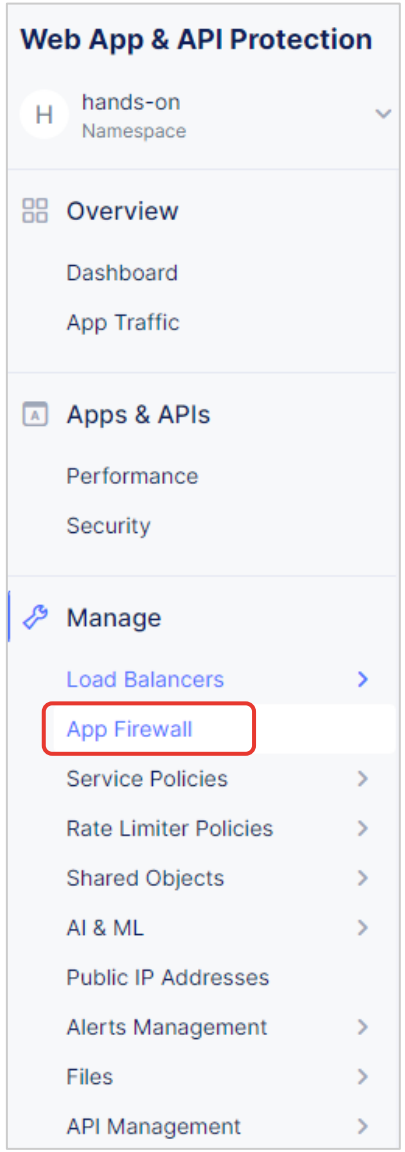


Dashboard を確認することで、
攻撃が検知されていることがわかります



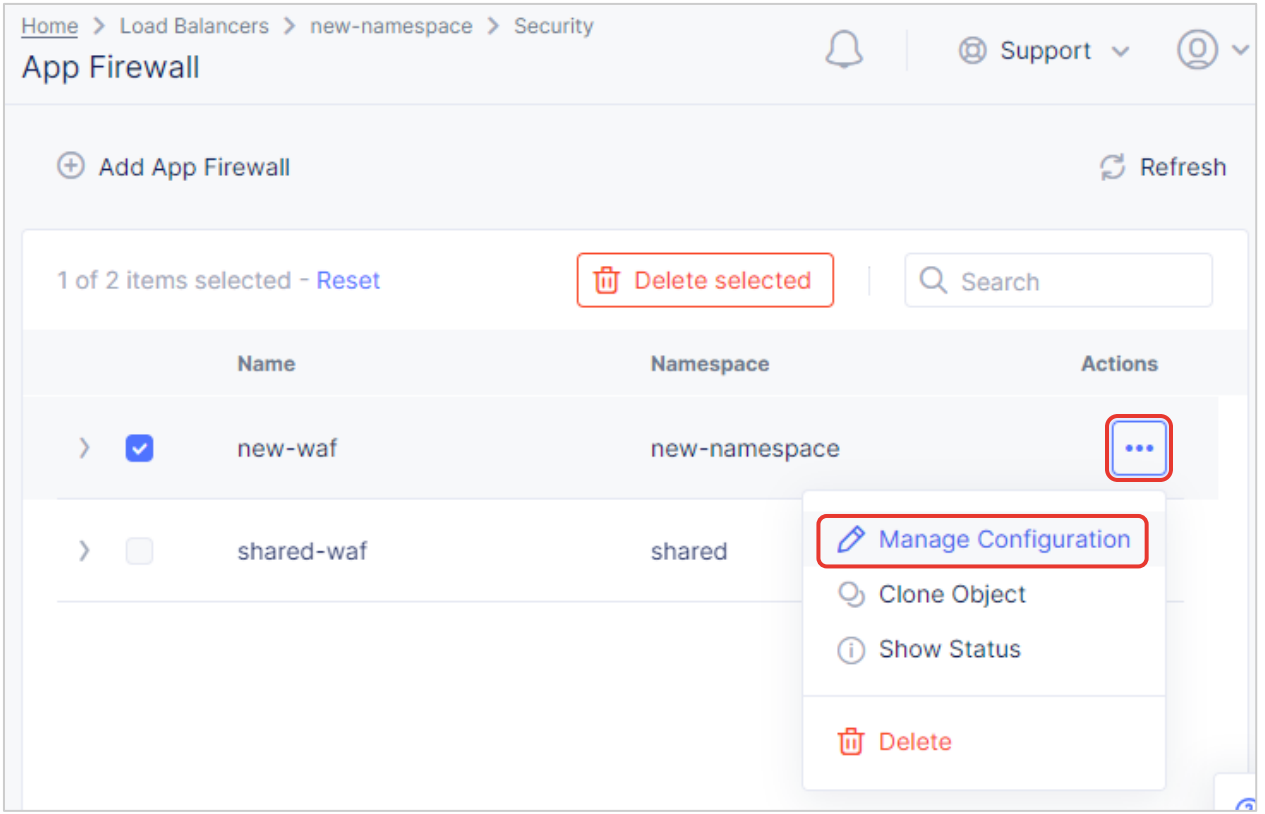
東京エレクトロン デバイス

App Firewall のカスタマイズ①



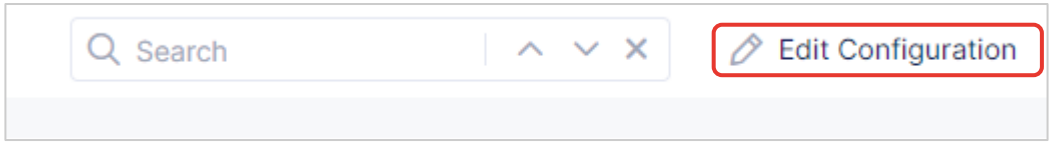
左ペインから [App Firewall] を選択します

作成した App Firewall の右端 ・・・ から
Manage Configuration を選択します

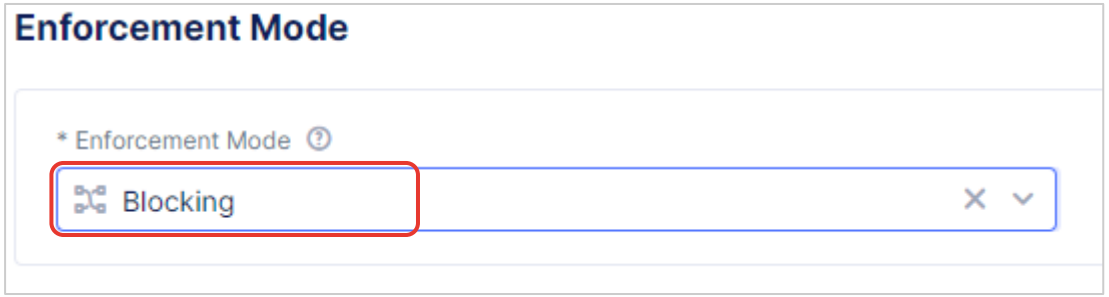


App Firewall のカスタマイズ②

右上、Edit Configuration を選択します

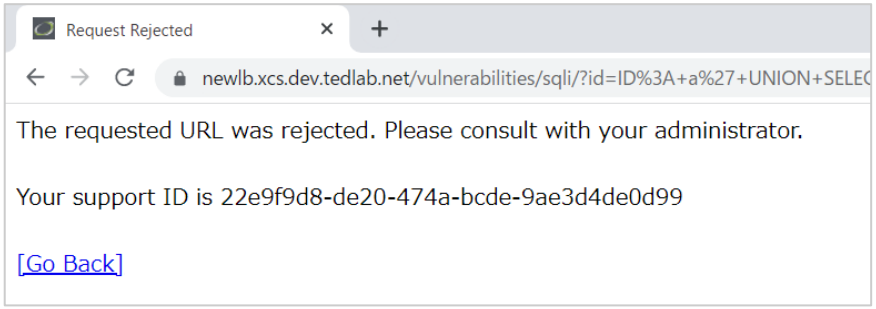


Enforcement Mode を Monitoring から Blocking に変更し、
Save App Firewall を押下して設定を保存します

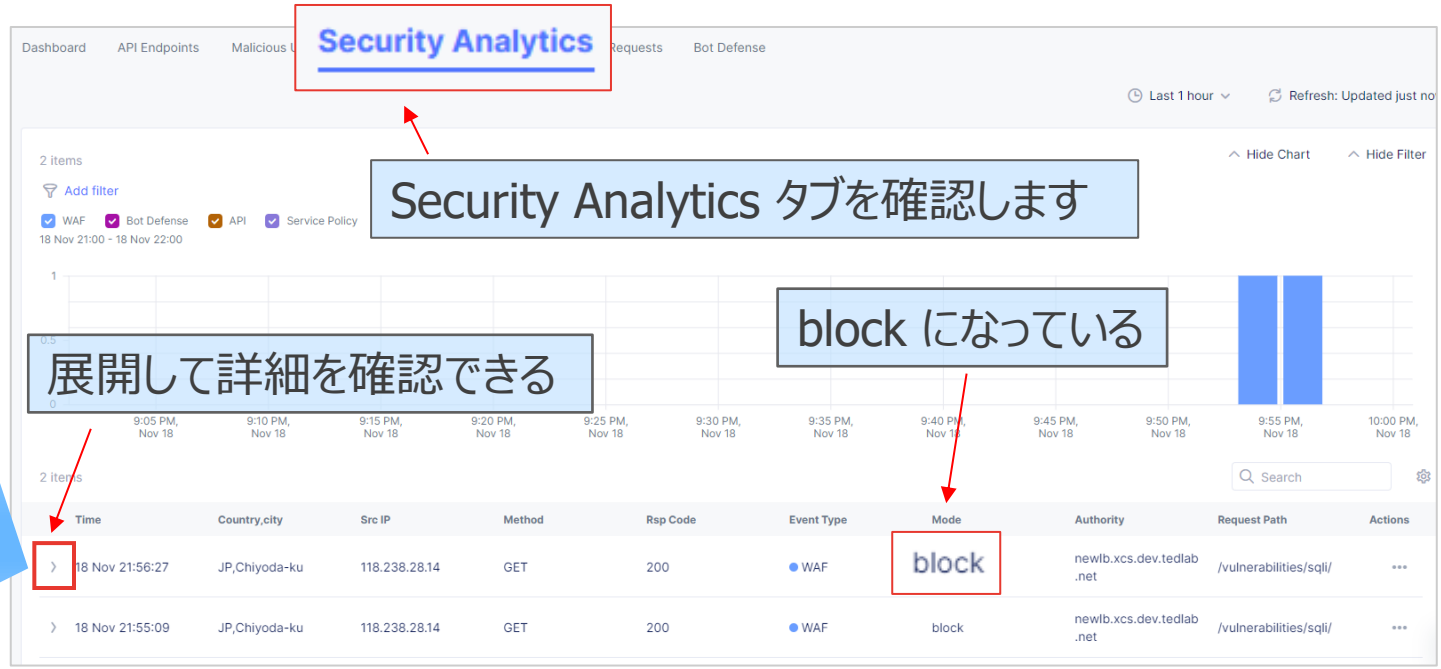


App Firewall のカスタマイズ③

攻撃に相当するアクセスを行ってみると
攻撃はブロックされる



詳細を確認することで攻撃の概要、
Signature IDなどを確認できる



App Firewall のカスタマイズ④　ーアラート表示期間の指定ー

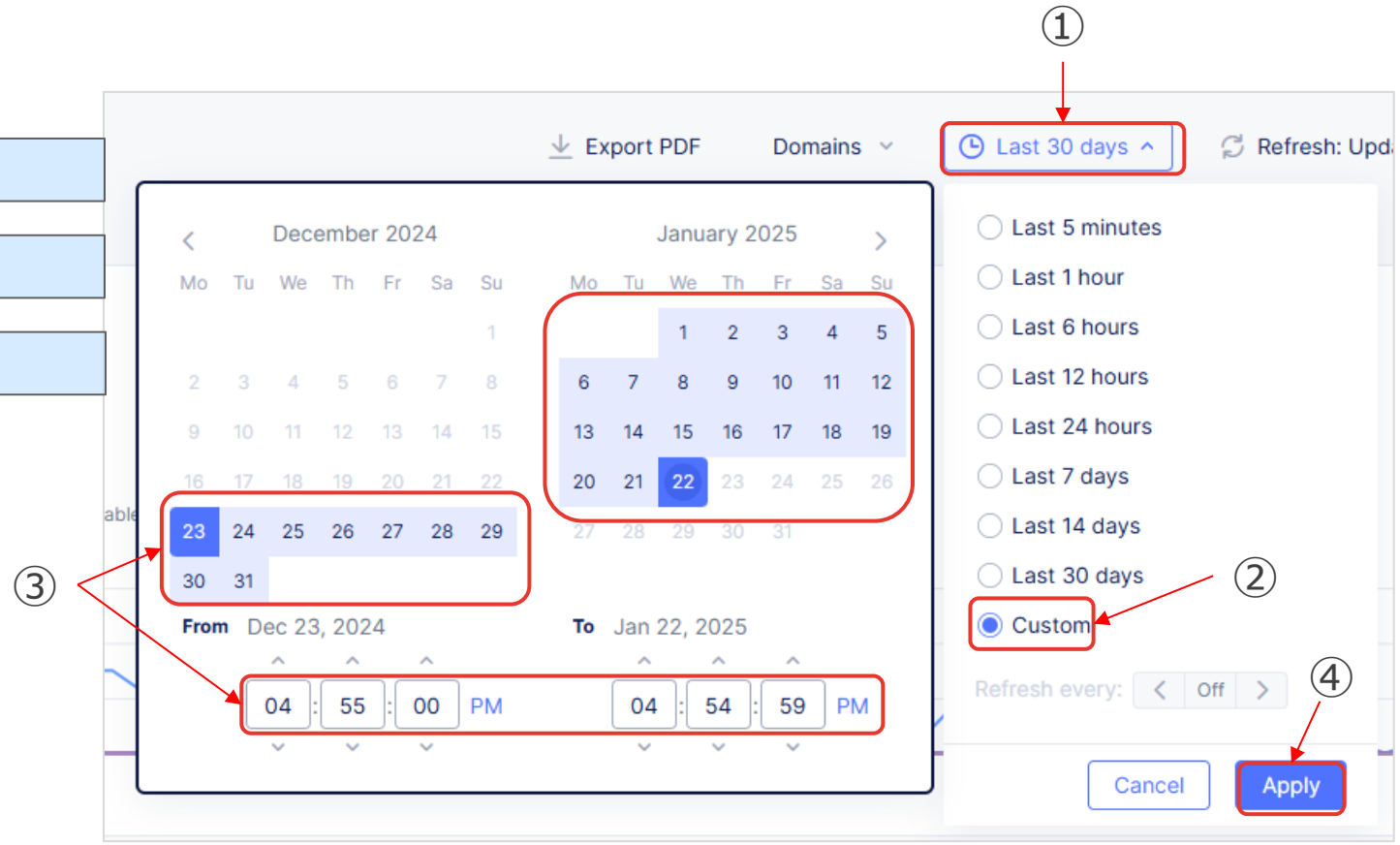
Dashboard ではアラートの表示期間を変更することができます

① Dashboard 右上に位置する表示期間をクリックします

② Custom を選択します（デフォルト：Last 5 minute）

③ 日付と時刻を選択して期間を指定します（最大1ヶ月）

④ Apply を押下して反映します

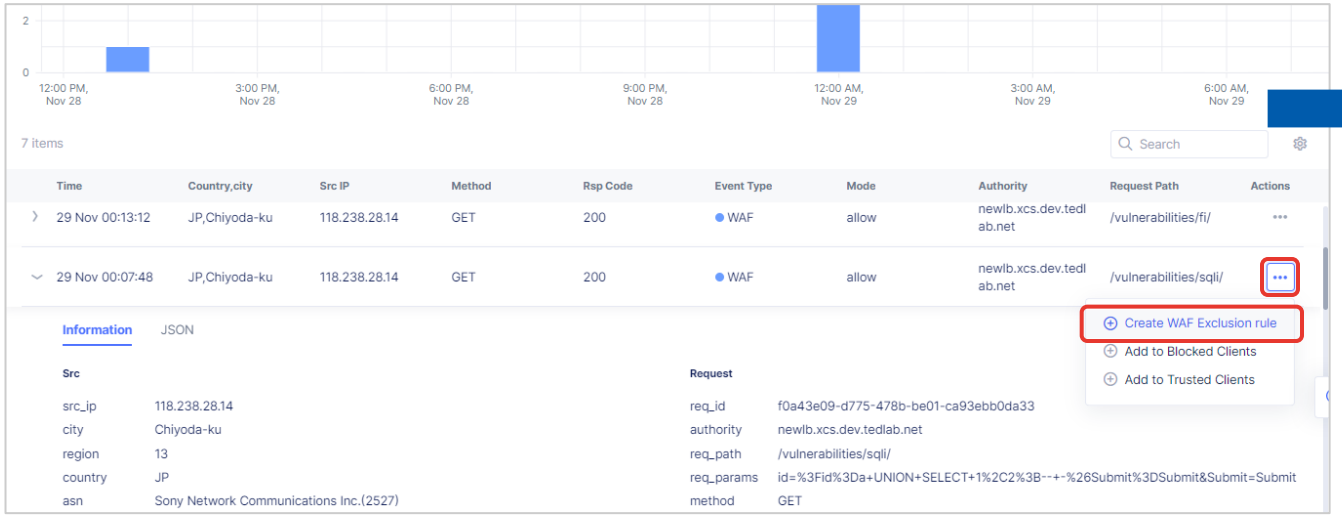


東京エレクトロン デバイス

App Firewall のカスタマイズ⑤　ーWAF 除外ルールの作成ー

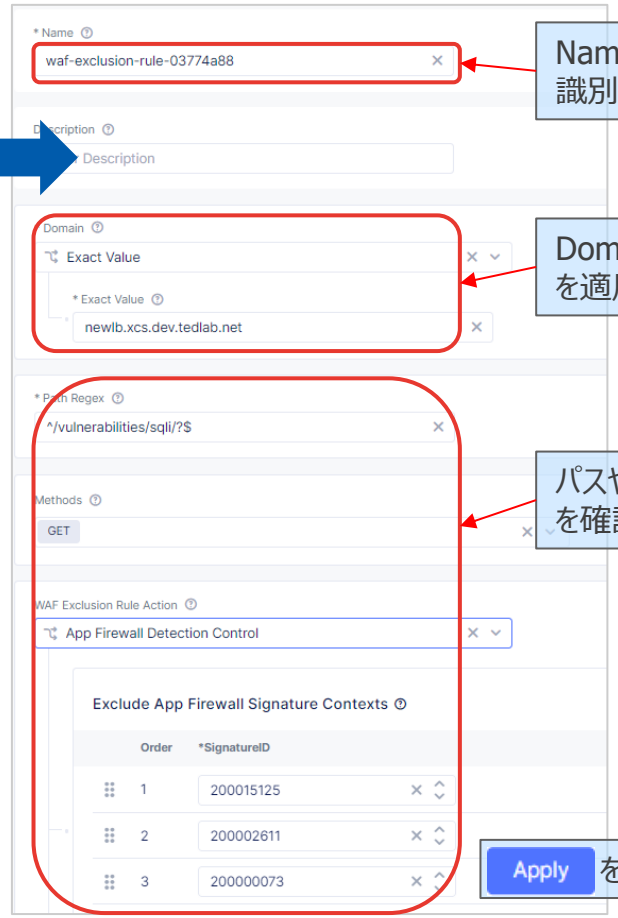
表示されているアラートから WAF Exclusion Rule を作成することができます。

除外したいアラートの右端 ・・・ から
Create WAF Exclusion rule を選択します



数回の画面遷移が行われ、WAF Exclusion Rule の設定画面に自動で移行します

画面の移行が完了すると以下のようなページが表示されます



Name は自動で入力されているため識別しやすい名称に変更します

Domain 毎に WAF Exclusion Rule を適用できます

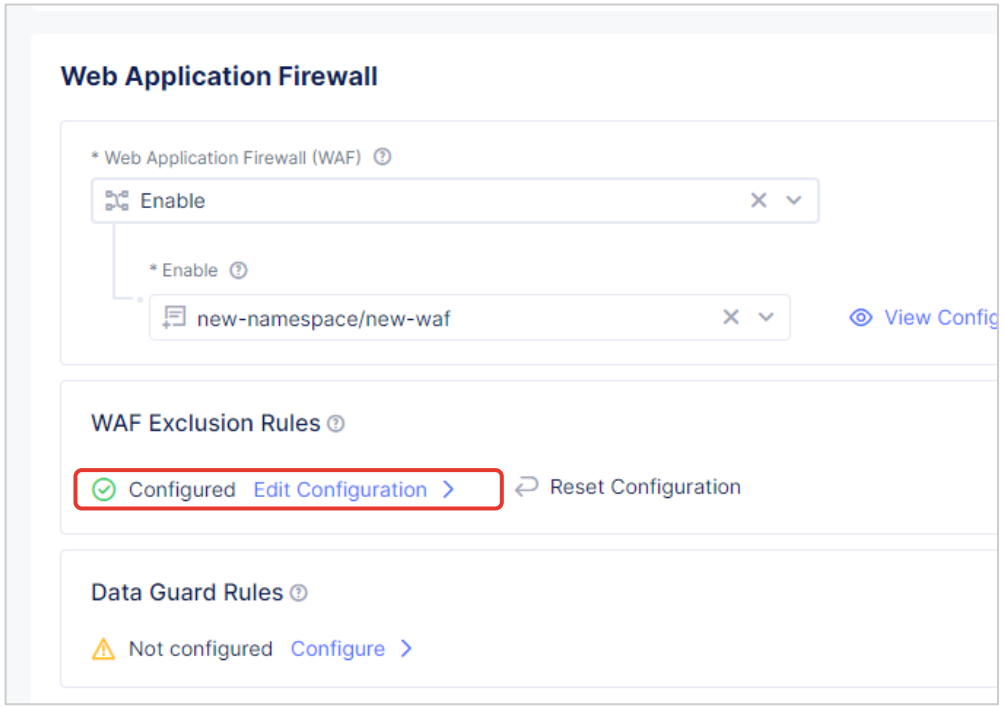
パスやリクエストメソッド、Signature ID を確認して必要に応じて編集できます

Apply を押下してルールを作成します

App Firewall のカスタマイズ⑥　ーLoad Balancer への Rule 適用ー

複数回 Apply を押下して、Load Balancer の
管理画面に移動します

Edit Configuration > を選択することで適用した
WAF Exclusion Rule を確認できます



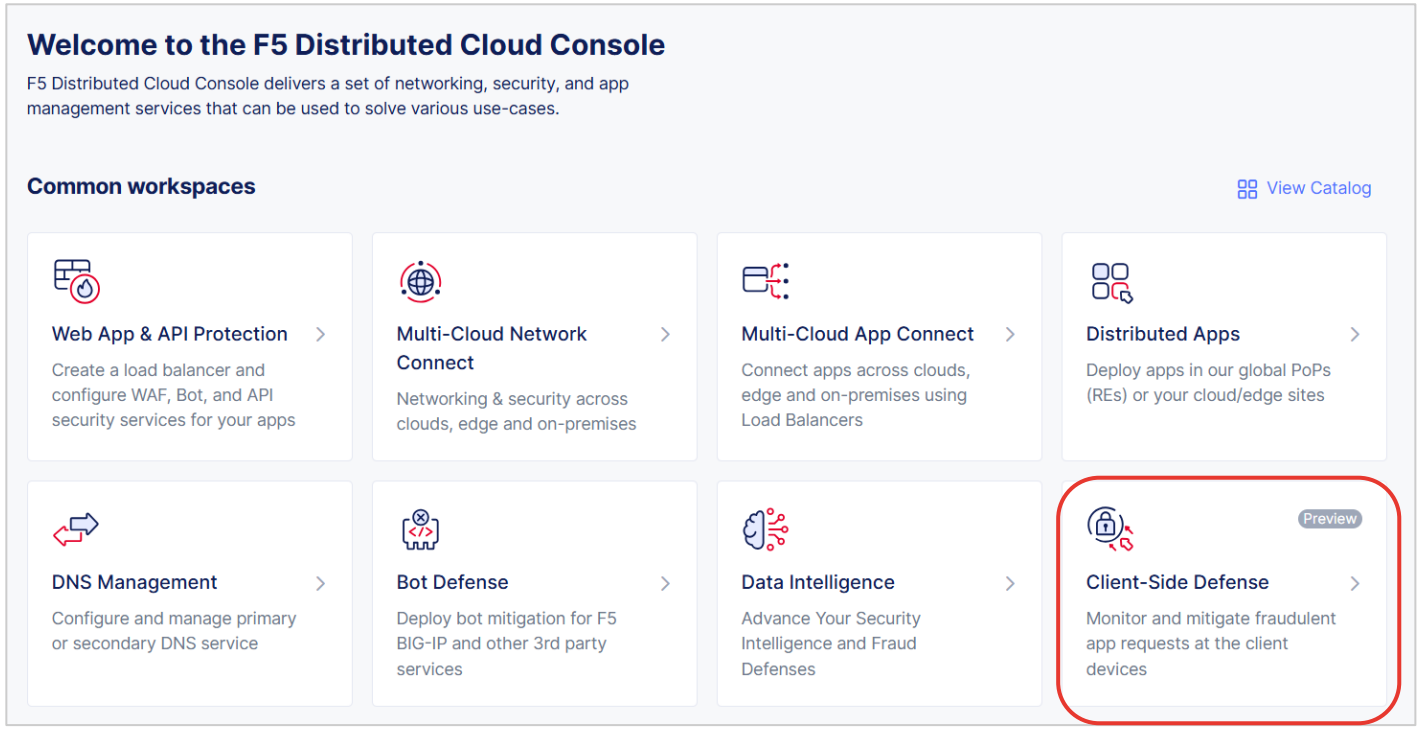
Save HTTP Load Balancer を押下して Load Balancer の設定として
保存します

デモ

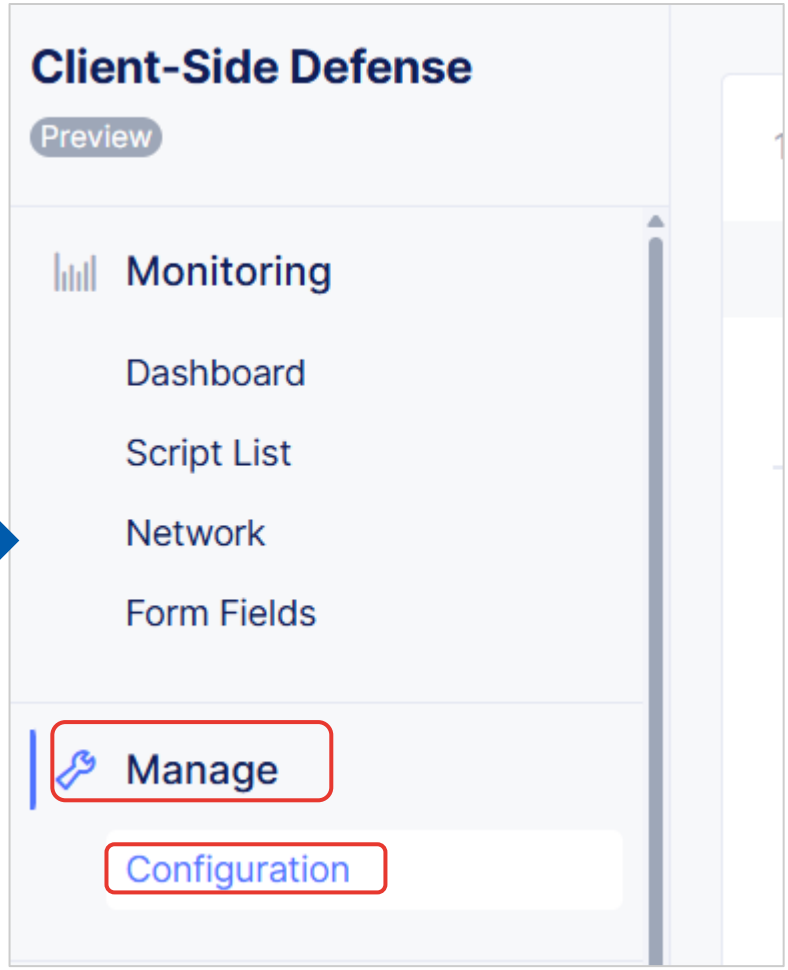
- Webスキミング対策（Client-Side Defense）の設定

Client-Side Defenseの設定方法①

Home メニューから Client-Side Defenseを選択します

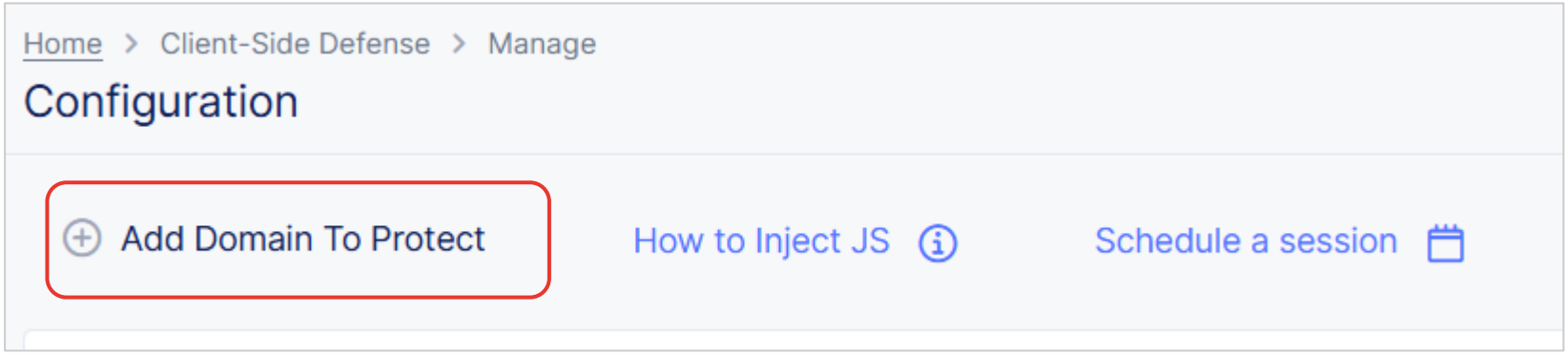


左ペインから
[Manage] – [Configuration] を選択します

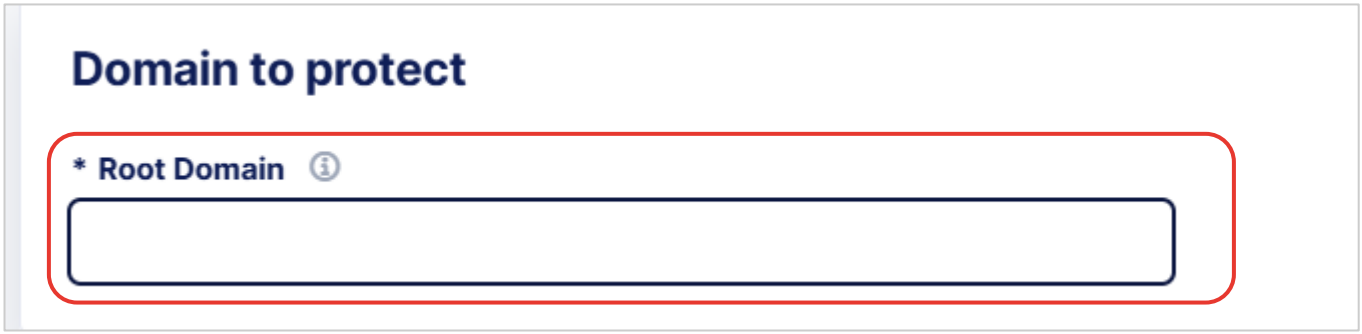


Client-Side Defenseの設定方法②

Add Domain To Protect をクリックします



ドメイン名を入力し、Save and Exitをクリックします
※ドメイン名はトップレベルドメインの1つ上の階層のドメインにする必要があります
例）example.com、example.co.jpなど

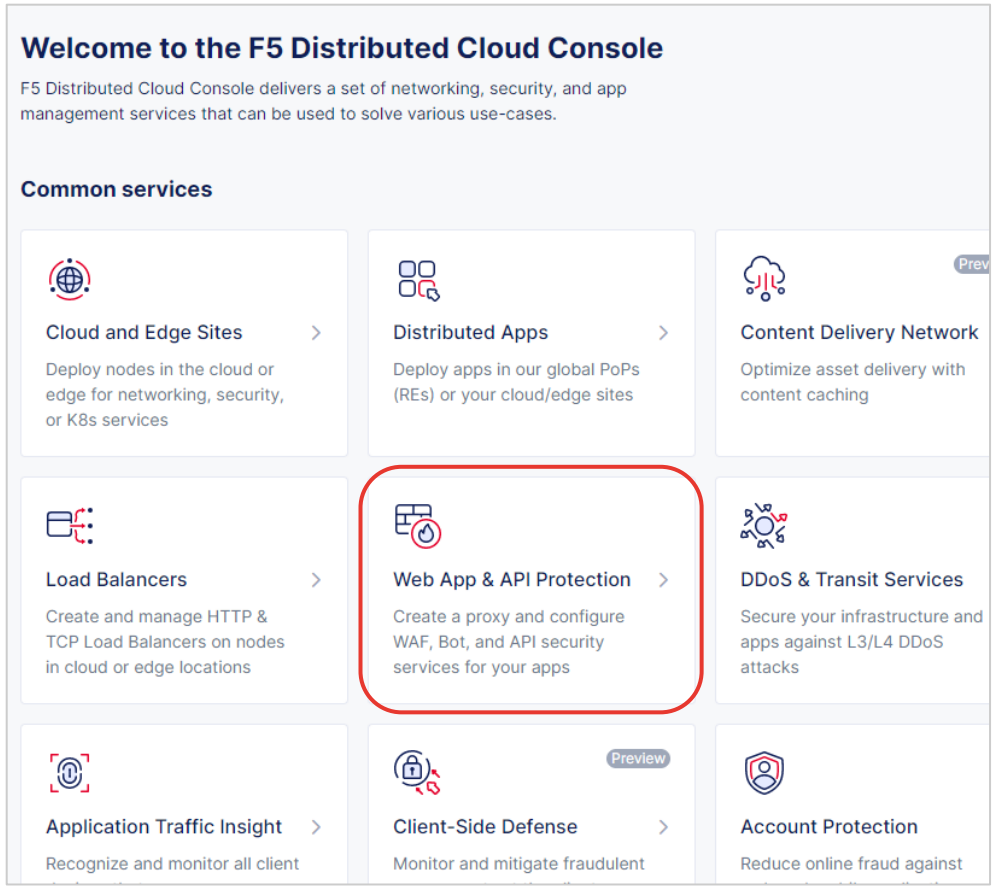


ドメイン名を入力後、 Add Domain to protect をクリック

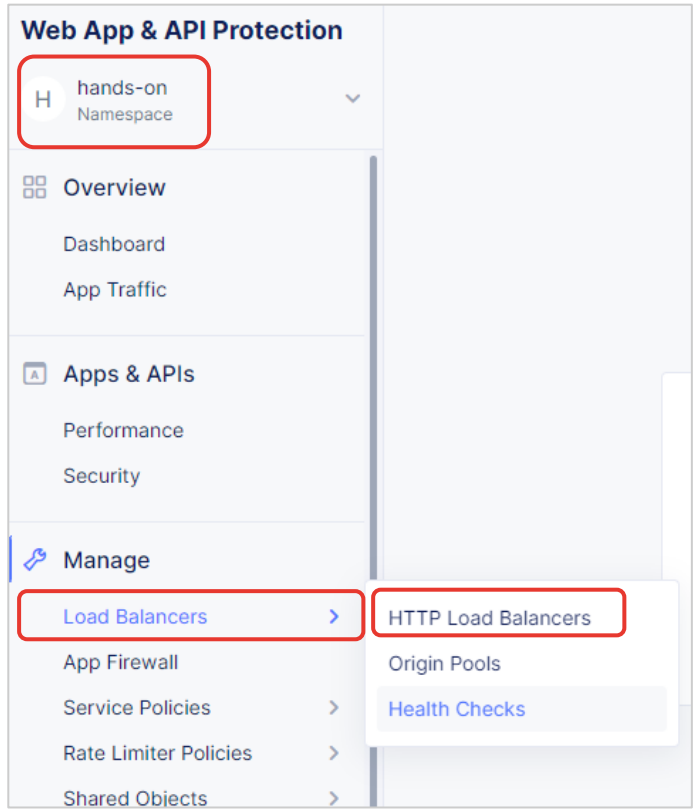
Client-Side Defenseの設定方法③

HTTP Load BalancerでClient-Side Defenseを有効化します

Home メニューから Web App & API Protection を選択します

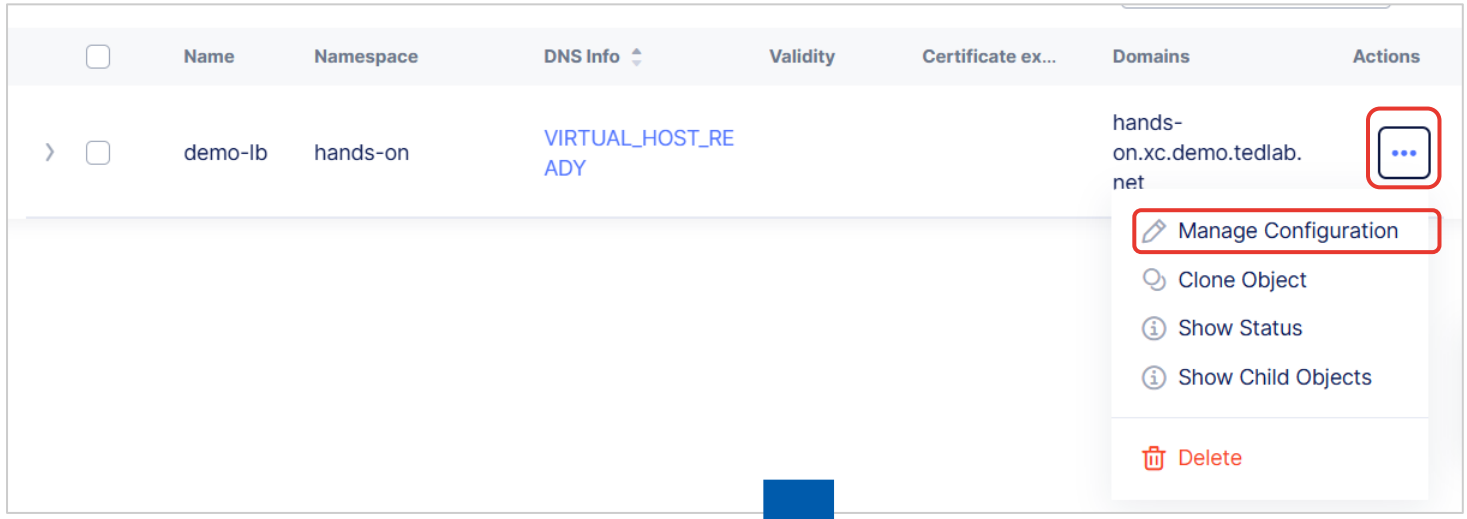


Namespaceが hands-on になっていることを確認し、左ペインから
[Manage] – [Load Balancers] –
[HTTP Load Balancers] を選択します

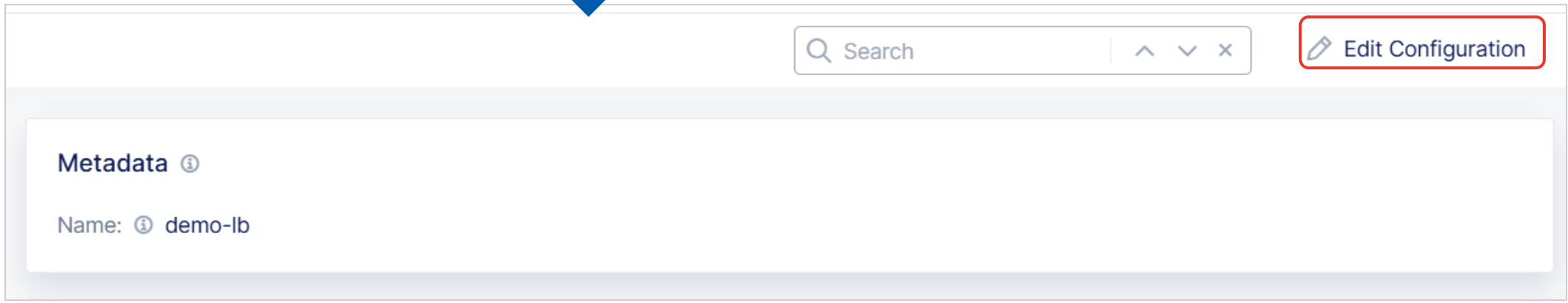


Client-Side Defenseの設定方法④

作成済みのHTTP Load BalancerのActions 列にある三点リーダー（・・・）をクリックし、
Manage Configurationをクリックします

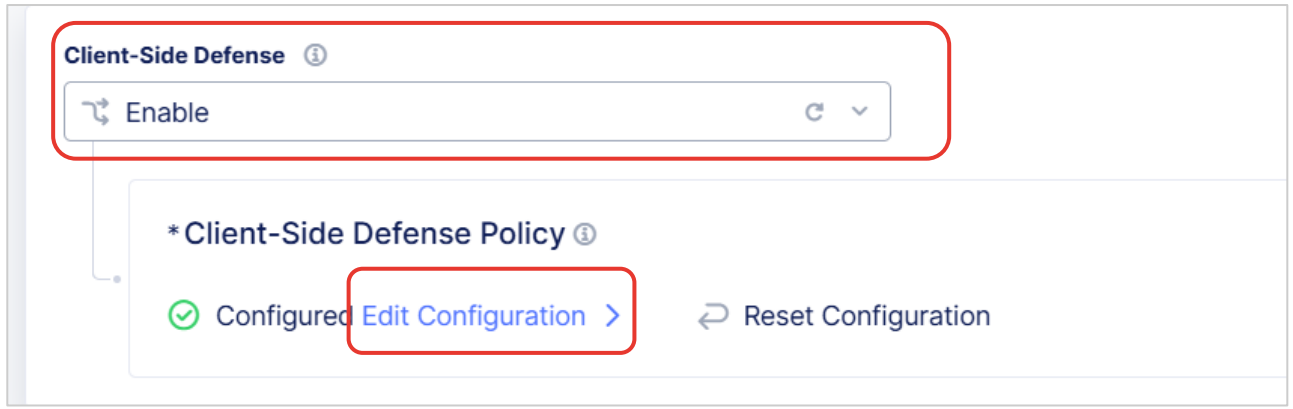


HTTP Load Balancerの設定画面に遷移後、画面右上にある
Edit Configurationをクリックします

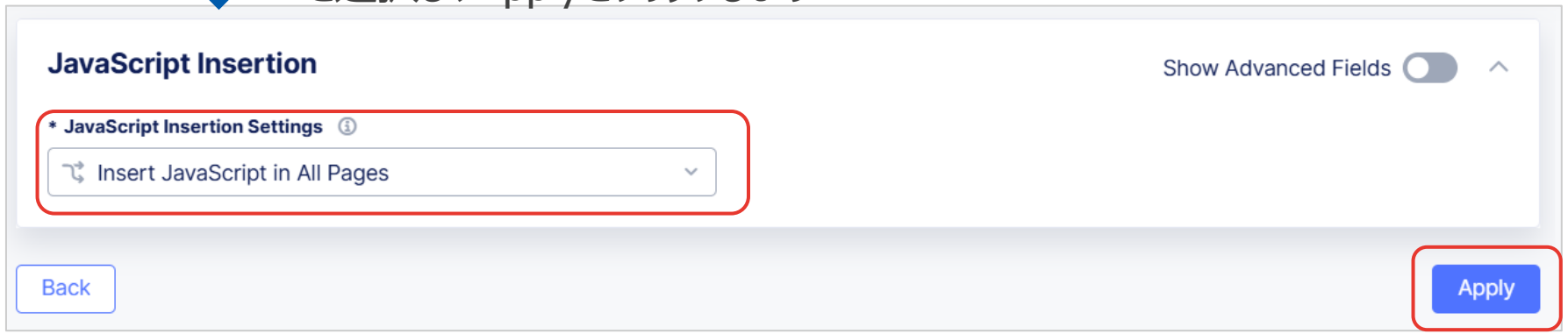


Client-Side Defenseの設定方法⑤

Client-Side Defenseのプルダウンから Enableを選択し、Edit Configurationをクリックします



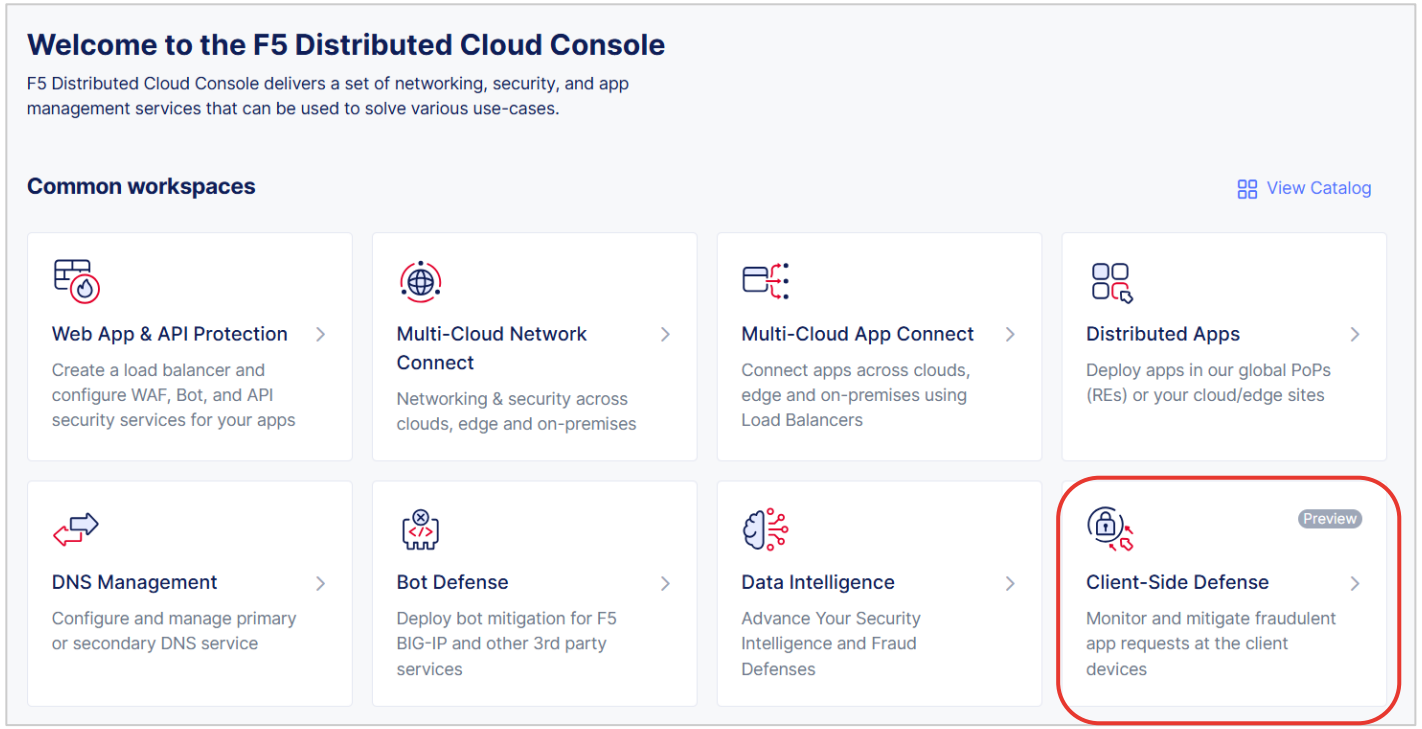
JavaScript Insertion Settingsのプルダウンから Insert JavaScript in All Pages を選択し、Applyをクリックします



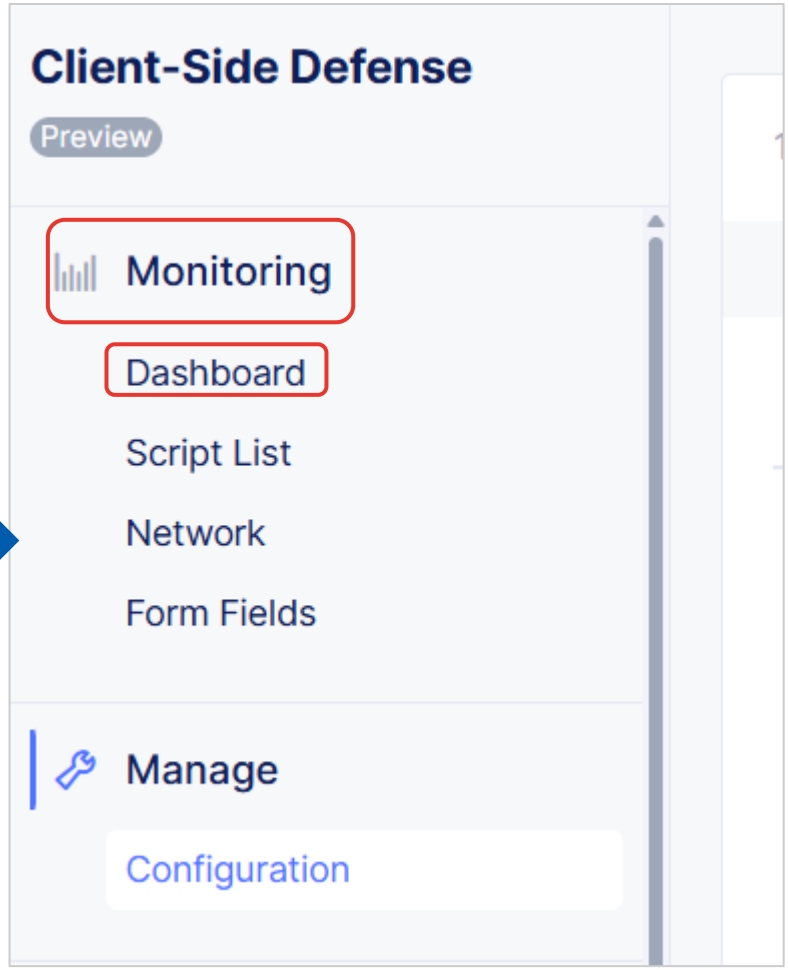
Applyをクリック後、画面最下部の Save HTTP Load Balancer をクリック

Client-Side Defenseのコンソール

Home メニューから Client-Side Defenseを選択します



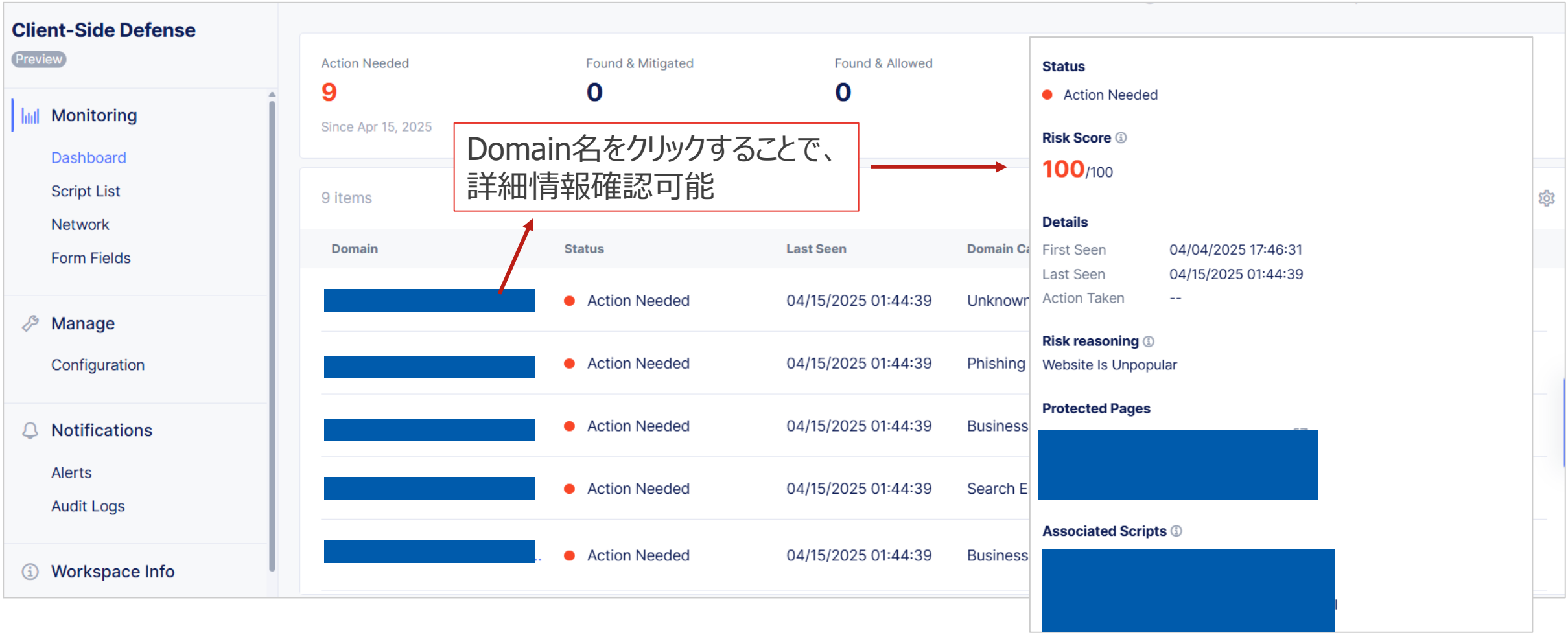
左ペインから
[Monitoring] – [Dashboard] を選択します



Client-Side Defenseのコンソール

Client-side Defense > Monitoring > Dashboard

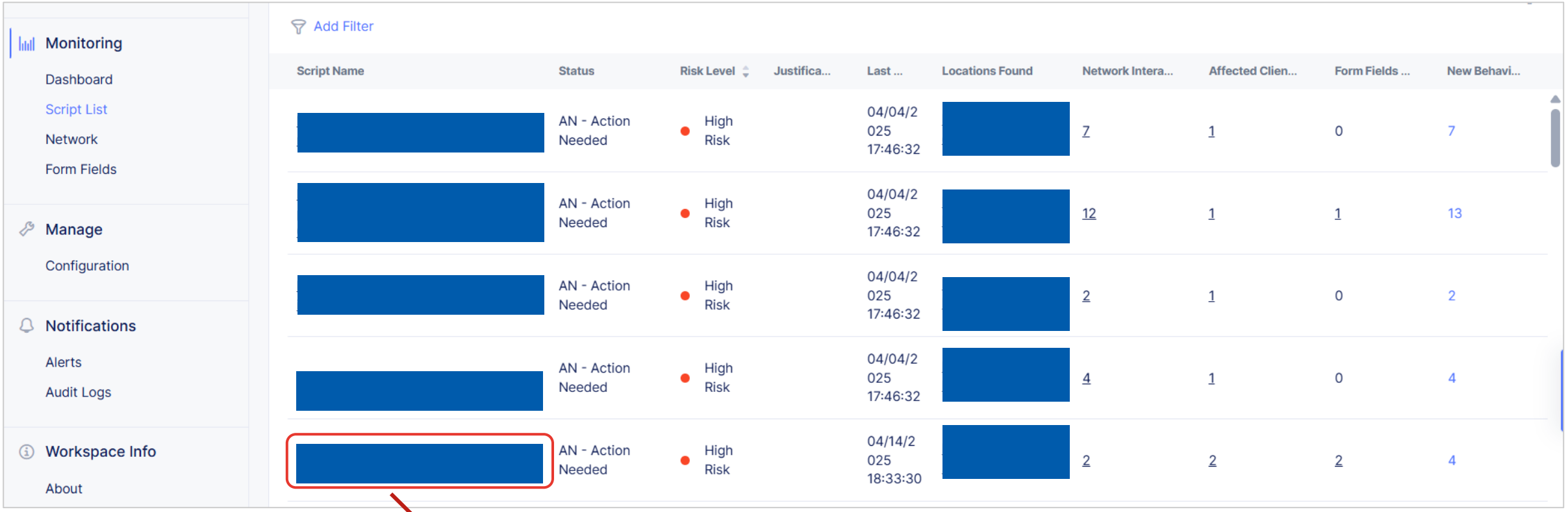
Dashboardでは、該当スクリプトがどのドメイン宛に通信されたかを一覧を確認できます



Client-Side Defenseのコンソール

Client-side Defense > Monitoring > Script List

Script Listでは、検知した各スクリプトのリスクの高さや検知した回数など一覧で確認できます

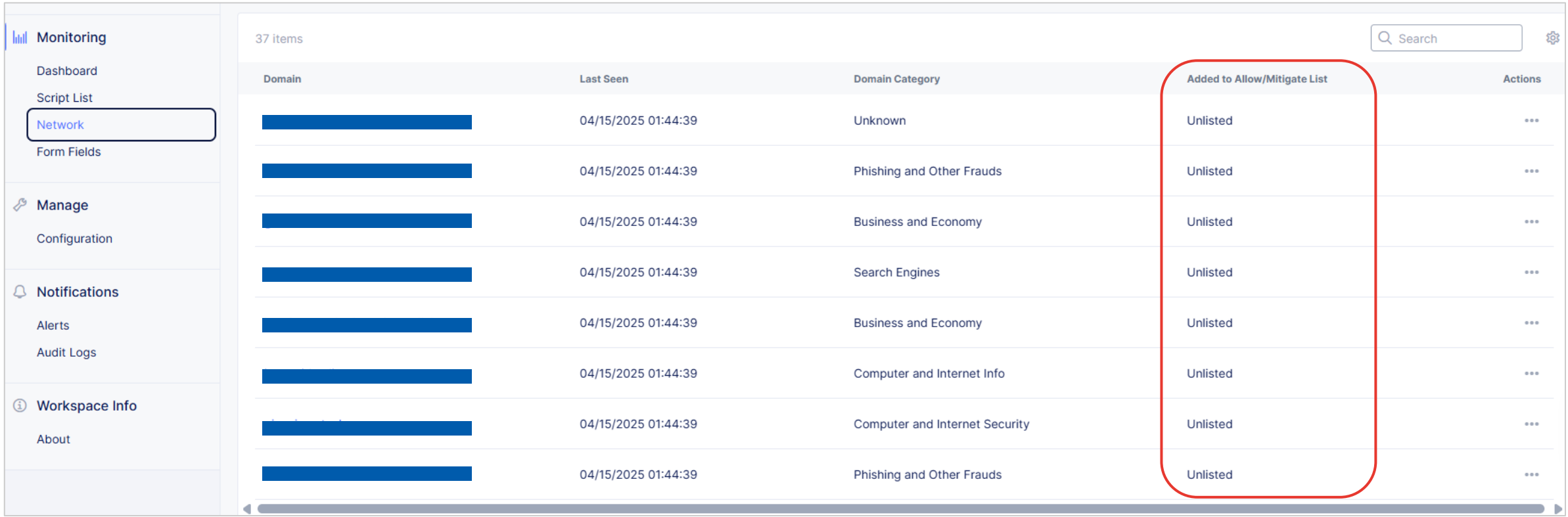


Script名をクリックすることで、検知した時間や接続先のドメイン名などを確認することが可能

Client-Side Defenseのコンソール

Client-side Defense > Monitoring > Script List > Network List

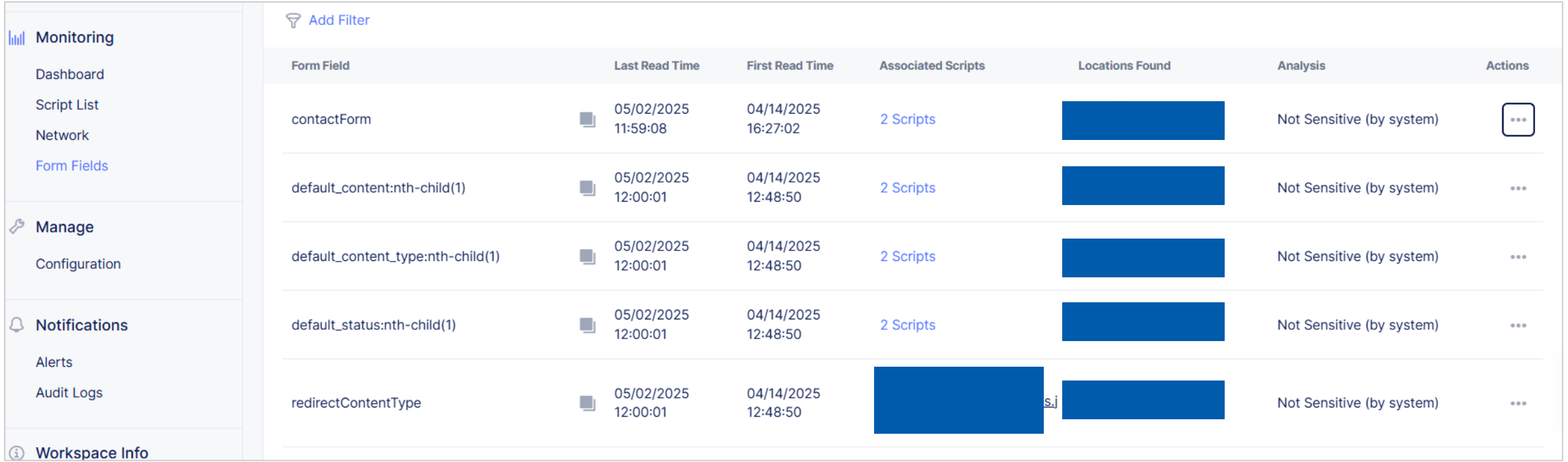
Network Listでは、該当スクリプトがどのドメイン宛に通信されたか、
またそのドメインは許可/拒否されているかを一覧を確認できます



Client-Side Defenseのコンソール

Client-side Defense > Monitoring > Script List > Form Field

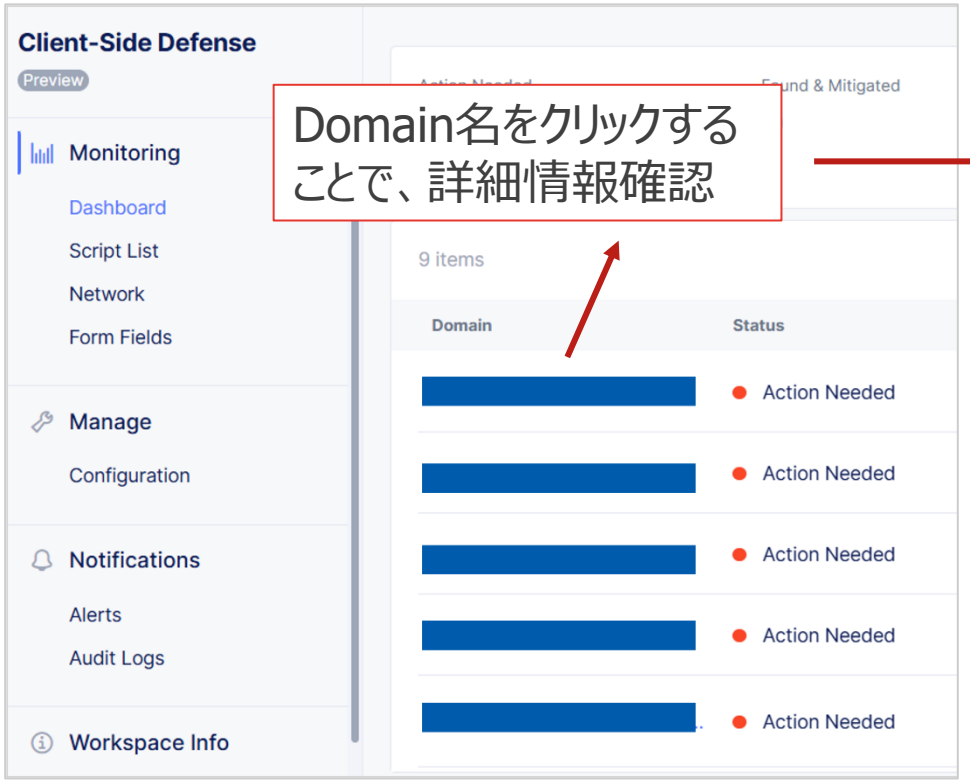
Form Fieldのどの部分が読み取りされているかを一覧で確認できます



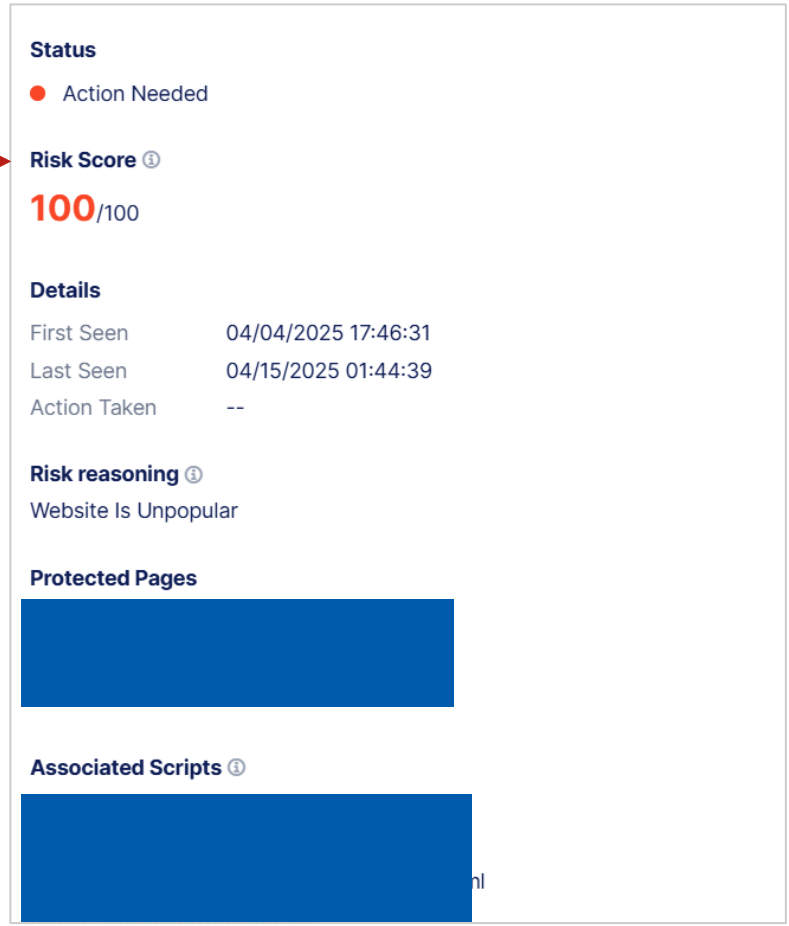
Scriptの通信先をAllow ListまたはMitigate Listに追加

Client-side Defense > Monitoring > Dashboard
Client-side Defense > Monitoring > Script List > Network List

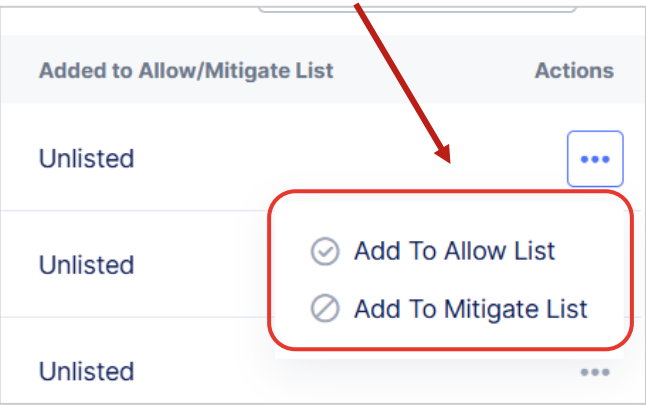
Dashboardまたは、Network Listの画面から、該当ドメインへの接続を許可/拒否の設定をすることが可能



Domain名をクリックすることで、詳細情報確認



各ドメインへの接続が想定されるものならば、Allow Listへ追加
想定されないものであれば、Mitigate Listへ追加



おまけ

どんな環境でも攻撃はきています

以下は、弊社の検証環境に対しての攻撃を可視化したものになります。
（主にWAFのシグネチャに該当した攻撃を可視化）

30日間の攻撃を可視化

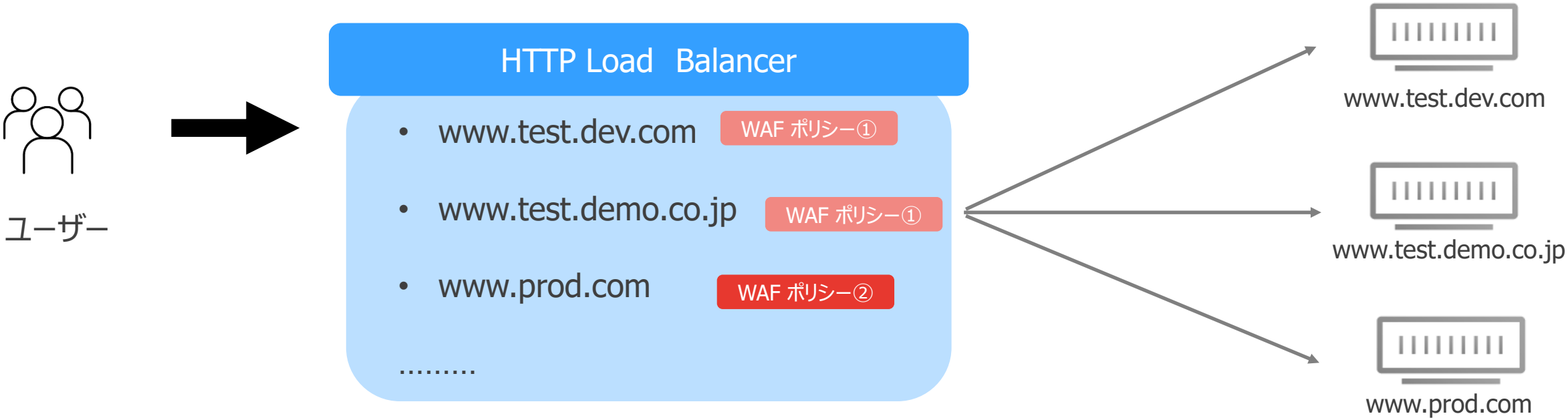
WAFの必要性についてのブログもありますので、こちらも合わせてご参照ください
https://cn.teldevice.co.jp/blog/p60360/

弊社環境には、顧客情報のような機密情報も無ければ、コンテンツを配信するだけのシンプルなWebサーバーしかない状態
それでも、定期的にいろいろな国から攻撃が来ている状況
攻撃者としては、何かしら悪用できるサーバー等がないかを常に探していると予想できる

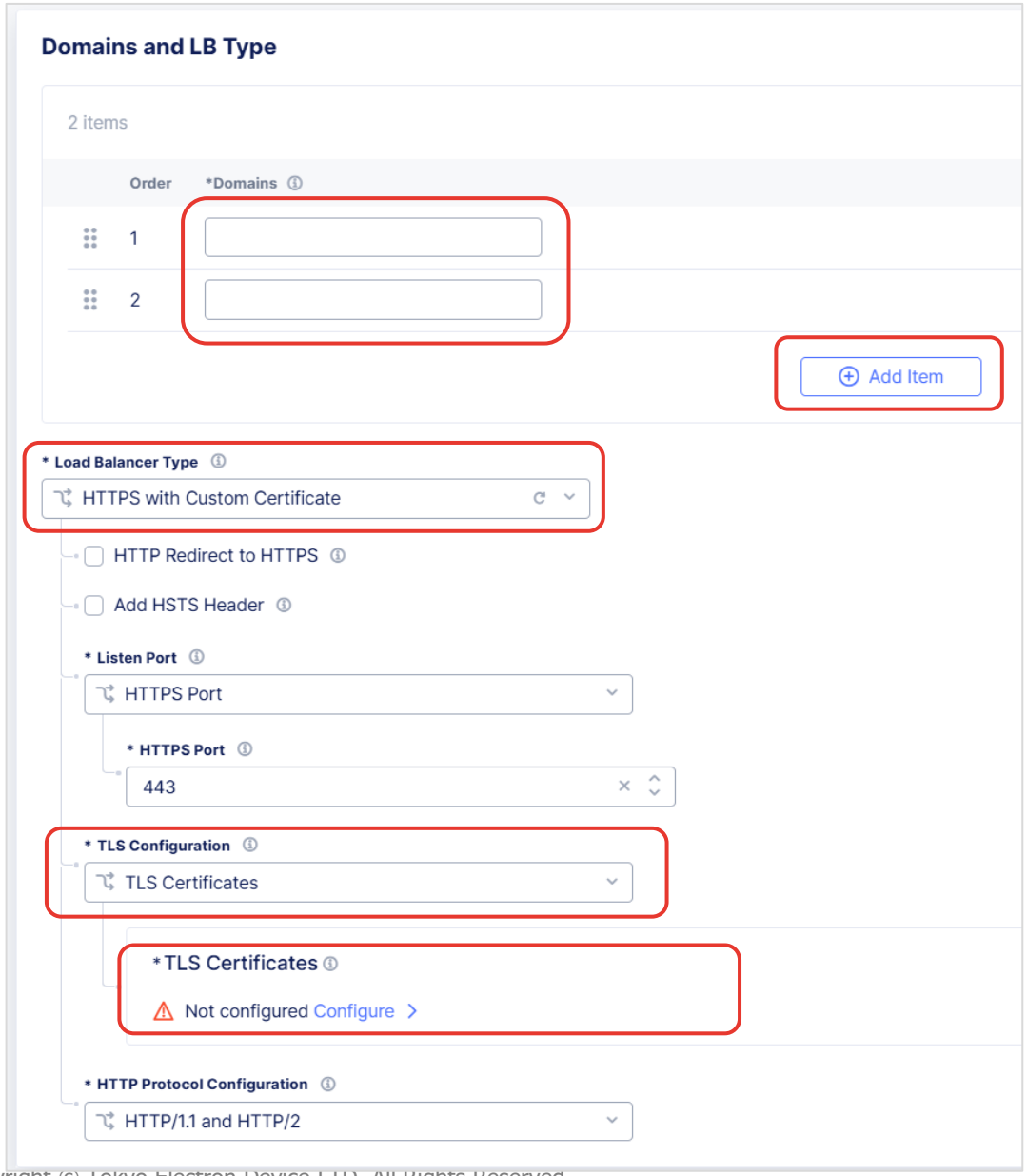
補足

1つのHTTP Load Balancer に複数 FQDN を登録して利用

- 1つの HTTP LB に対して、複数の FQDN を集約することが可能です
  - 最大で32　FQDNまで集約することができます
- FQDN 単位で WAF のポリシーやバランシング先などを制御することも可能です



複数 FQDN を登録及び、証明書の登録



- Domains

  Add Itemを押下して複数FQDN を入力します

- Load Balancer Type

  HTTPS の場合、証明書を自動生成するか手動インポートかを選択できます

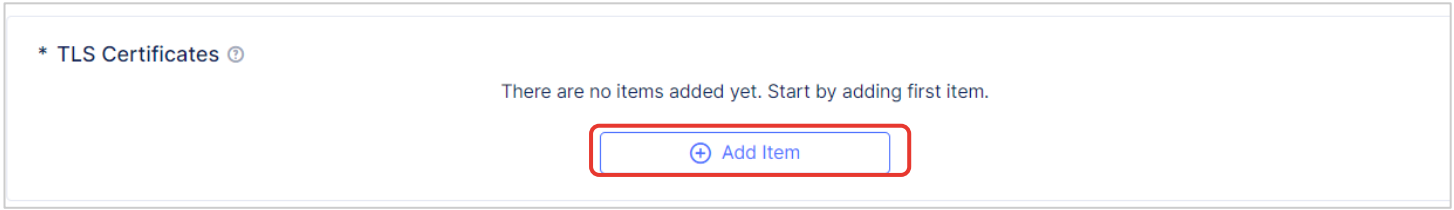
  - HTTPS with Automatic Certificate
    （自動生成:XC の DNS に権限移譲している場合のみ）
  - HTTPS with Custom Certificate
    （お客様持ち込みの証明書を使用する場合）

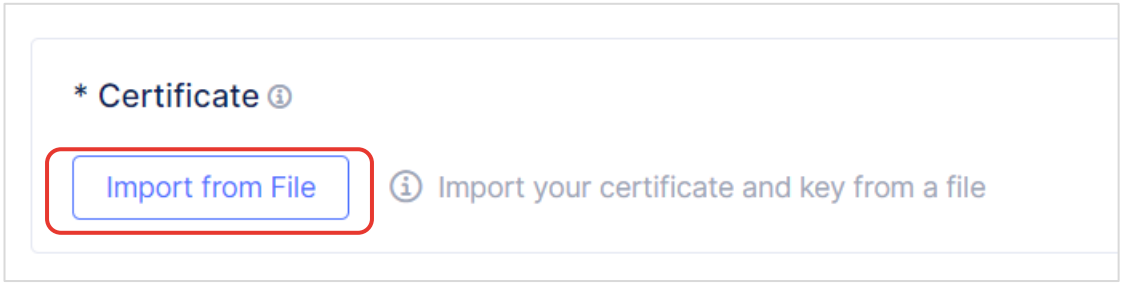
- TLS Configuration

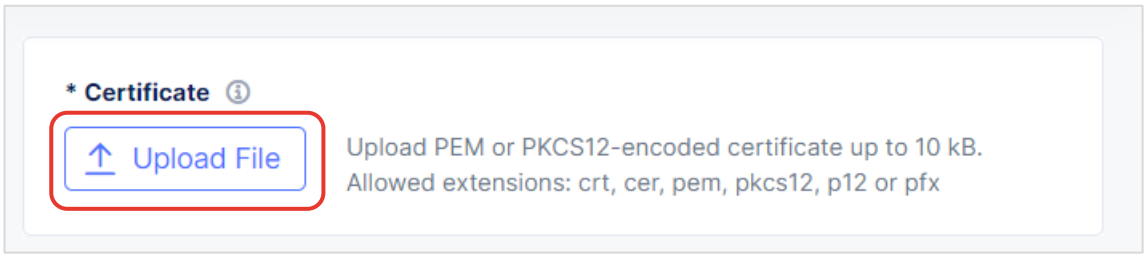
  - Configure を押下して、証明書の設定をします
    （HTTPS with Custom Certificate を選択した場合）

証明書の登録①

Add Item 押下して証明書の登録をします





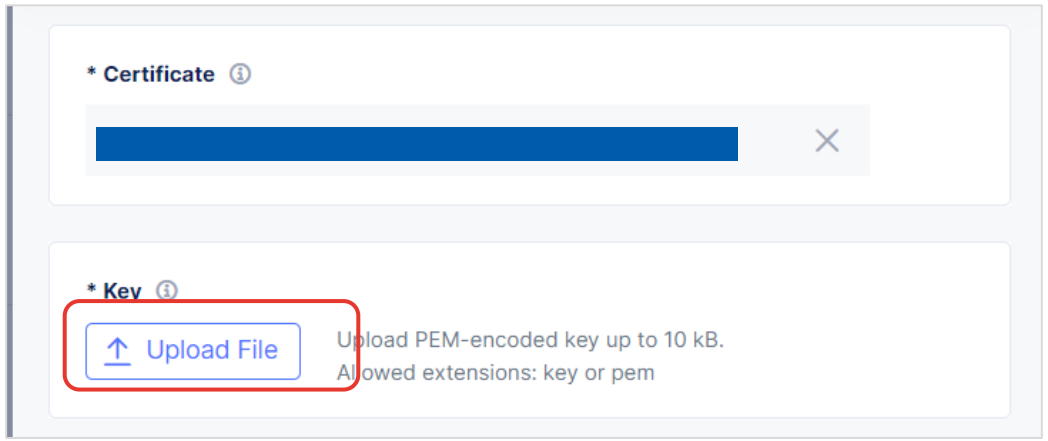


- Certificate

  Upload Fileをクリックして証明書をアップロードします

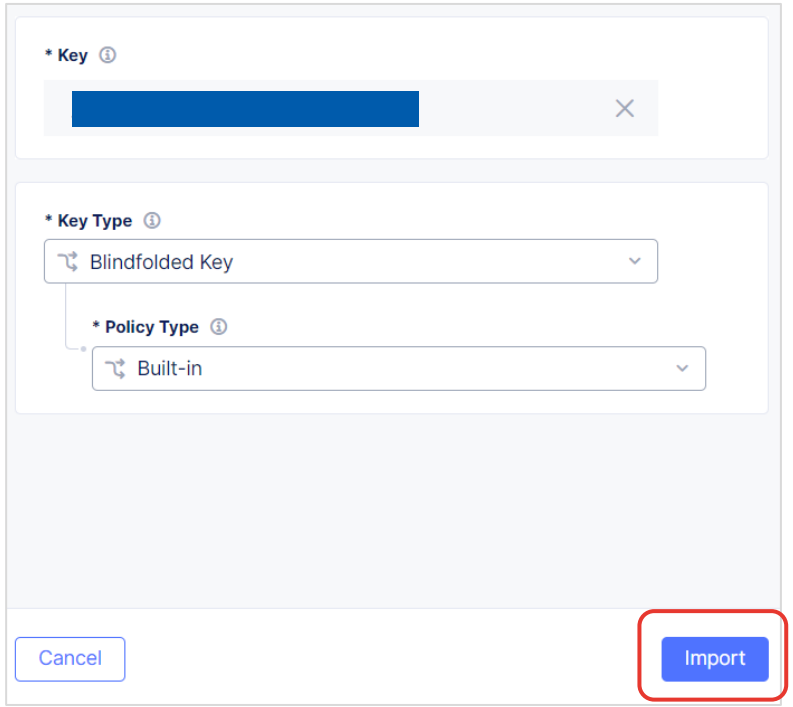
  中間 CA 証明書が存在する場合は、その証明書も含める形で設定欄に張り付けてください

証明書の登録②



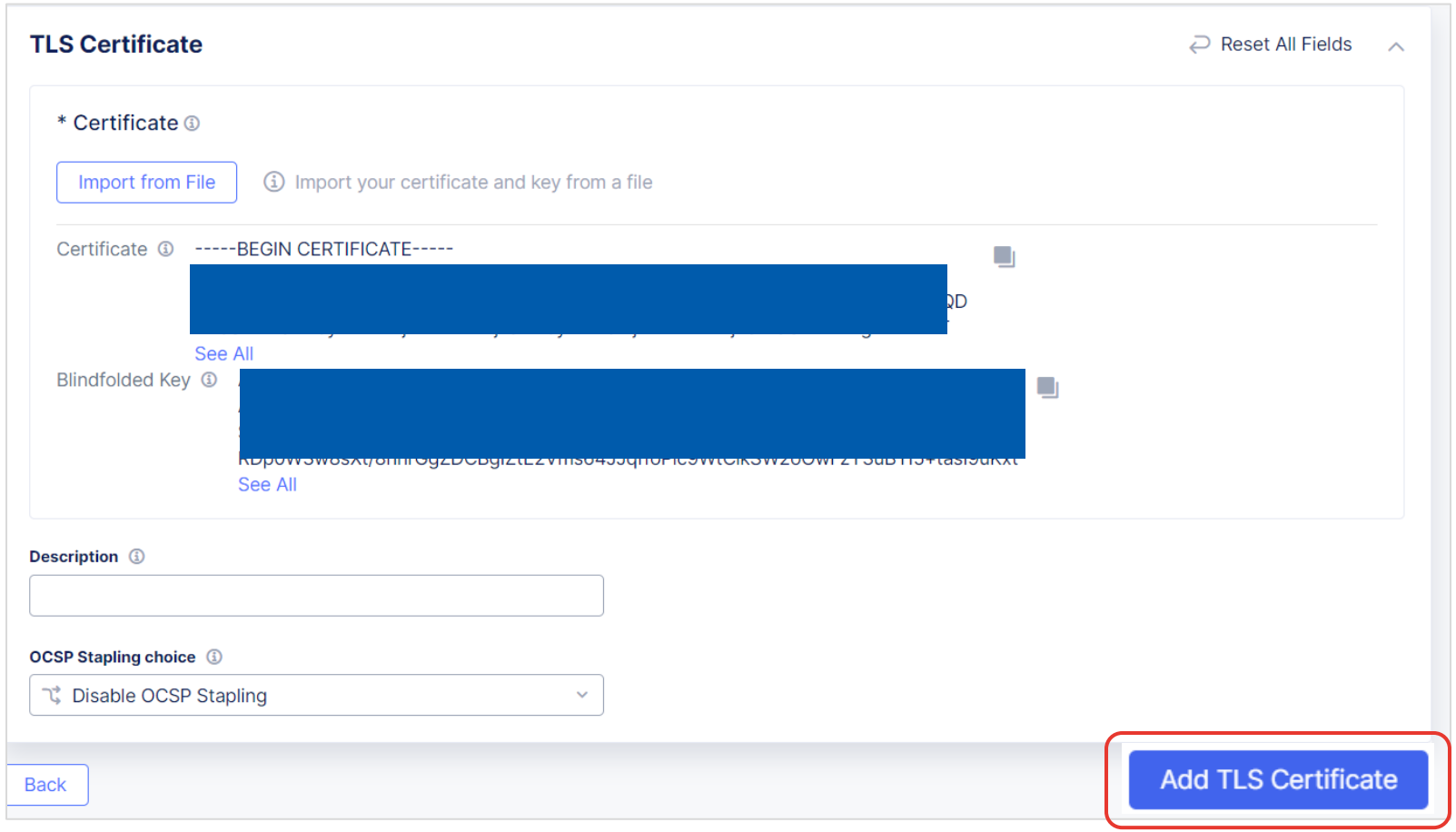
- Key

  Upload Fileをクリックして秘密鍵をアップロードします



Importをクリックします

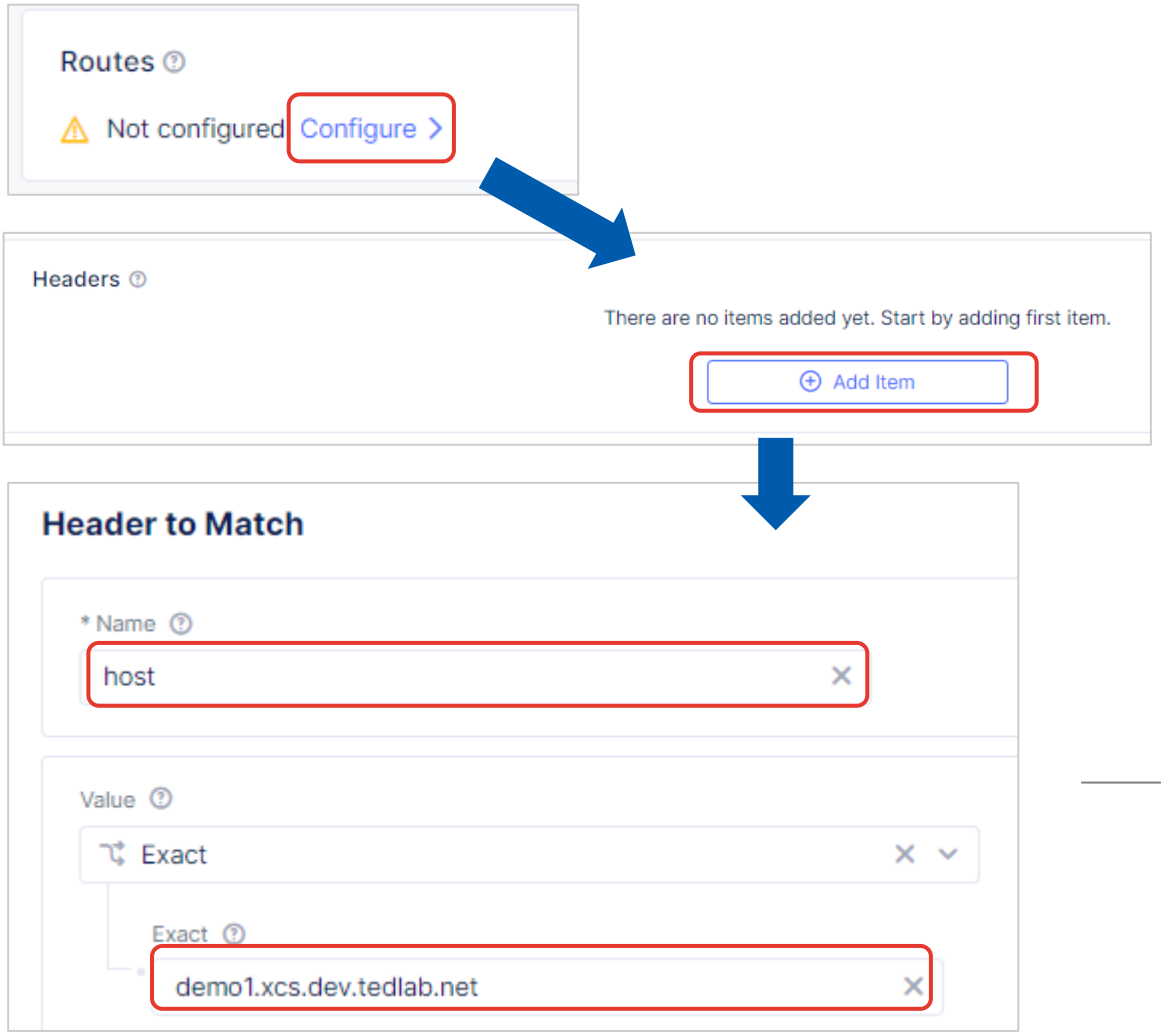
証明書の登録③



Add TLS Certificateをクリックします

FQDN 毎にバランシングを制御①

HTTP Load Balancer の Routes で制御することが可能です
Routes では、クライアントリクエストの host ヘッダを利用することで、バランシング先を変更したり、WAF のポリシーを変更できます



HTTP Load Balancer で任意の host ヘッダを認識させるには、Headers を設定していきます

Header to Match で host ヘッダで制御したい FQDN を入力します

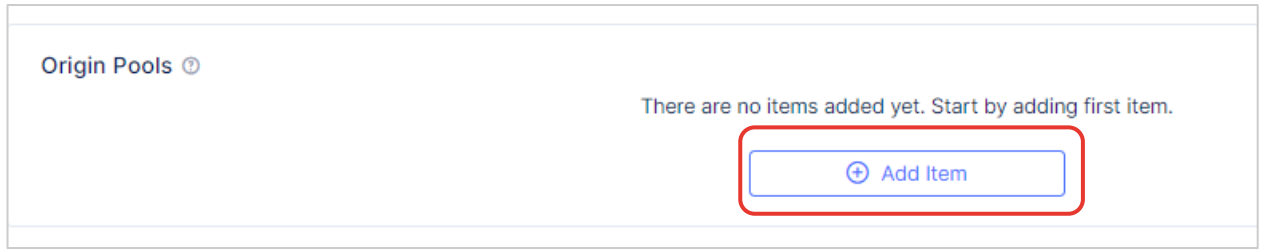
- Name: host
- Value: Exact
- Exact: <対象FQDN名>

※Value: Regexの場合はRegexを指定

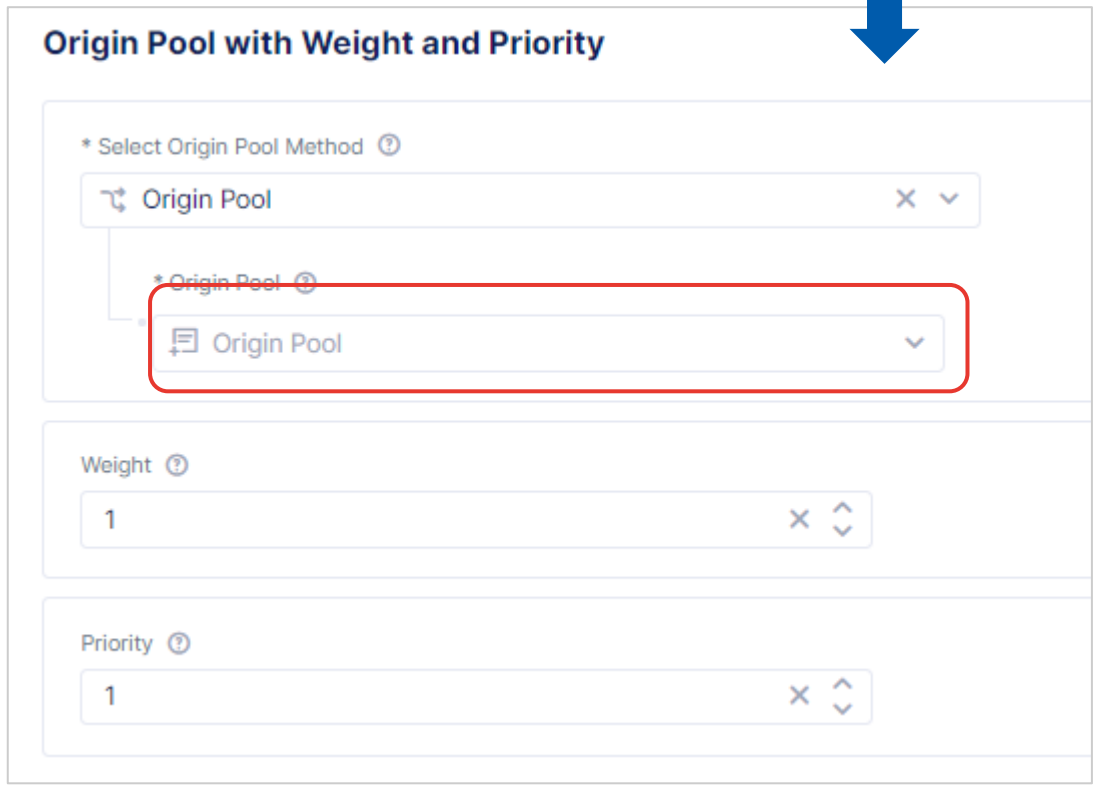
入力後、 Apply を押下して登録を完了します

FQDN 毎にバランシングを制御②

Host ヘッダを指定した後は、その host ヘッダを受け取ったときに、どの Origin Pool にバランシングするか設定します



Origin Pools の項目で、バランシング先を指定します

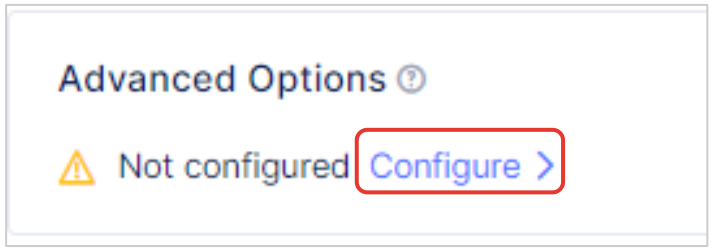


Origin Pools with Weight and Priority で、バランシング先の Origin Pool を指定します

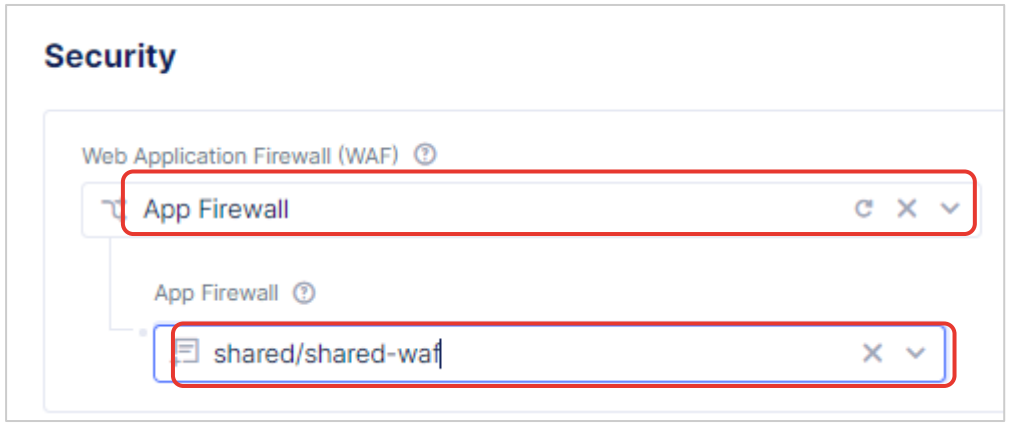
入力後、 Apply を押下して登録を完了します。

FQDN 毎に WAF のポリシーを制御

Routes では、WAF のポリシーを制御することができます。
FQDN や path 毎の制御を組み合わせて使うことで、条件に応じて WAF のポリシーを使い分けることが可能です



WAF のポリシーは、Advanced Options から選択することができます



Security の Web Application Firewall (WAF) で App Firewall を選択します

App Firewall で任意の WAF ポリシーを選択します

入力後、 Apply を押下して登録を完了します

東京エレクトロンデバイス
F5 XC サービスメニュー

東京エレクトロンデバイス F5 XC サービスメニュー

F5 XC デモ

セキュリティ機能を中心にオンライン形式で実施しております
セキュリティ以外の内容についても随時ご相談ください

構築支援メニュー

PoC支援・構築支援サービスをご用意しております
お気軽にご相談ください

日本語サポート

弊社経由でご契約いただいた場合は、日本語での
ヘルプデスクサポートがついてきます

アラート通知サービス

F5XCのセキュリティアラートをTEDでモニターします
アラート発報時は、アラート内容を日本語に翻訳し、
初期対応方法を含めてお客様にメールでご案内する
サービスです

F5 XC の紹介やデモのご要望があればご連絡ください

F5製品の情報発信しています！

F5XCの紹介！ブログ更新中！

https://cn.teldevice.co.jp/blog/search/?q=F5XC

ブログ

YouTube

F5 XC の各機能を動画で体験！

F5 XC - YouTube

NGINXがまるっと分かる！ブログ更新中！

https://cn.teldevice.co.jp/blog/search/?q=NGINX

ブログ

YouTube

NGINXを簡単解説！

F5 NGINX - YouTube