F5 Advanced WAF

Web アプリケーションファイアウォール 簡単セットアップガイド（V14.1 対応）

F5 Networks Japan

目次

1. はじめに

本セットアップガイドにて F5 Advanced Web Application Firewall（以下、Adv.WAF）の設定方法についてご案内します。

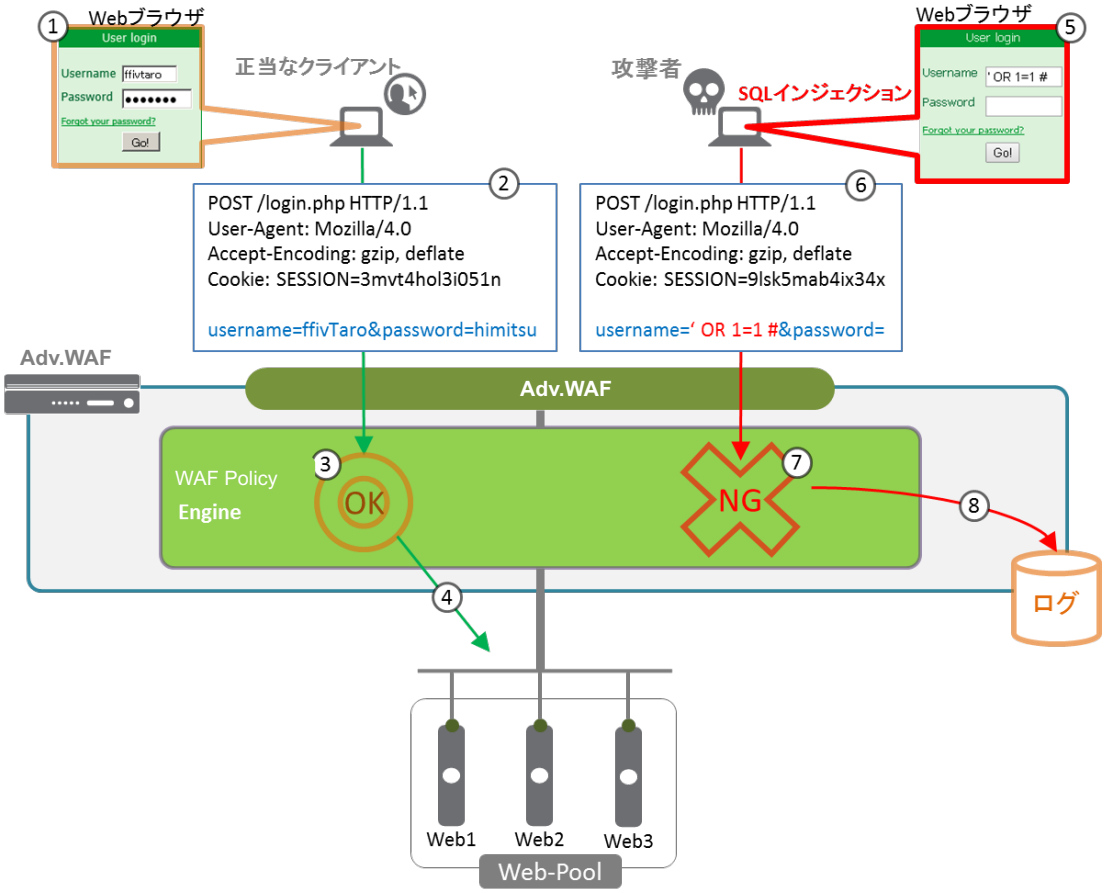
Adv.WAF は、Web アプリケーションファイアウォールです。Adv.WAF によって、Web アプリケーション特有の攻撃に対する防御が可能となります。また、L7 レベルの DoS 攻撃に対する防御機能も兼ね備えています。

Adv.WAF と Local Traffic Manager (以下、LTM) を一つの筐体内に実装することができるので、高度なサーバロードバランシングを行いつつ、Web アプリケーションファイアウォールとしても利用するということが一つのプラットフォーム上で実現できます。

本ガイドでは、Adv.WAF の設定をスムーズに進められるように、必要となる典型的なセットアップ手法を、豊富なスクリーンショットを交えて解説します。

1.1. Adv.WAF 動作概要

Adv.WAF の基本動作イメージを以下に示します。



SQL インジェクション攻撃の例：

① 正当なクライアントが、ブラウザのフォーム認証画面にユーザ名とパスワードを入力して「Go!」ボタンを押す。
② そのユーザ名とパスワードの値が入った HTTP リクエストが、Adv.WAF の VS へ送られる。
③ Adv.WAF は、そのユーザ名とパスワードをチェック。ポリシーに違反していないことを確認。
④ Adv.WAF は Web-pool へその HTTP リクエストを転送する。

⑤ 攻撃者が、ブラウザのフォーム認証画面で、ユーザ名として「' OR 1=1 #」と入力して「Go!」ボタンを押す。(SQL インジェクション攻撃)
⑥ ユーザ名の変数に値：「' OR 1=1 #」が入った HTTP リクエストが、Adv.WAF の VS へ送られる。
⑦ Adv.WAF は、そのユーザ名とパスワードをチェック。不正な値：SQL インジェクション攻撃であることを検知。
⑧ その攻撃をブロックし、ログとして記録する。

以降、このような Web アプリケーションを狙った攻撃を防御する設定方法を示します。

2. Adv.WAF ポリシー作成の概要

このセクションでは、Adv.WAF ポリシー作成ステップの概要を記載します。

ポリシー作成にはいくつかのアプローチがありますが、本ガイドの作成ステップにより、早く且つかなり高い精度で攻撃を防御できるポリシーが生成できます。

本ガイドでは、Web アプリケーションのセキュリティポリシーを素早く作成することを目的とした、Rapid Deployment Security Policy (以下、RDSP) をベースポリシーとして利用します。

［Note］
Adv.WAF ポリシーの作成方法には他に、Policy Builder 機能 (自動的にパラメータ、URL、ファイル拡張子などを学習して、ポリシーを自動作成する機能)があります。しかし、以下の理由から本ガイドの対象外としています。

- 学習の期間として 1 ヶ月～3 ヶ月を基本とする。(時間がかかる)
- 自動的にポリシーが作られるので、どのような設定がなされたのかを管理者が把握しづらい
  (Adv.WAF の深い理解が必要)
- 管理者が、自動的に生成されたポリシー設定の変更要否の最終判断をしなければならない
  (Adv.WAF の深い理解が必要)

2.1. ［Step1］ベースとなる Adv.WAF ポリシーを作成する

(1) Rapid Deployment Security Policy（RDSP）の利用

RDSP には以下のセキュリティチェック機能がデフォルトで含まれています。（v13.1.x 現在）

Performs HTTP compliance checks	HTTP プロトコル標準に合致しているかのチェック
Checks for mandatory HTTP headers	必要な HTTP ヘッダのチェック
Stops information leakage	情報漏えい停止
Prevents illegal HTTP methods from being used in a request	HTTP Request の中の違反 HTTP method を防御
Checks response codes	Response コードをチェック
Enforces cookie RFC compliance	Cookie が RFC 標準に合致しているかのチェック
Applies attack signatures to requests	攻撃シグネチャによる検知
Detects evasion technique	イベーション（回避）テクニックの検知
Prevents access from disallowed geolocations	無許可の地域からのアクセスを防御
Prevents access from disallowed users, sessions, and IP addresses	無許可のユーザ、セッション、IPアドレスからのアクセスを防御
Checks whether request length exceeds defined buffer size	定義済みバッファサイズ超過のチェック
Detects disallowed file upload content	無許可のアップロードファイルコンテンツの防御
Checks for characters that failed to convert	コンバートに失敗した文字のチェック
Looks for requests with modified ASM cookies	ASM Cookie が改ざんされたリクエストかどうかのチェック

その他、RDSP の設定において、以下 2 つについての無効化を検討します。

① 無効化を推奨する機能
- Failed to convert character: 日本語サイトの場合、誤検知しやすいため。

② 無効化を検討すべき機能
- Data Guard: パフォーマンスへの影響を考慮し、不要なら OFF にすべき。

(2) シグネチャセットの割当て

利用するアプリケーションに応じて、最小限の攻撃シグネチャセットを適用します。

2.2. ［Step2］ シグネチャのチューニング （誤検知の対策）

誤検知が発生した場合に対して、以下 3 つの手段のいずれかを利用します。

① ［対策 1］誤検知したパラメータをホワイトリスト化する（シグネチャ検知はしない状態にする）

② ［対策 2］シグネチャの誤検知が発生したパラメータで、そのシグネチャを無効化する（ホワイトリスト化する）

③ ［対策 3］誤検知したシグネチャを全体で無効化し、そのシグネチャで守りたいパラメータだけに適用する

2.3. ［Step3］ File Type の設定を行う（強制ブラウジング対策）

強制ブラウジング攻撃の対策として、File Type の設定を行います。

(1) File Type の学習

Web アプリケーションページの巡回により、File Type を学習します。

(2) File Type の設定

その学習結果を見て、以下 2 つのどちらの対応を取るかを判断し、設定します。

① Disallow File Type で対応する （利用しない File Type で、危険と判断されるものを指定する）

② Allow File Type で対応する （利用する File Type をすべて指定する）

2.4. ［Step4］ 追加機能の要否検討

以下の機能が必要かどうかを判断し、必要なものを有効にします。

① Data Guard（クレジットカード番号のマスキング）
② L7 DoS への対策
③ Brute Force への対策
④ CSRF への対策

2.5. 最終的なチューニング

［Step1］～［Step4］によって、Web アプリケーションへの攻撃はかなり高い精度で防御可能と考えます。

ここまでのステップで対応できない攻撃に対する対策としては、以下が考えられます。

① ホワイトリスト利用による詳細設定

② カスタマイズしたシグネチャ作成による対策

③ iRules による攻撃回避

上記①の手法については、サンプルとして 2 つの攻撃手法とその防御設定にて解説します。

上記②及び③については、大半のケースがお客様の環境依存となることから、ヒアリングを実施させて頂き、その情報を元に対策する必要がでてきます。
この対応については F5 コンサルティングサービスにてお受けいたしますので、必要となった際には弊社へご連絡ください。

2.6. ［ご参考］利用するセキュリティ機能の優先度

Adv.WAF のセキュリティ機能は豊富なので、どの機能を使えばよいか判断に迷うことがあります。

以下表は一つのガイドラインとして、一般的な Web アプリケーションで利用すべき機能の優先度を記載しました。

Web アプリケーション次第で利用すべき機能は異なりますが、目安として参考にしてください。

◎：ご利用を推奨する機能です。
〇：多くの Web アプリケーションで適用することが望ましいと考えられる機能です。
▽：必要に応じて適用、という位置づけです。

	設定項目		推奨	コメント
1	**Rapid Deployment Security Policy**		◎	一部、無効化を推奨する機能あり（既述）
2	**シグネチャ**		◎	必要最小限のシグネチャセットのみを適用。
3	ホワイトリスト	**File Type**	◎	Adv.WAF を通過させるファイルを指定。 ファイルタイプ数によっては Disallow の設定の方が、管理が楽だと思われる。
		URI	▽	これらはアプリケーションによっては数が膨大になる。 よって、例外処理が必要となるものだけ抜き出して設定したほうが良い。
		Parameter	▽	
		Header	▽	例) Cookie の改ざんによる影響が考えられる場合。
4	**DoS**		〇	多量のリクエストによるアプリケーションへの影響が懸念される場合
5	**Brute Force**		〇	不正ログインを防止したい場合
6	Web Scraping		▽	ボットによる情報収集を拒否したい場合
7	**CSRF**		〇	CSRF による攻撃を守りたい場合
8	Data Guard		▽	クレジットカード番号漏洩がありえる場合
9	Session 管理		▽	ログイン後にしか表示しない画面を制限したい場合

2.7. ［ご参考］Evasion（回避）テクニックとは

RDSP によって有効になる機能のうち、Evasion テクニックとはどのようなものかがやや分かりにくいと思いますので、いくつかの具体的な攻撃をサンプルとして解説します。

- ディレクトリ・トラバーサル（Directory Traversal）

  URL の一部ではない、「../」のような文字例を URL に組み込んで HTTP リクエストを送ってくる攻撃。
  例えば、攻撃者から「http://example.com/../../../private_file.txt」というリクエストが発せられたとき、システムは、Web アプリケーションがユーザに提供すべきではないファイルを返してしまうかもしれない。

  この攻撃は、Web Server root をバイパスして、さまざまなファイルをリクエストすることを目的としている。
  この攻撃によって、システムファイル、またはプライベートディレクトリ・リソースが奪取される可能性がある。

- マルチプル・エンコーディング（Multiple Encoding）

  複数回の%エンコーディング(URL エンコードともいう)を繰り返し実施することで、文字を隠蔽する攻撃。
  例えば、%エンコーディングによって、「<」は「%3c」に変換される。攻撃者は、これを攻撃に使うために、もう一度エンコードして隠蔽する。「%3c」をもう一度エンコードすると、「%253c」になる（「%」はエンコードすると「%25」になる）。

  この攻撃によってメタキャラクタを隠蔽できるので、クロスサイトスクリプティングやディレクトリ・トラバーサル、SQL インジェクションなどのような攻撃を、一回のデコーディングでは検知できないようにすることが可能になる。

- ベア・バイト・デコーディング（Bare byte decoding）

  ASCII 文字は 7 ビットで英数字と少数の記号を表現するコードであり、0x00～0x7F の 128 種類の文字を扱う。
  0x00～0x7F の範囲より上の 0x80～0xFF の範囲の文字（＝最上位ビットが 1 の文字であった場合、ブラウザによっては最上位ビットが無視され、0x00～0x7F として扱われる。

  例えば、メタキャラクタ:「<」は ASCII では「0x3C」なので、Web アプリケーションは「0x3C」の入力をエスケープすることでクロスサイトスクリプティング攻撃を防いでいる。しかし、「0xBC」が防がれていないと、ブラウザ上では「<」として処理され、クロスサイトスクリプティング攻撃が成立してしまう、という可能性がある。

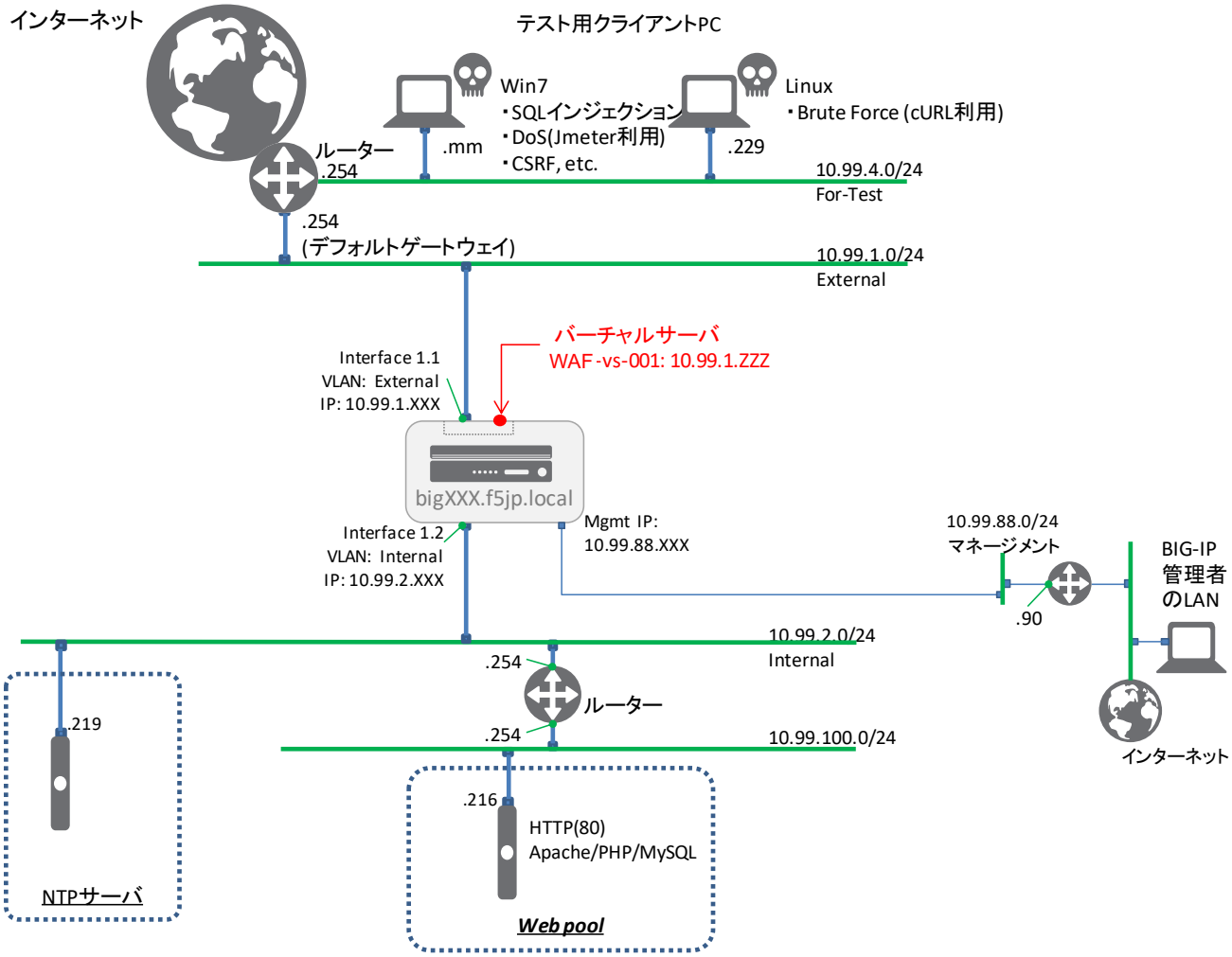
	US-ASCII		**最上位ビットを 1 にセット**	
文字	16 進	2 進	16 進	2 進
"	0x22	00100010	0xA2	10100010
<	0x3C	00111100	0xBC	10111100
>	0x3E	00111110	0xBE	10111110

Evasion テクニック検知を有効にすることで、このような既知の攻撃を防御することが可能となります。

3. スタンドアローン

3.1. スタンドアローンのネットワークサンプル

まずは、冗長化しない状態を想定して、1 台のみ設定していきます。



ここでは、Adv.WAF の Virtual Server は 10.99.1.ZZZ:80 を設定します。

プールメンバーには、以下の WEB サーバ:10.99.100.216:443 を設定します。

Adv.WAF のデフォルトゲートウェイは、インターネット方向を想定したルーター:10.99.1.254 に設定します。

Web サーバのデフォルトゲートウェイは、Adv.WAF の Internal インタフェース宛(10.99.2.XXX)に設定します。

動作確認は、テスト用に設置した PC(図中の「テスト用クライアント」)から行うこととします。

3.2. 構成の概要

テスト用クライアント PC からバーチャルサーバ：WAF-vs-001 に対して、Web アプリケーションに対する攻撃を仕掛けます。Adv.WAF によって、その攻撃をどのように検知・防御するかを記載します。

本ガイドで攻撃対象となる Web アプリケーションは、Apache, PHP, MySQL で構成されたオークション用サイトを利用します。

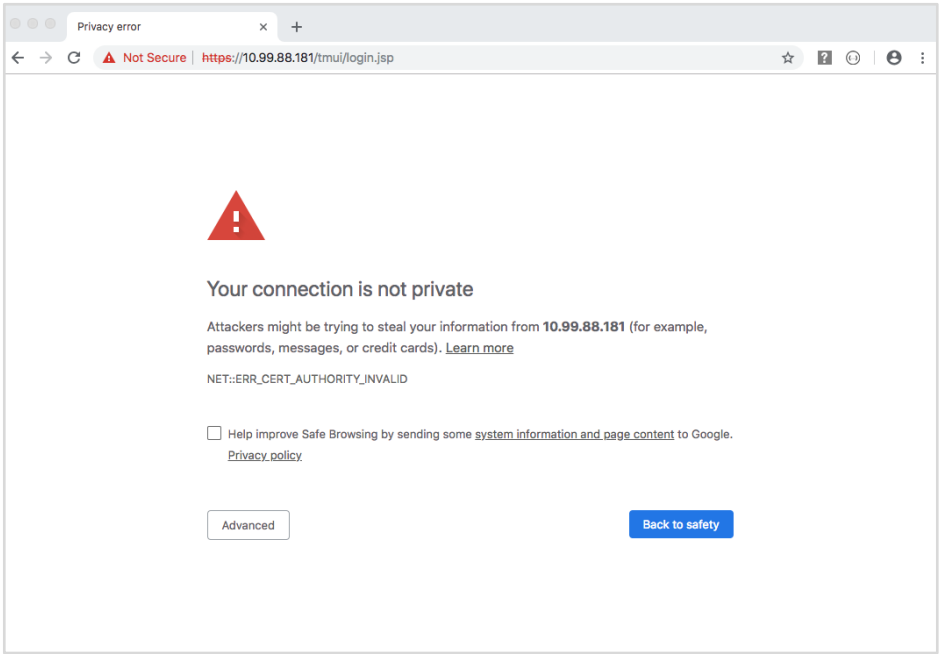
Web アプリケーションにログイン可能なユーザとして、F5 LAB 環境では 8 つ（f5user1～f5user8）を設定しています。また、本ガイドは、Adv.WAF の初期設定から LTM のサーバロードバランシング設定ができるまでの基本的な知識は習得済みである前提とします。

また、管理用のマネージメント IP アドレスは設定済みである前提とします。

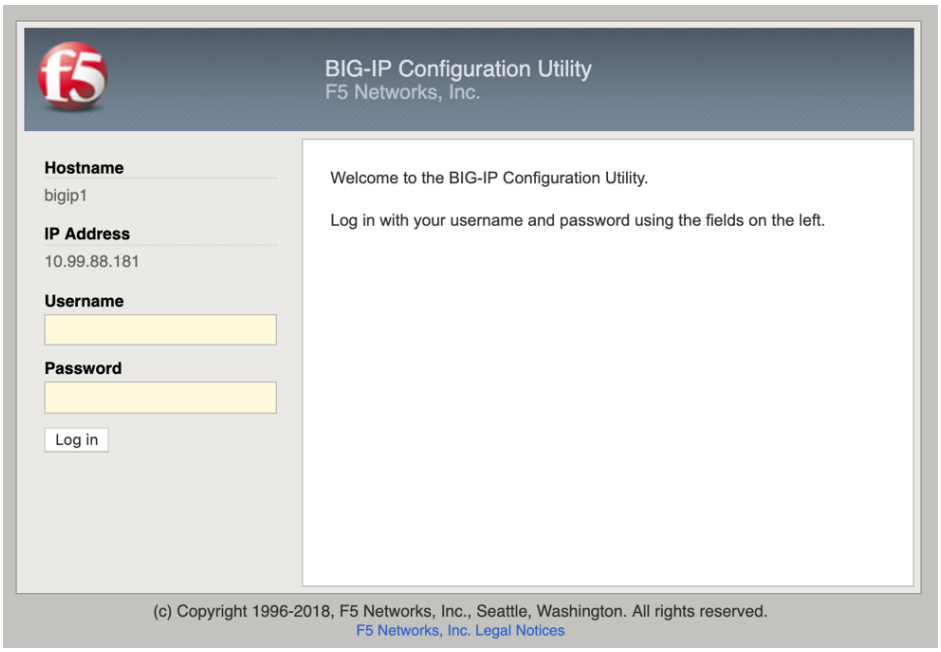
3.3. 初期設定

3.3.1. 管理ポートへの GUI アクセス

(1) 管理用 PC から、設定した BIG-IP の管理 IP アドレスへ、HTTPS でアクセスします。デフォルトの証明書は、正式に取得した証明書ではないため、以下のような画面が現れますが、「続行する」を選択してください。

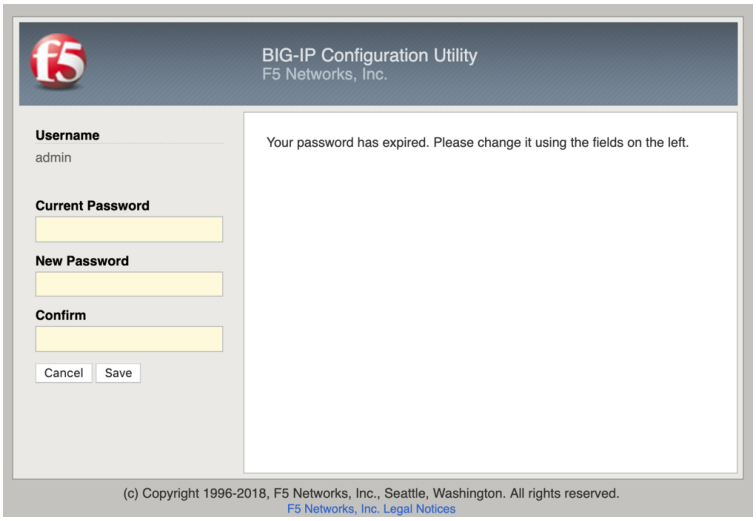


(2) ログイン画面が現れますので、以下のデフォルトの ID と Password でログインしてください。
ID：admin
Password：admin

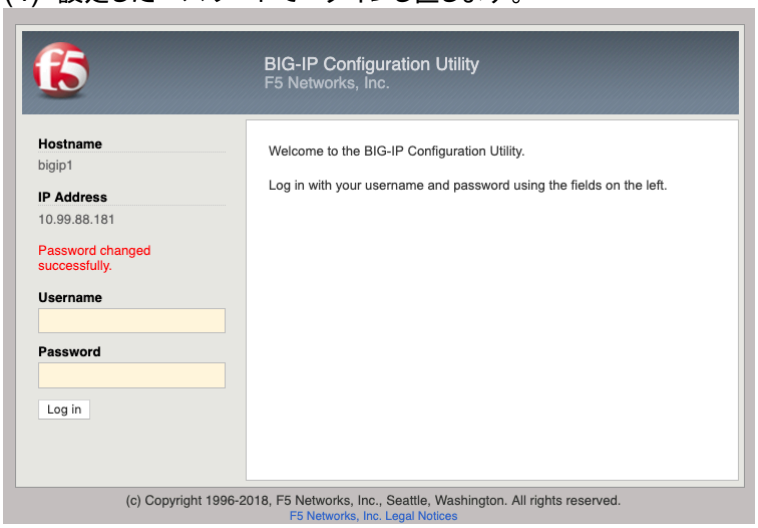


(3) バージョン 14.0 より、デフォルトで BIG-IP のセキュアパスワードポリシーが有効となっています。パスワードポリシーを変更しない限り、v13.0 以前のデフォルトパスワードは利用できません。

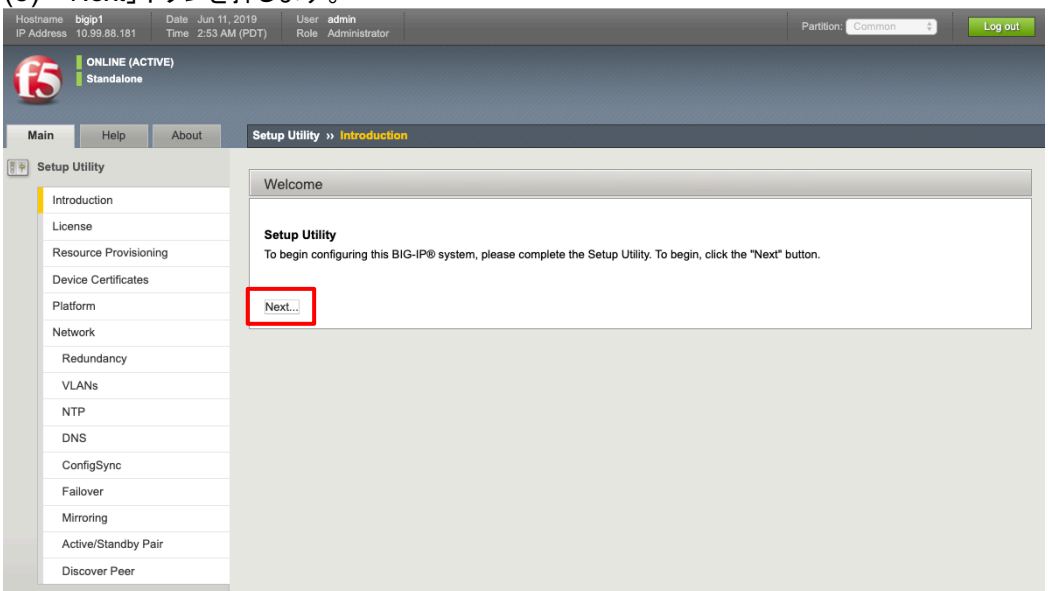
F5LAB では以下のように設定し、Save ボタンを押します。
Current Password: admin
New Password: ilovef5
Confirm: ilovef5



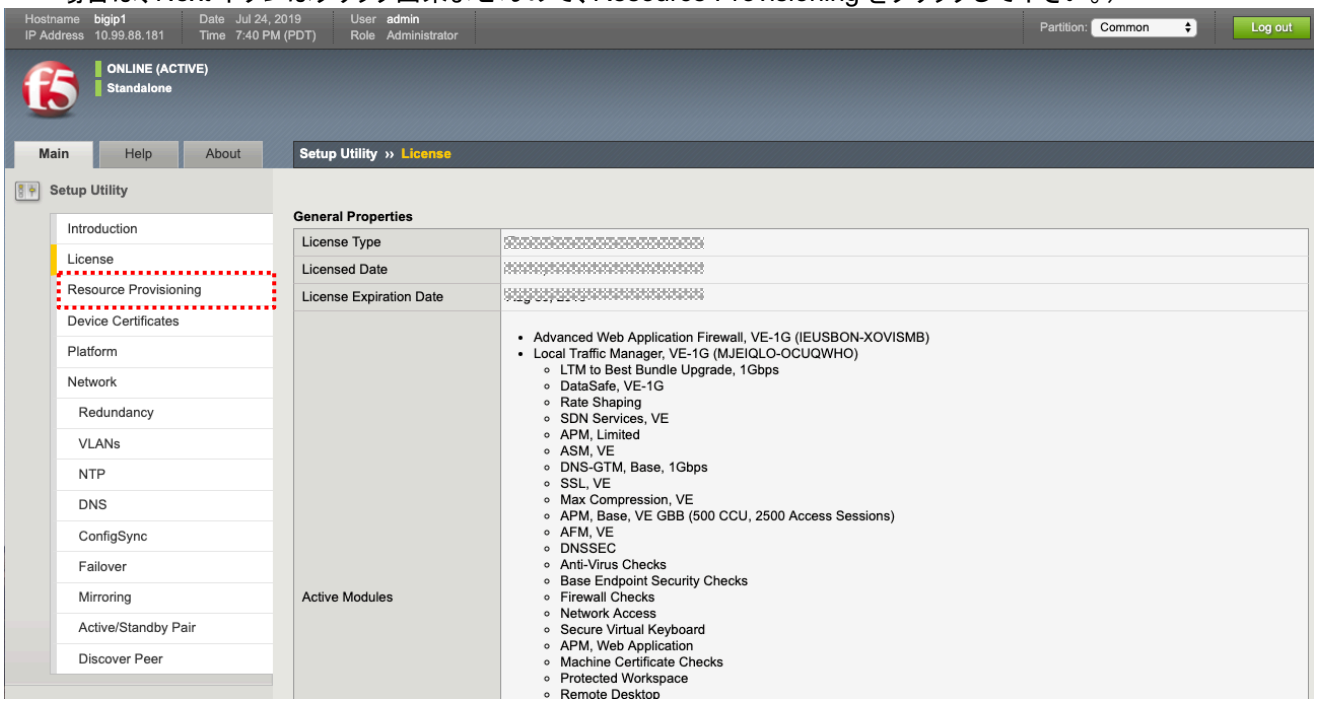
(4) 設定したパスワードでログインし直します。



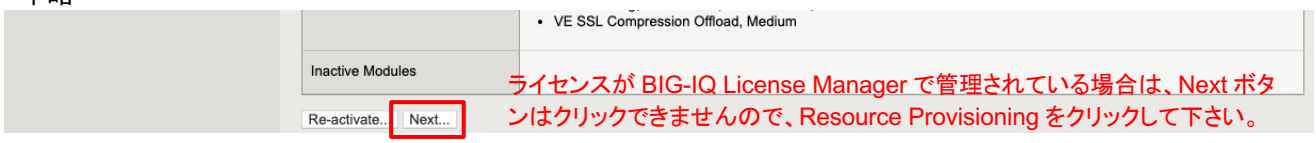
(5) 「Next」ボタンを押します。



(6) ライセンス画面が出ます。「Next」ボタンを押します。（ライセンスが BIG-IQ License Manager で管理されている場合は、Next ボタンはクリック出来ませんので、Resource Provisioning をクリックして下さい。）

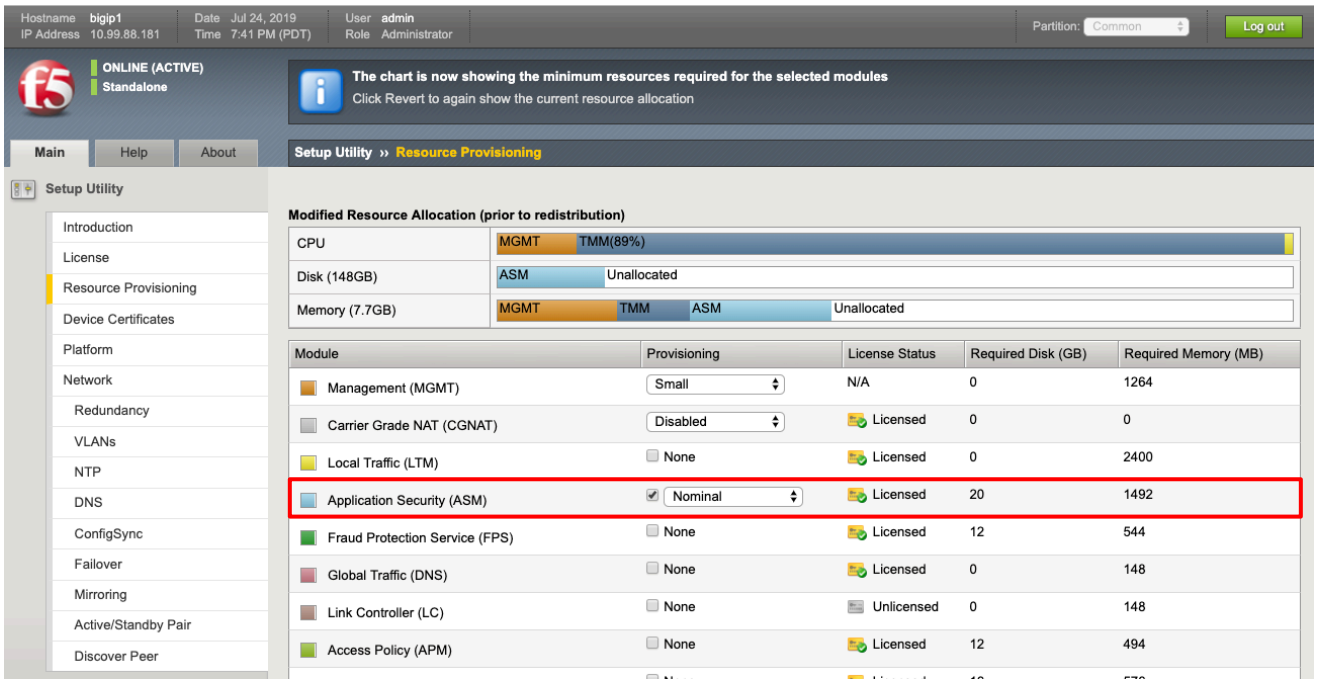


~中略~



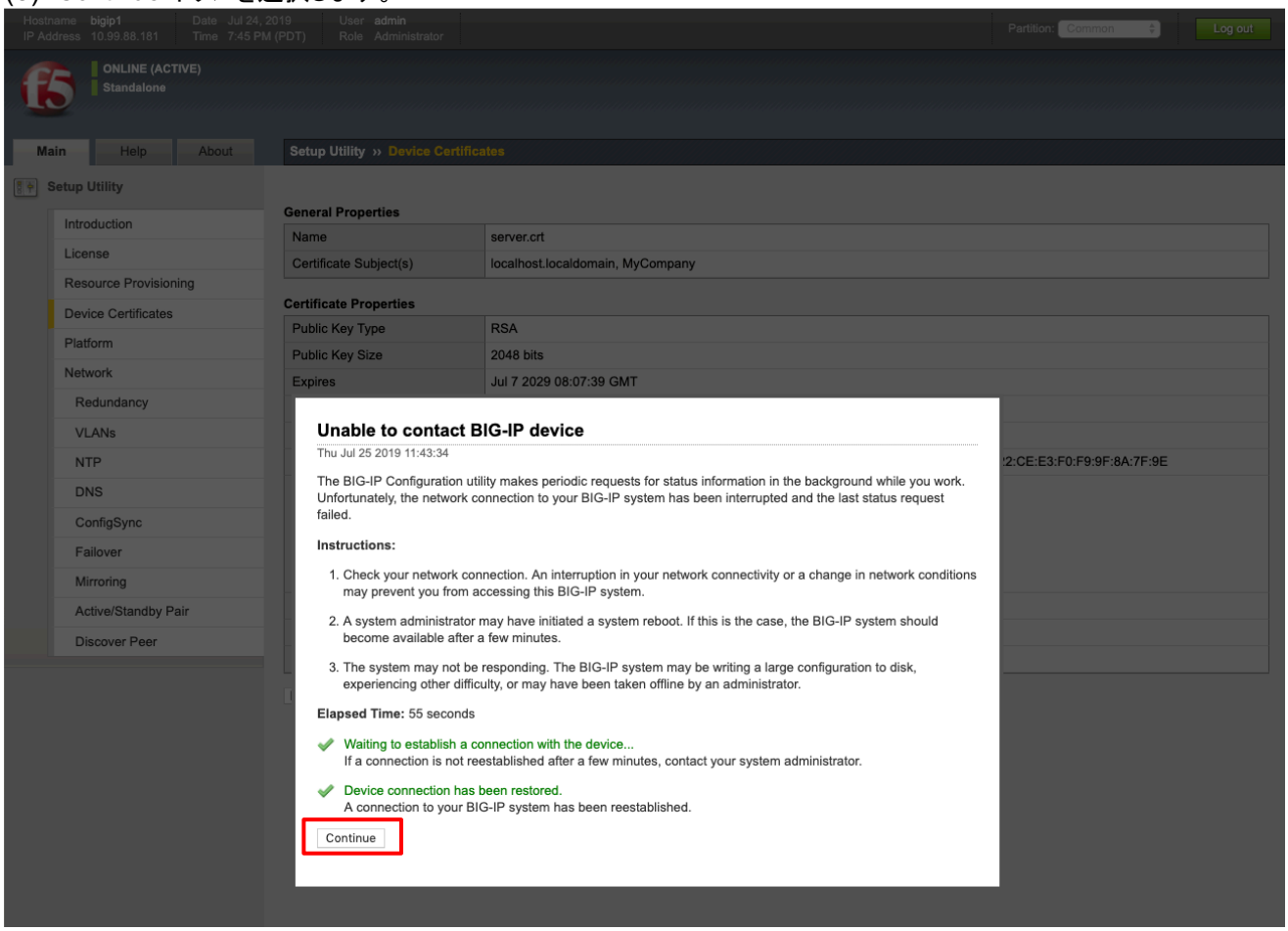
ライセンスが BIG-IQ License Manager で管理されている場合は、Next ボタンはクリックできませんので、Resource Provisioning をクリックして下さい。

(7) プロビジョニング画面がでますが、デフォルトで選択されている LTM をはずし、ASM にチェックを入れて、Nextボタンを押します。（プロビジョニングにはしばらく時間がかかります。）

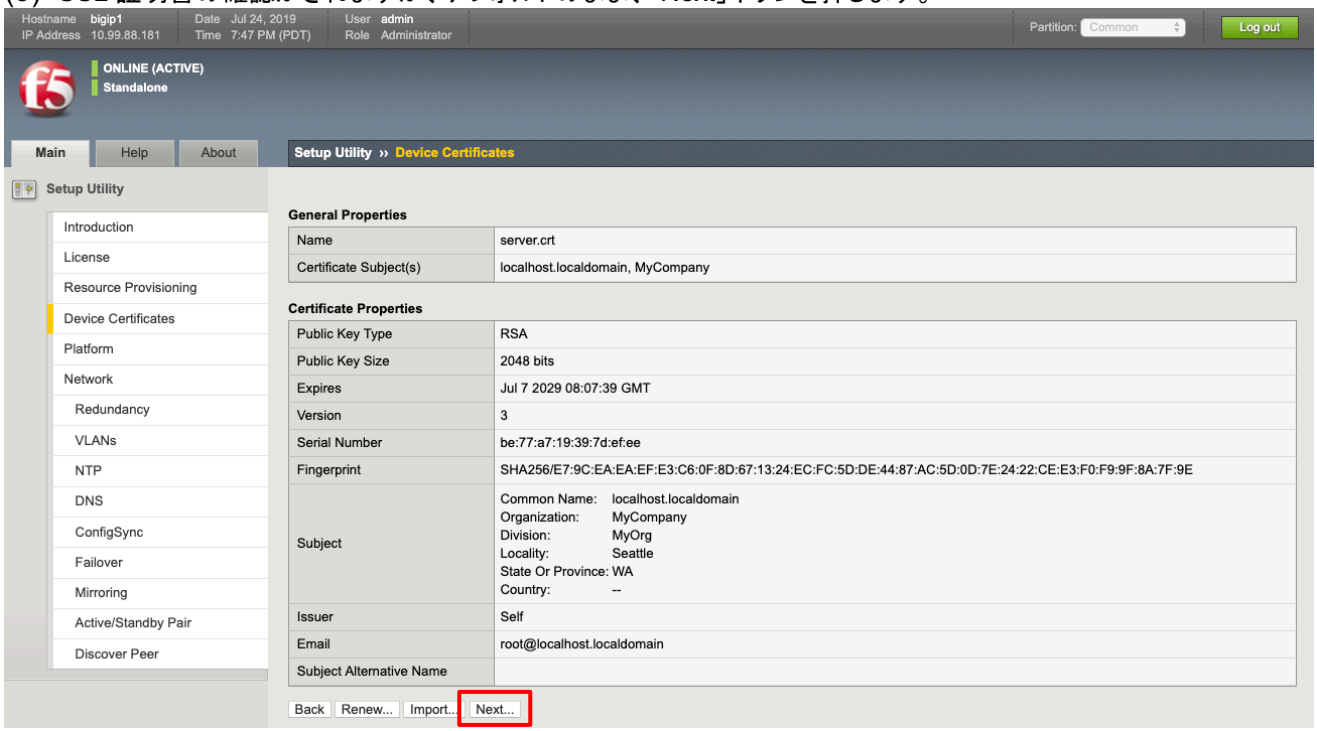


~中略~

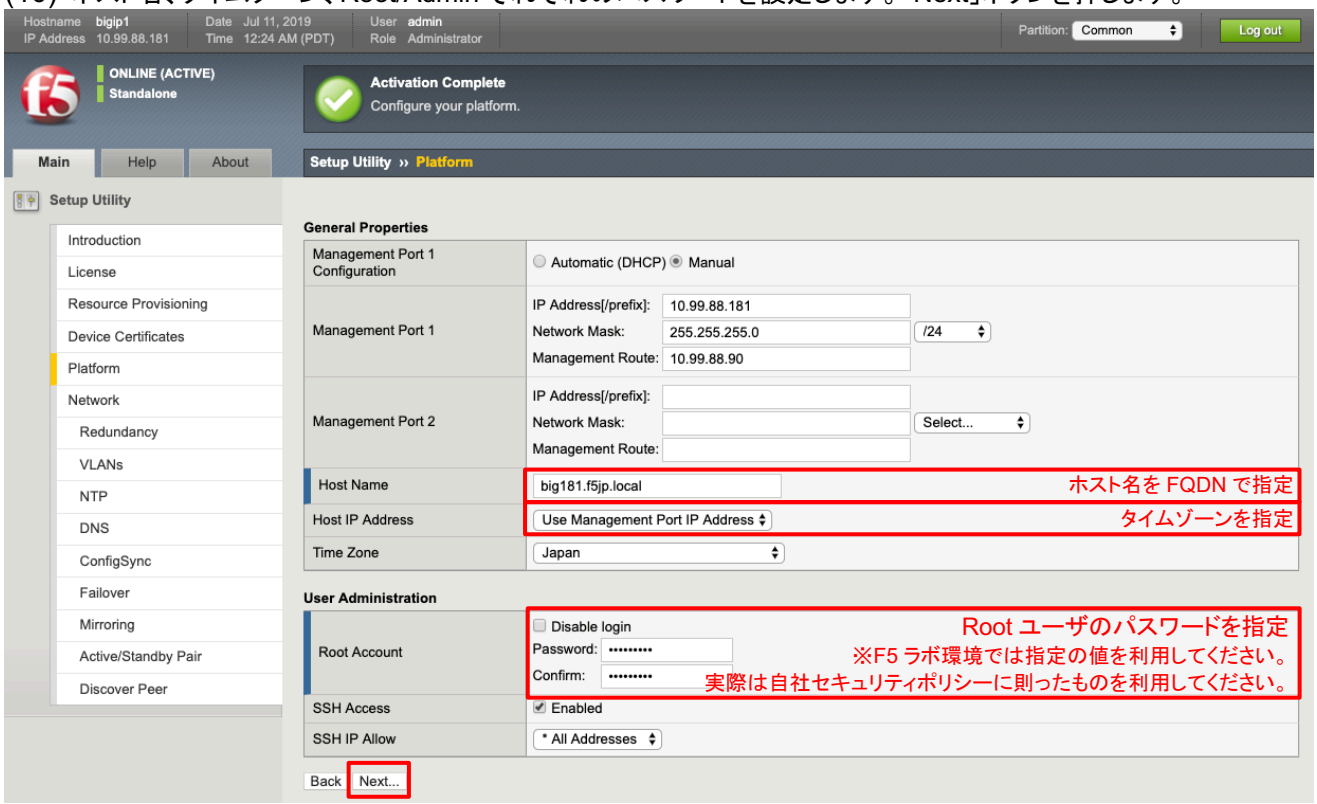
(8) Continue ボタンを選択します。



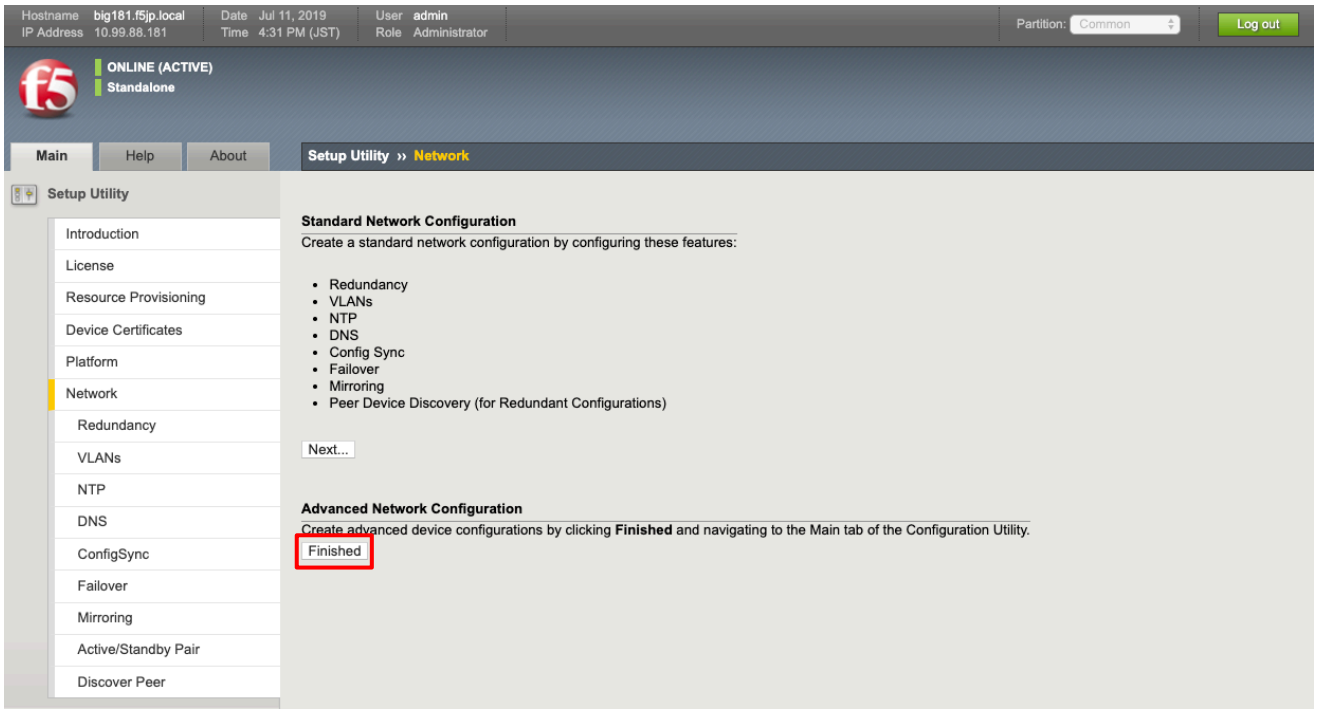
(9) SSL 証明書の確認がされますが、デフォルトのまま、「Next」ボタンを押します。



(10) ホスト名、タイムゾーン、Root/Admin それぞれのパスワードを設定します。「Next」ボタンを押します。



(11) この後、Standard Network Configuration の「Next」を押すことでウィザード形式にて冗長化も含めた設定が可能ですが、ここではスタンドアローン構成にするため、Advanced Network Configuration の「Finished」ボタンを押します。



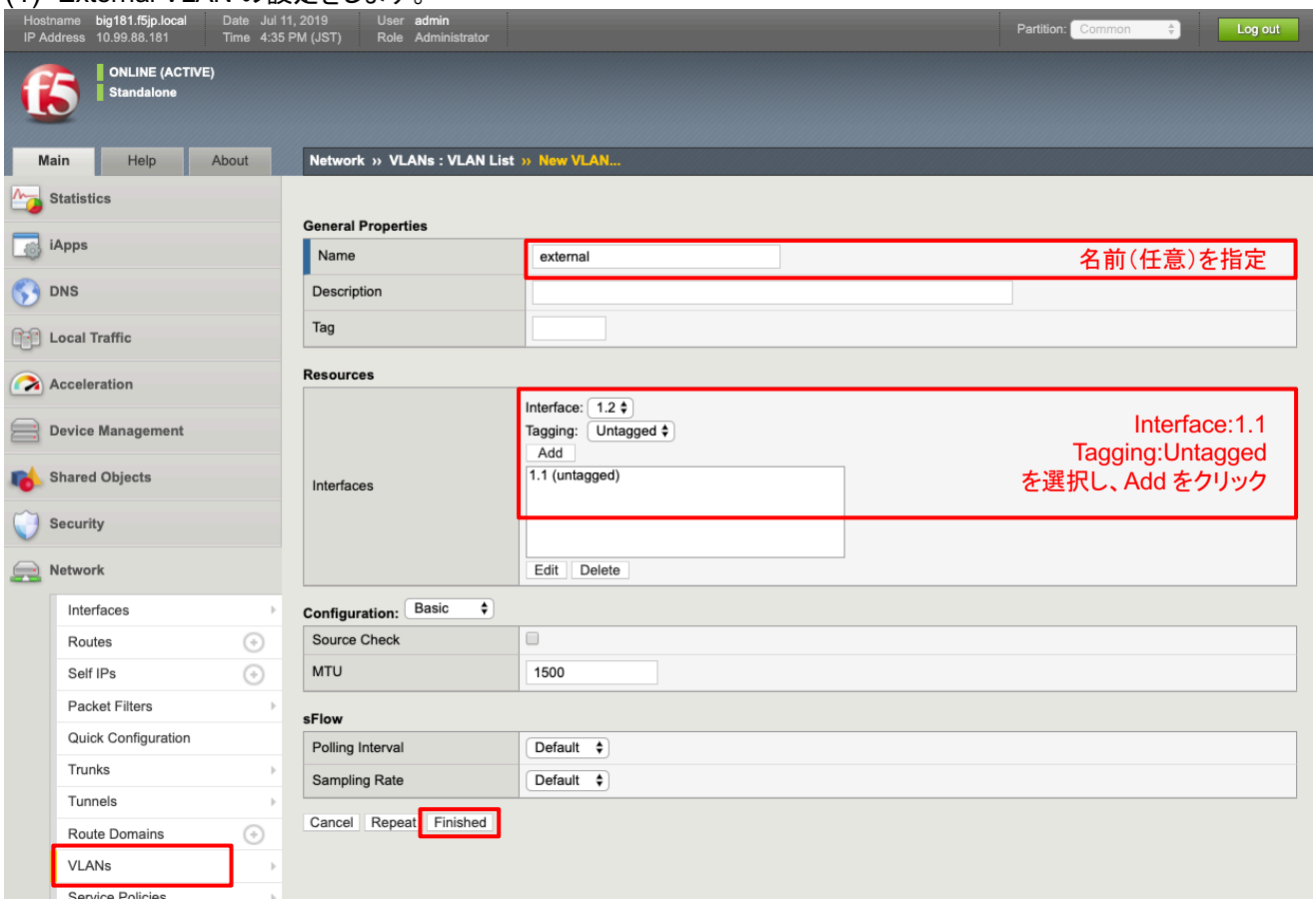
4. ネットワークの設定

ネットワーク構成に合わせて、VLAN，VLAN インタフェースへの IP 設定およびルーティング設定を行います。

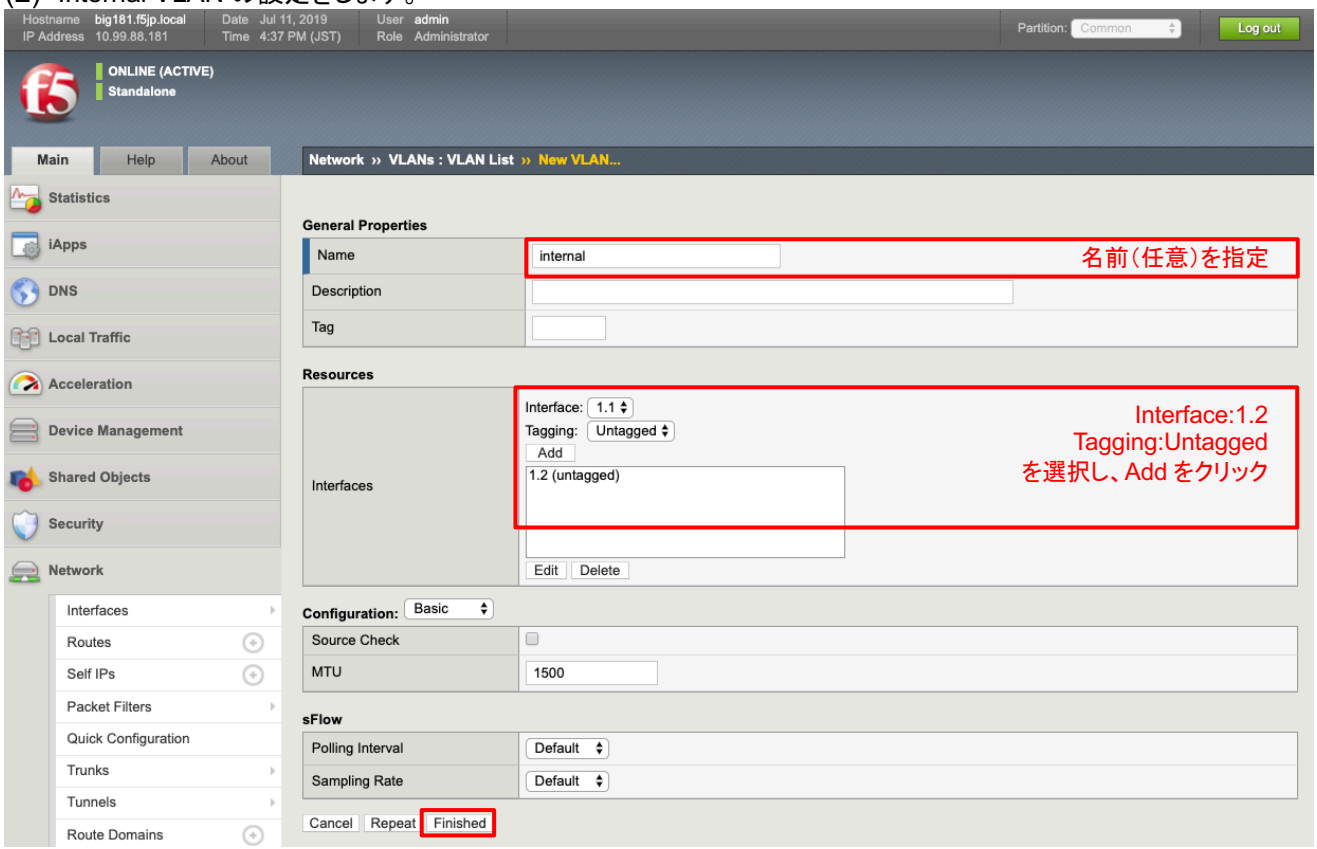
4.1. VLAN の作成

まず、VLAN を作成します。「Network」→「VLAN」で表示された画面の右上にある「Create」ボタンを押します。

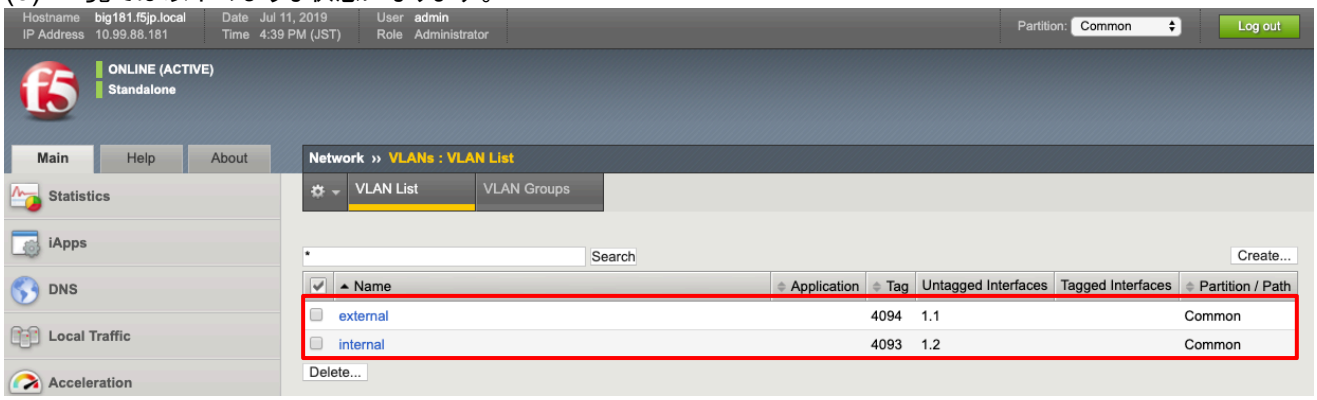
(1) External VLAN の設定をします。



(2) Internal VLAN の設定をします。



(3) 一覧では以下のような状態になります。

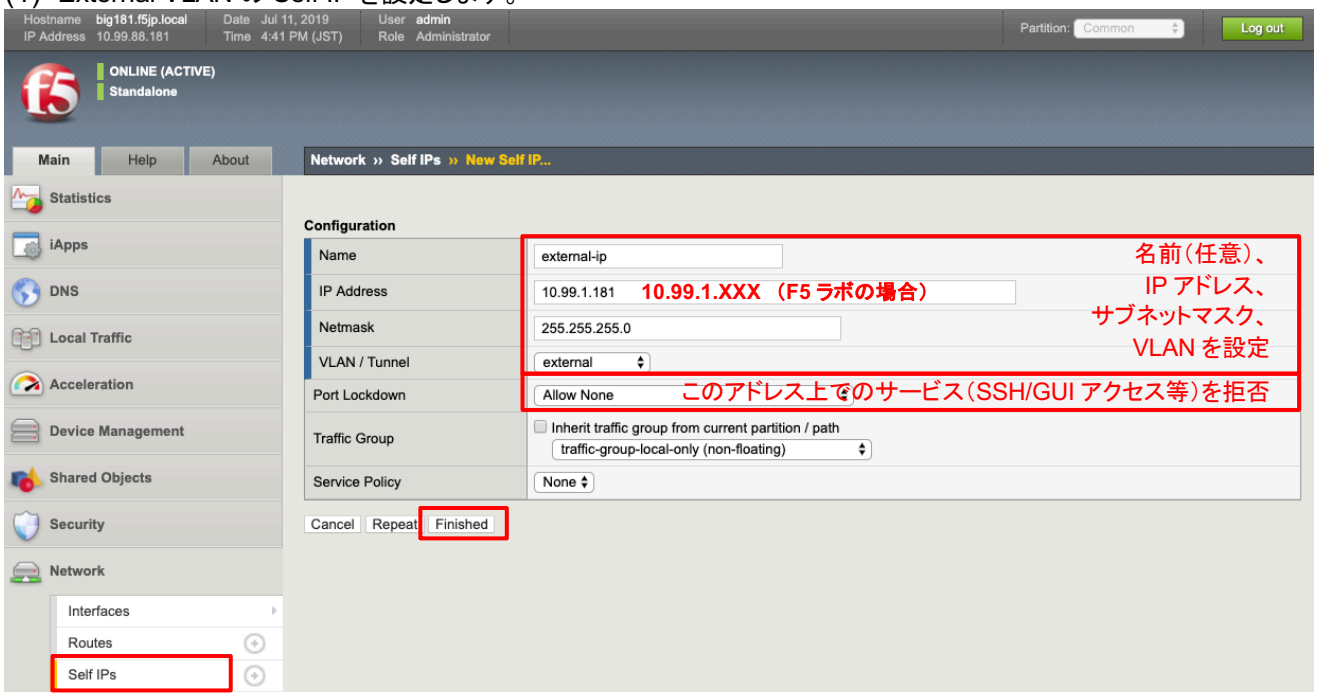


4.2. Self IP の設定

設定した VLAN それぞれに対して、IP アドレスを設定していきます。
Adv.WAF 自身に設定する IP アドレスを、Self IP と呼びます。

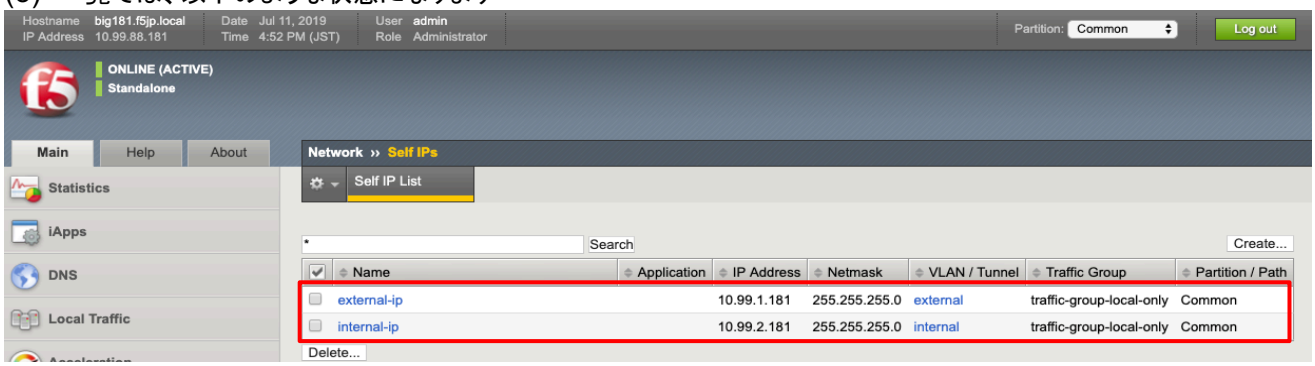
「Network」→「Self IPs」で表示された画面の右上にある「Create」ボタンを押します。

(1) External VLAN の Self IP を設定します。



(2) internal VLAN の IP 設定

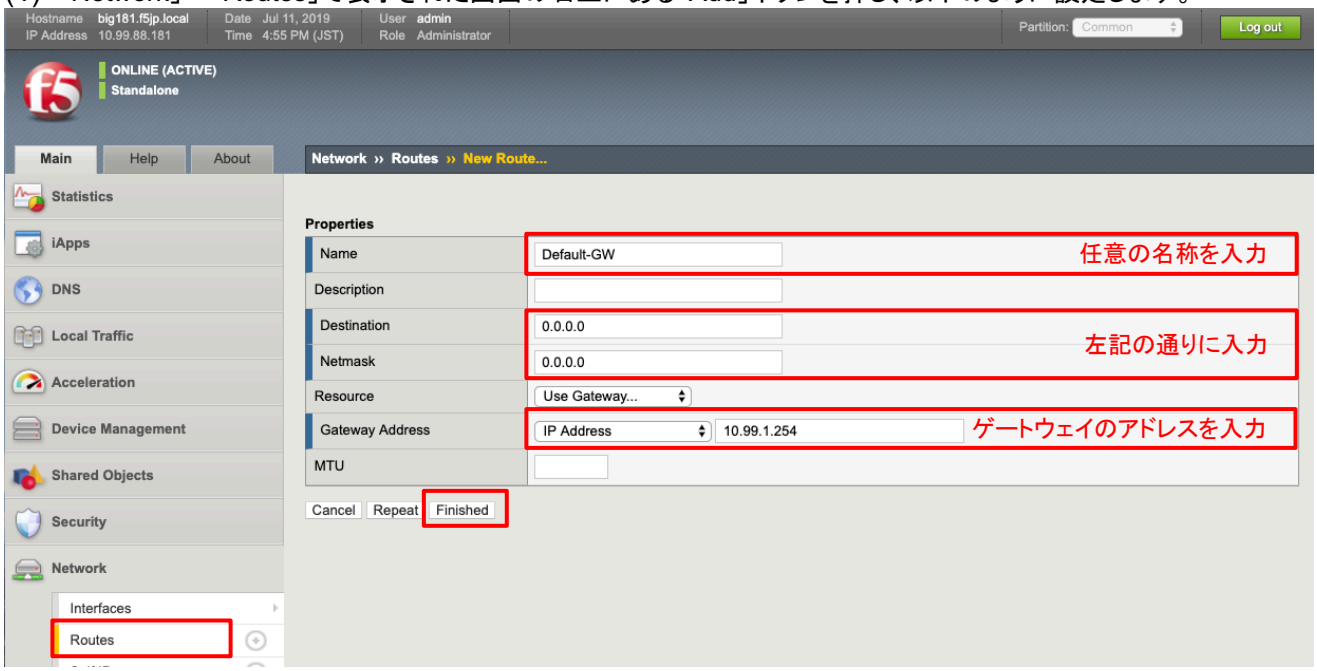
(3) 一覧では、以下のような状態になります



4.3. ルーティングの設定

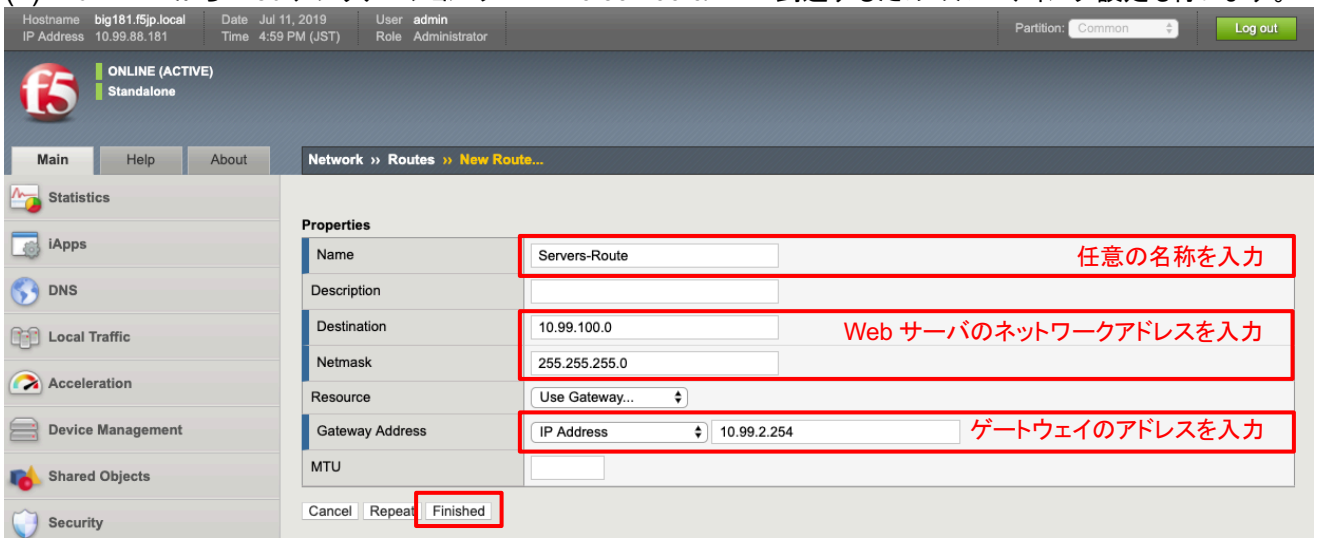
4.3.1. デフォルトゲートウェイの設定

(1) 「Network」→「Routes」で表示された画面の右上にある「Add」ボタンを押し、以下のように設定します。

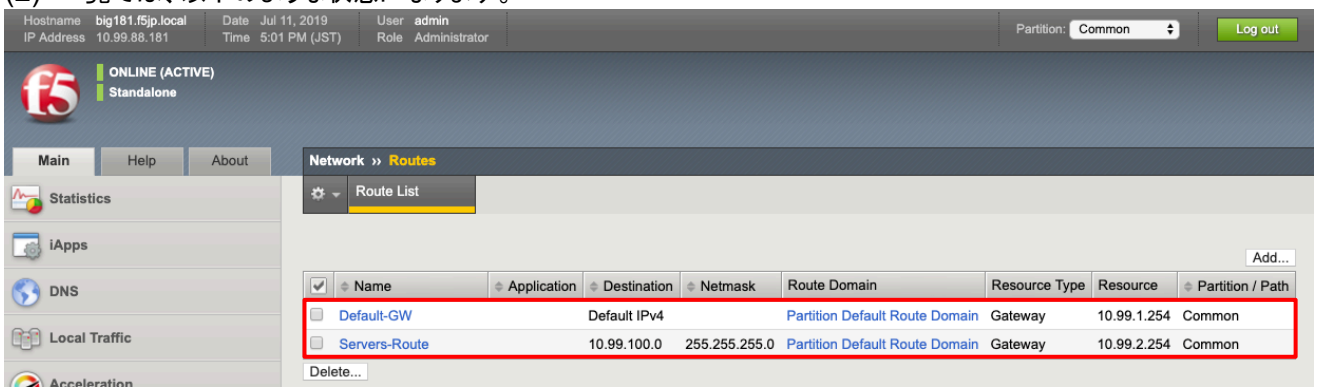


4.3.2. Web アプリケーションサーバへのルーティング設定

(1) Adv.WAF から Web アプリケーションサーバ：10.99.100.0/24 へ到達するためのルーティング設定も行います。



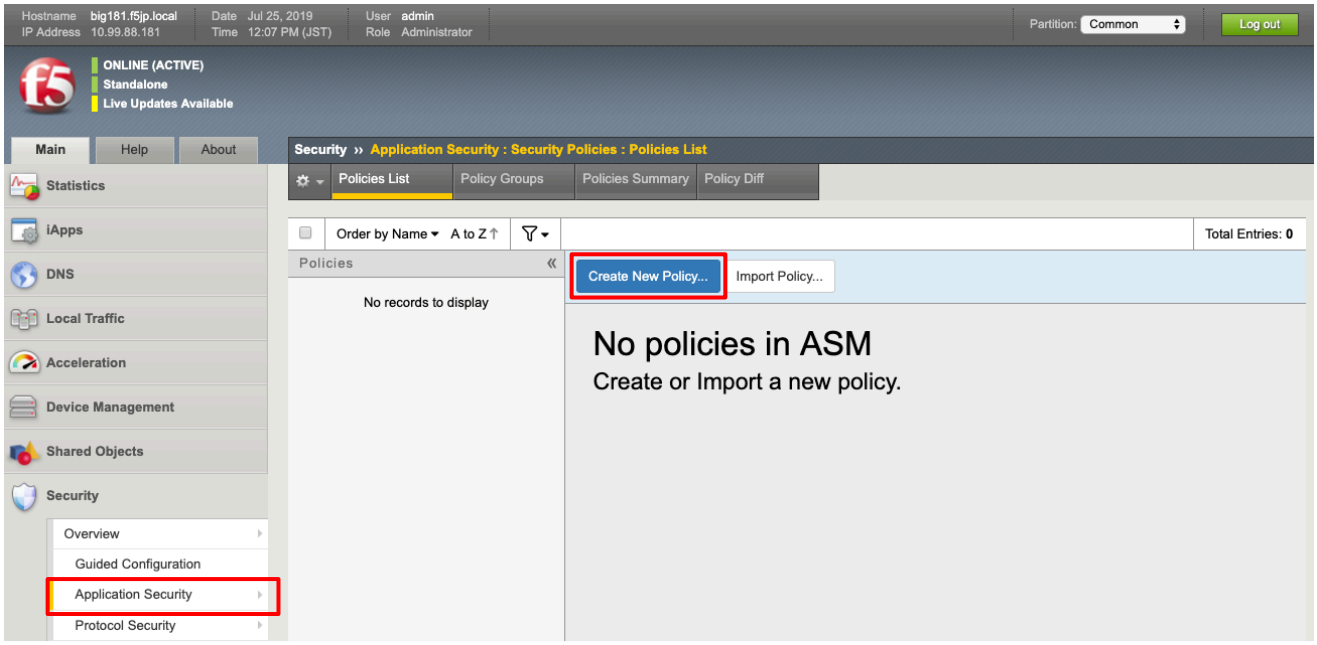
(2) 一覧では、以下のような状態になります。



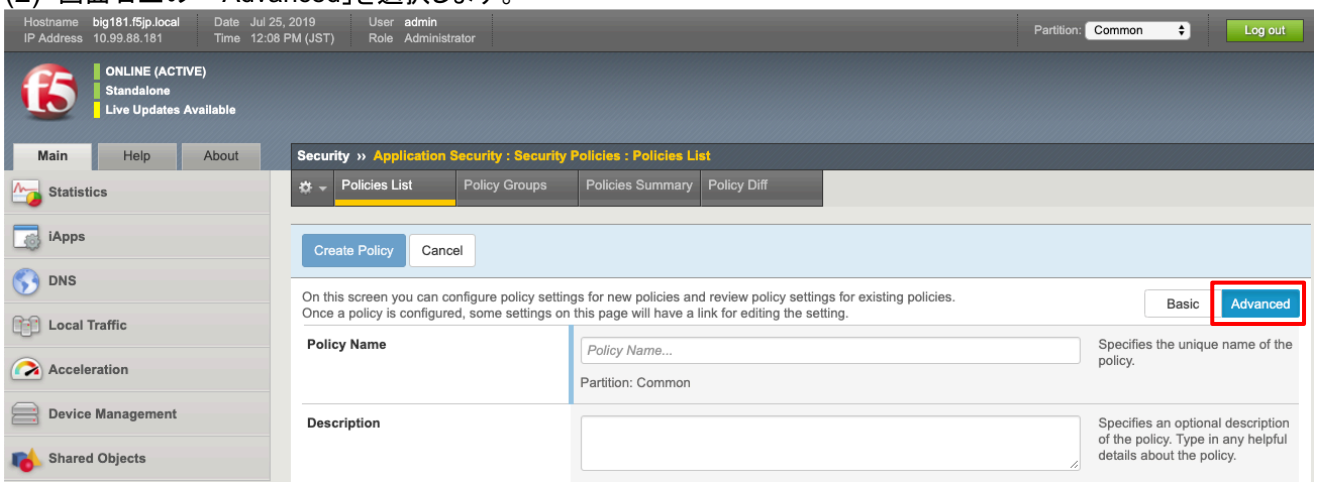
5. ［Step1］ベースとなる WAF ポリシーを作成する

5.1. Deployment Wizard による設定

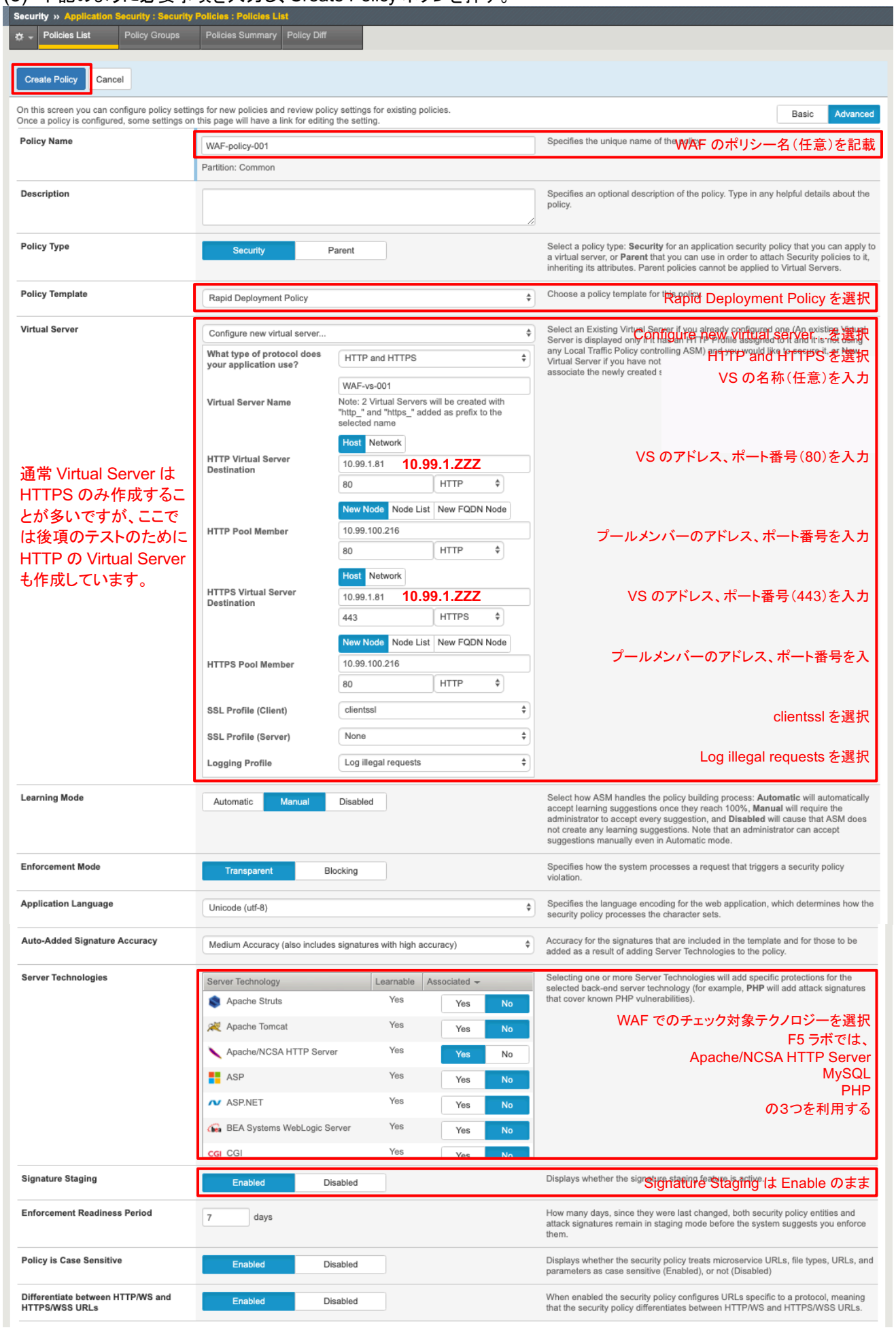
(1) 「Security」→「Application Security」→「Security Policies」で表示された画面中央の「Create New Policy…」ボタンを押します。



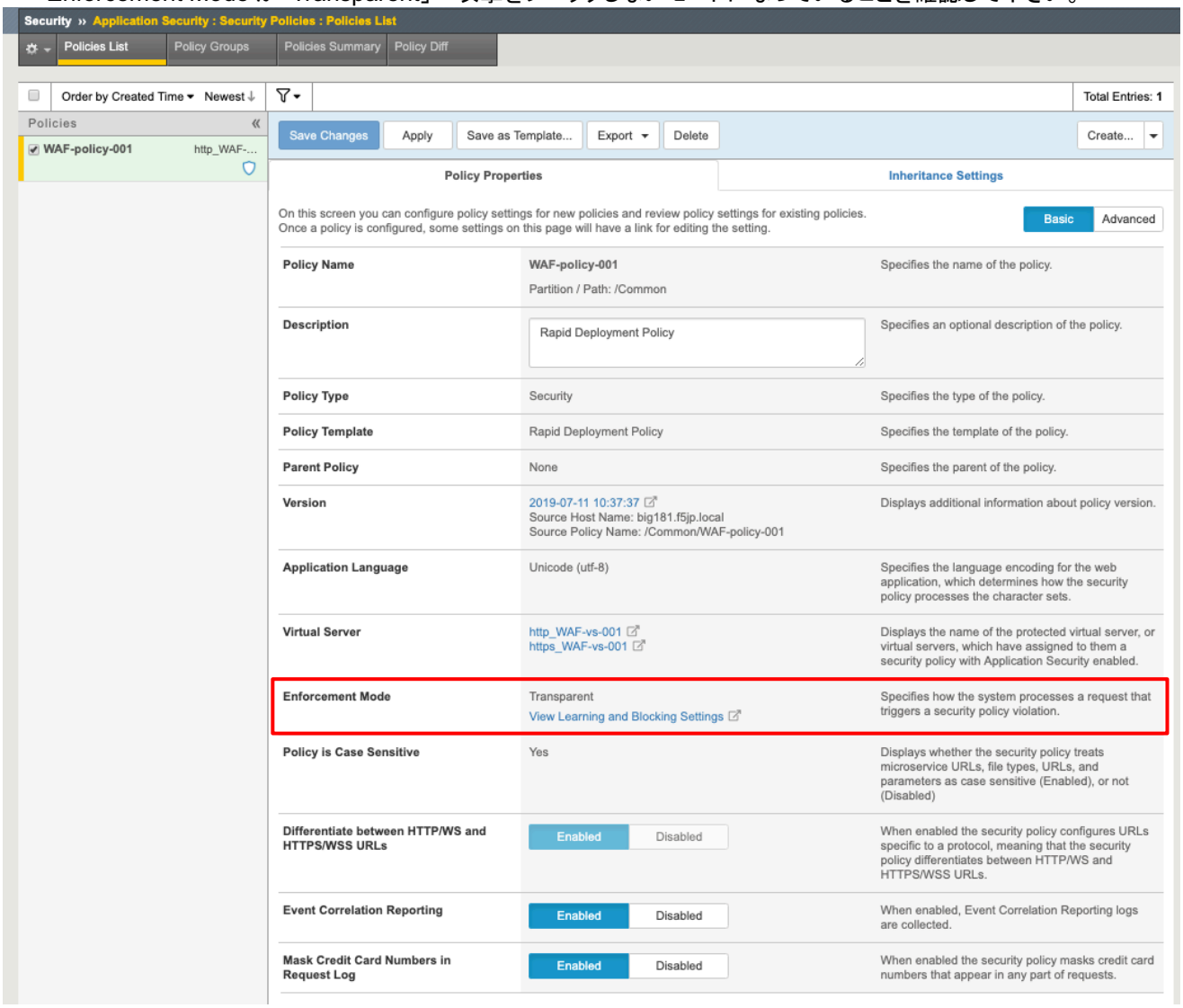
(2) 画面右上の 「Advanced」を選択します。



(3) 下記のように必要事項を入力し、Create Policy ボタンを押す。



(4) ポリシーが作成されると以下のように表示されます。
Enforcement Mode が「Transparent」＝攻撃をブロックしないモードになっていることを確認して下さい。

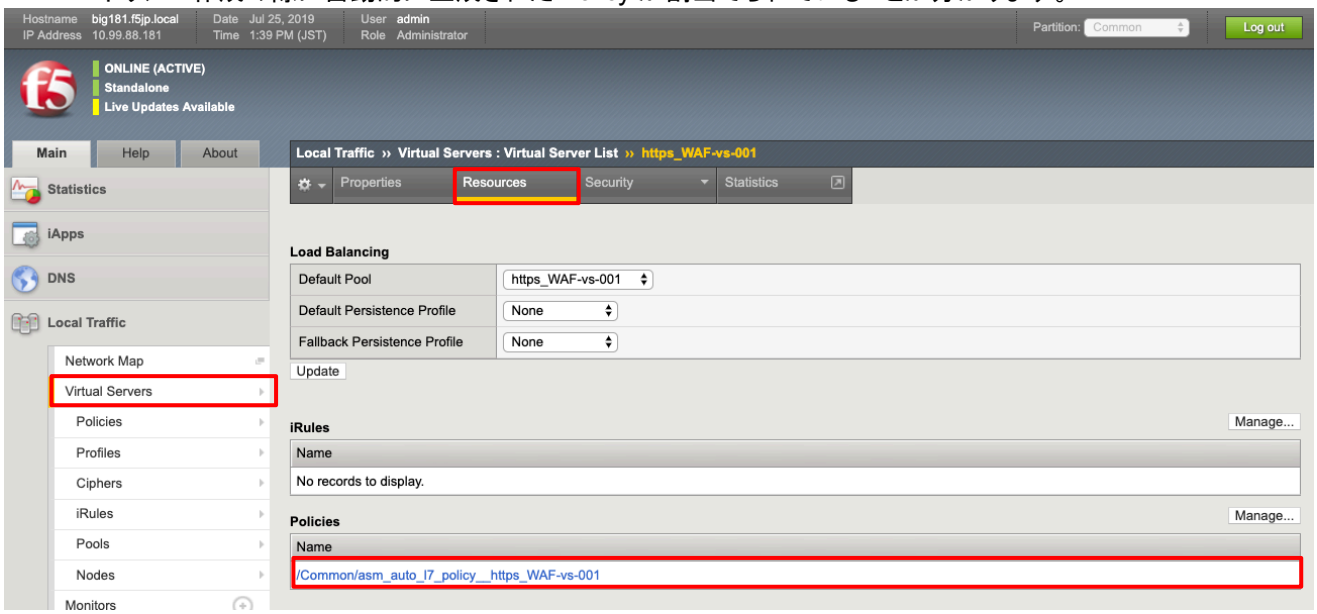


5.2. Virtual Server の設定の確認

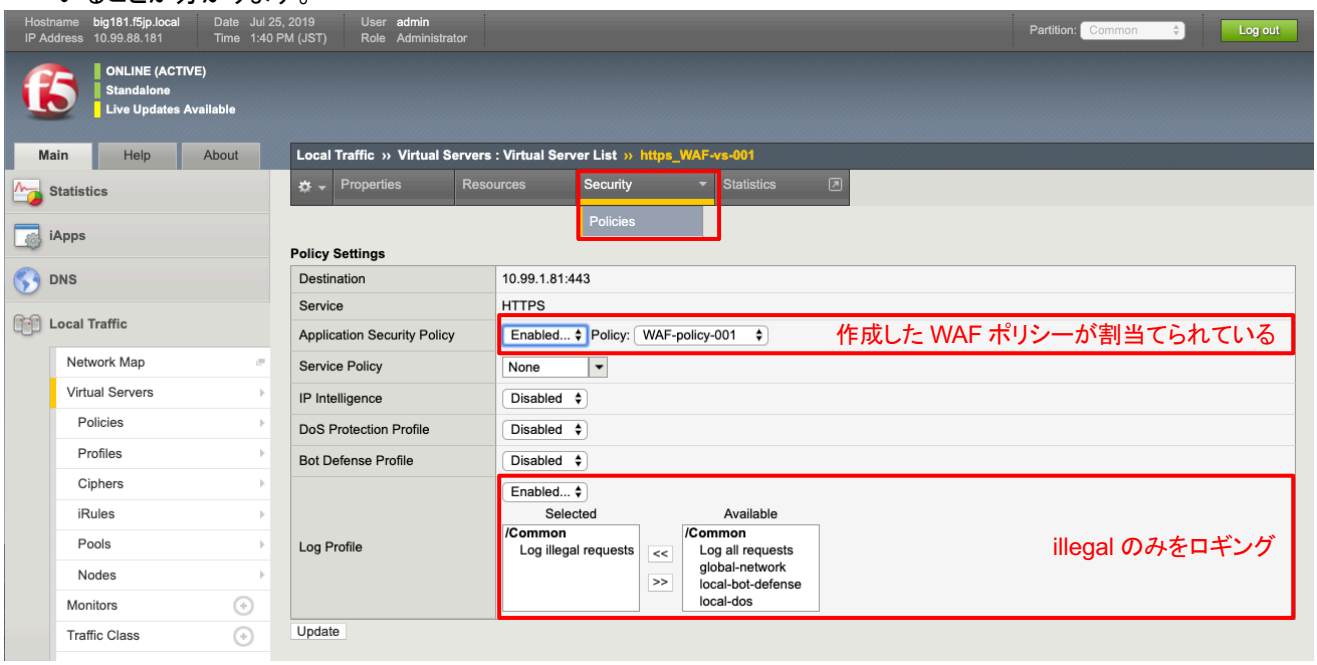
Deployment Wizard で設定した Virtual Server の設定を確認します。

5.2.1 Policies の確認

(1) 「Local Traffic」→「Virtual Servers」で表示された、作成済み https 用 Virtual Server を選択します。
その上に表示された「Resources」タブを選択します。
WAF ポリシー作成の際に自動的に生成された Policy が割当てられていることが分かります。



(2) 「Security」タブの「Policies」をクリックします。
生成した WAF ポリシーが割当てられていることが分かります。また、Log は、Log illegal requests が割当られていることが分かります。



(3) 作成済み http 用 Virtual Server も、(1)(2)と同様の内容を確認します。

5.3. Learning and Blocking Setting の確認と変更

Adv.WAF のセキュリティ機能の ON/OFF のほとんどは、Learning and Blocking Setting で設定します。

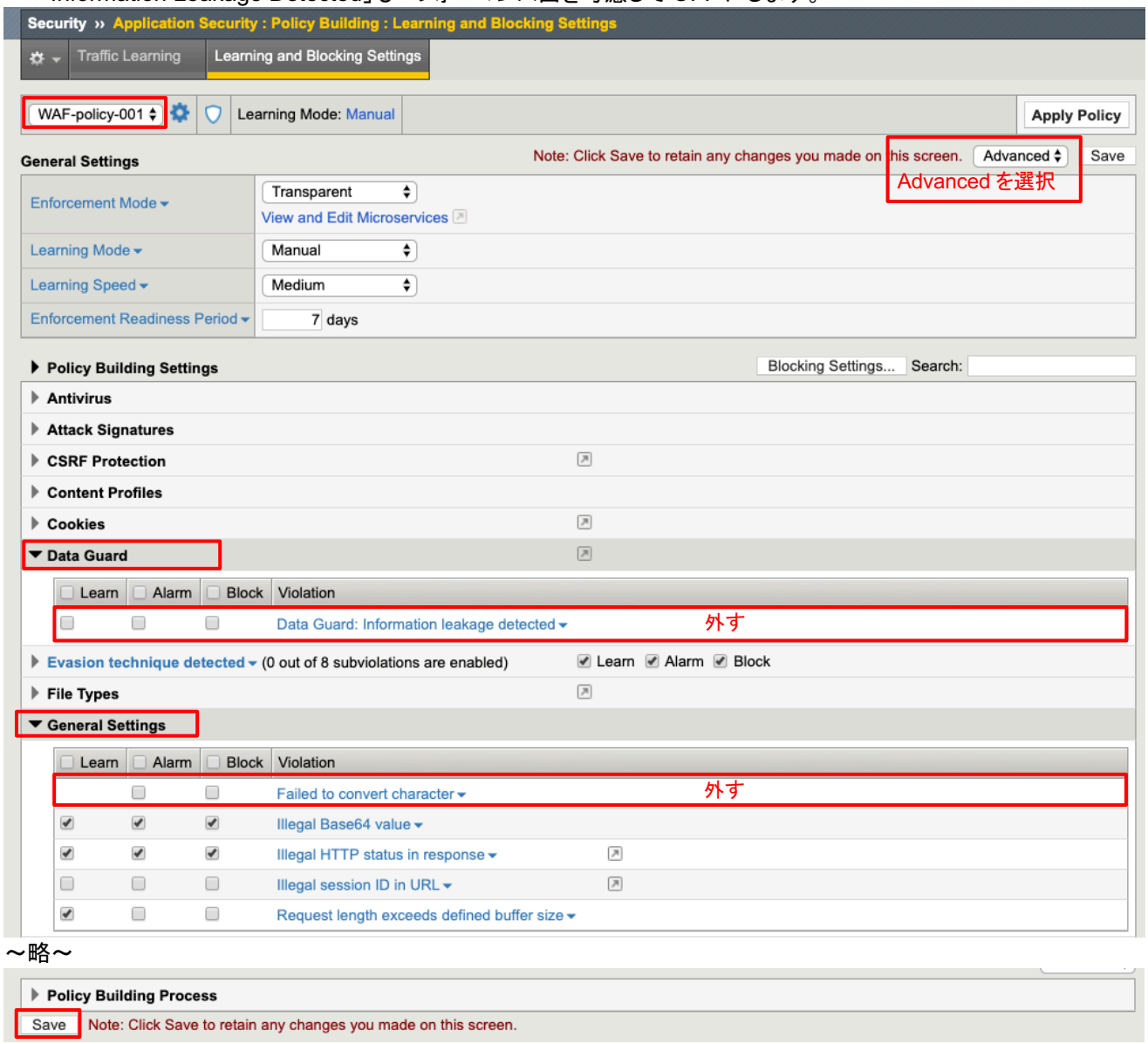
「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面を確認します。

Rapid Deployment Security Policy によって、いくつかの Violations の「Learn」と「Alarm」が有効になっていることを確認します。Enforcement Mode が Transparent なので、「Block」はすべてグレーアウトされています。

- 「Learn」・・・・ Manual Traffic Learning 画面で、検知した攻撃を学習するモード
- 「Alarm」・・・・ Event Logs 画面で、攻撃を検知したことを示すログを出力するモード
- 「Block」・・・・ 攻撃をブロックするモード

5.3.1. Blocking Setting の設定変更

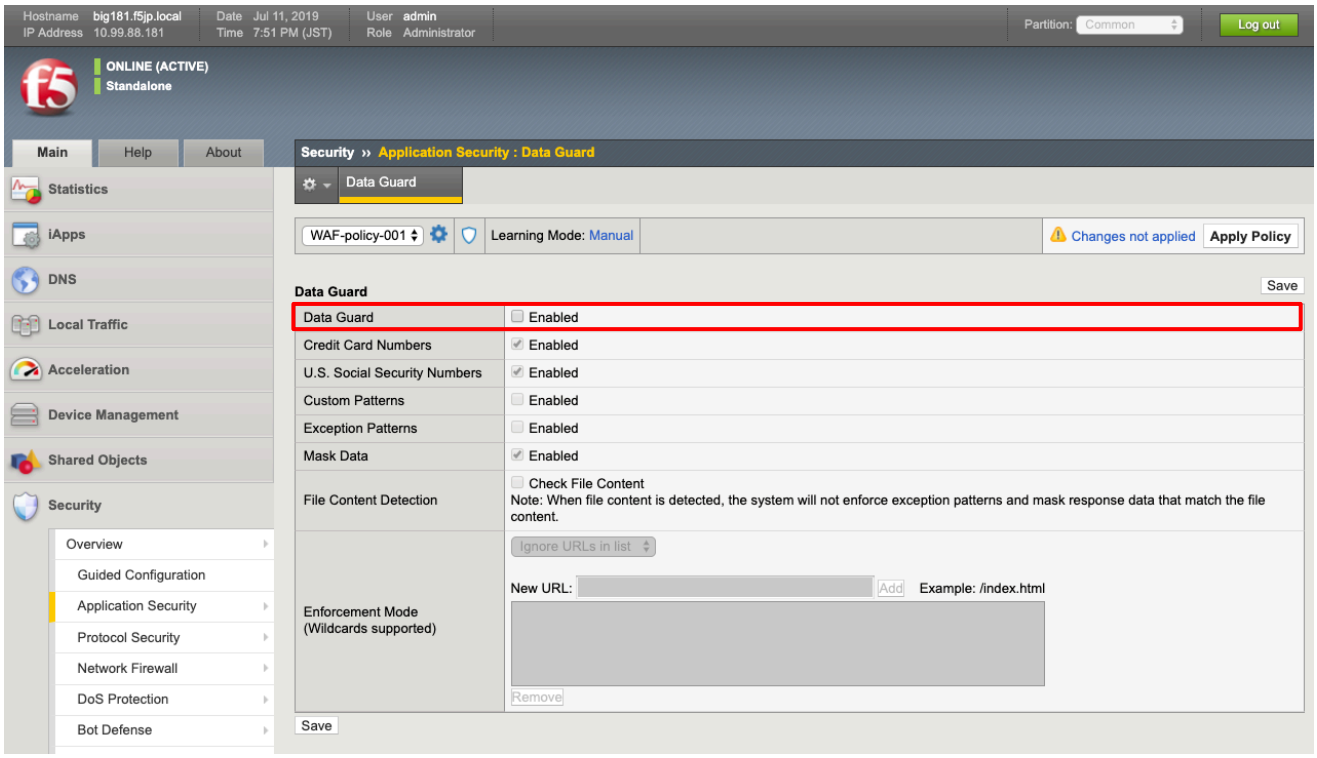
(1) 日本語サイトの誤検知の防止策として、「Failed to convert character」を OFF にします。また、「Data Guard: Information Leakage Detected」もパフォーマンス面を考慮して OFF にします。



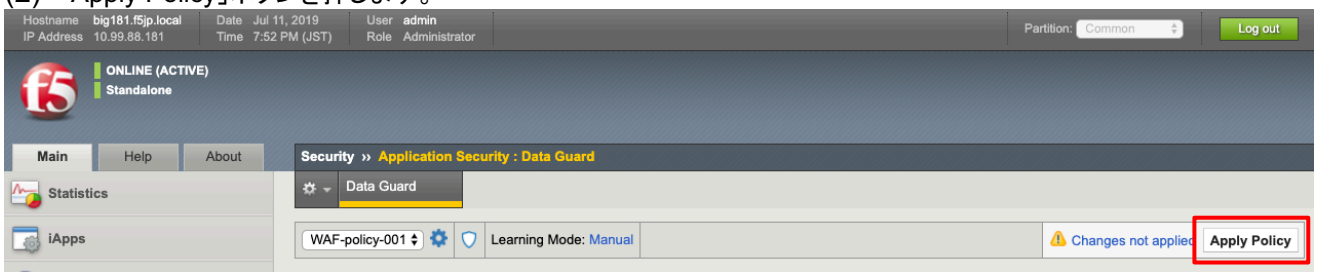
(2) 変更後、「Save」ボタンを押します。

5.3.2. Data Guard の無効化-2

(1) 「Security」→「Application Security」→「Data Guard」で表示された画面で、「Data Guard」のチェックが外れていることを確認します。



(2) 「Apply Policy」ボタンを押します。



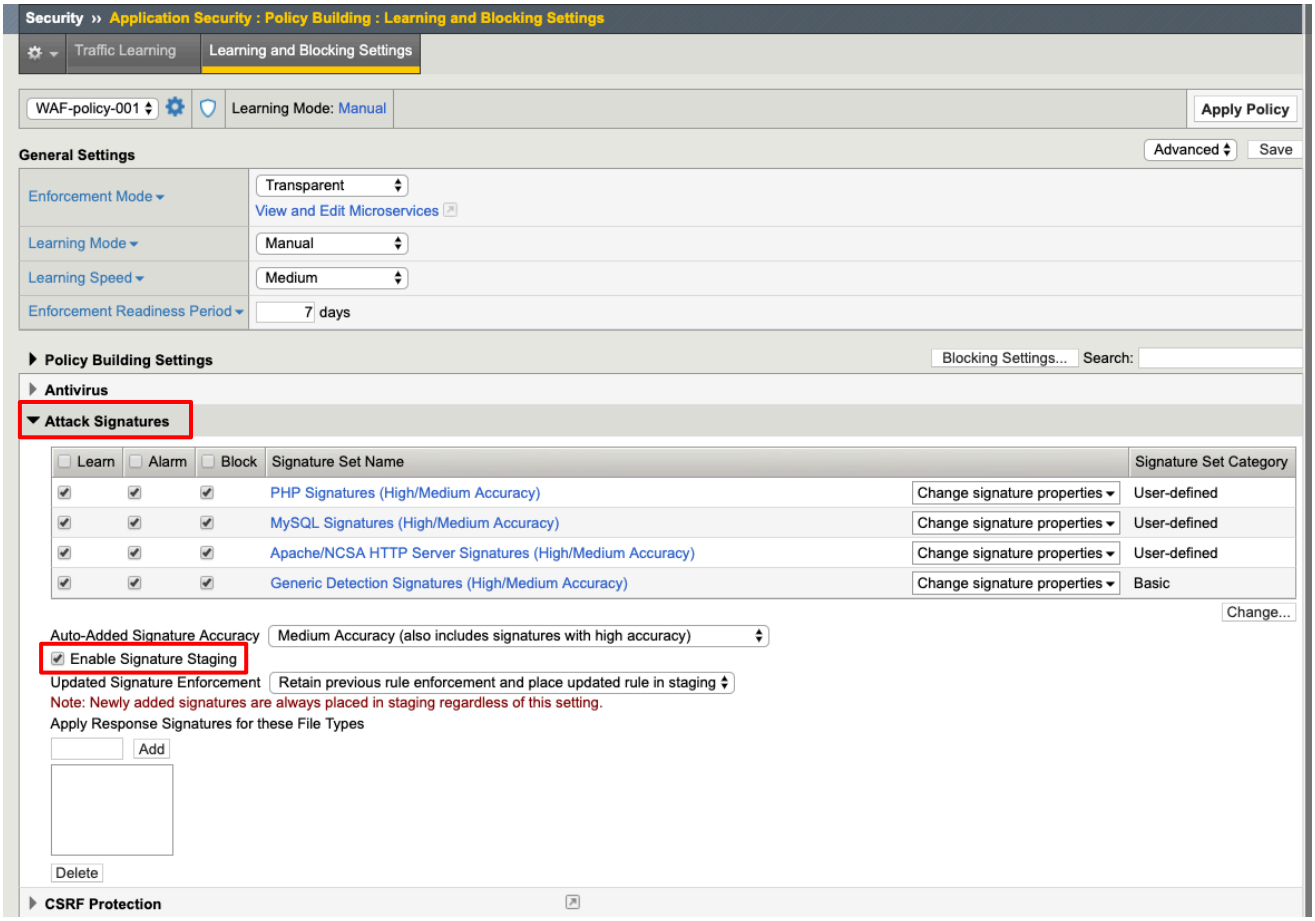
Attack Signature の状態確認及び変更は次の章で行います。

5.4. シグネチャの状態確認

5.4.1. Attack Signature Configuration

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」の「Attack Signatures」を選択します。

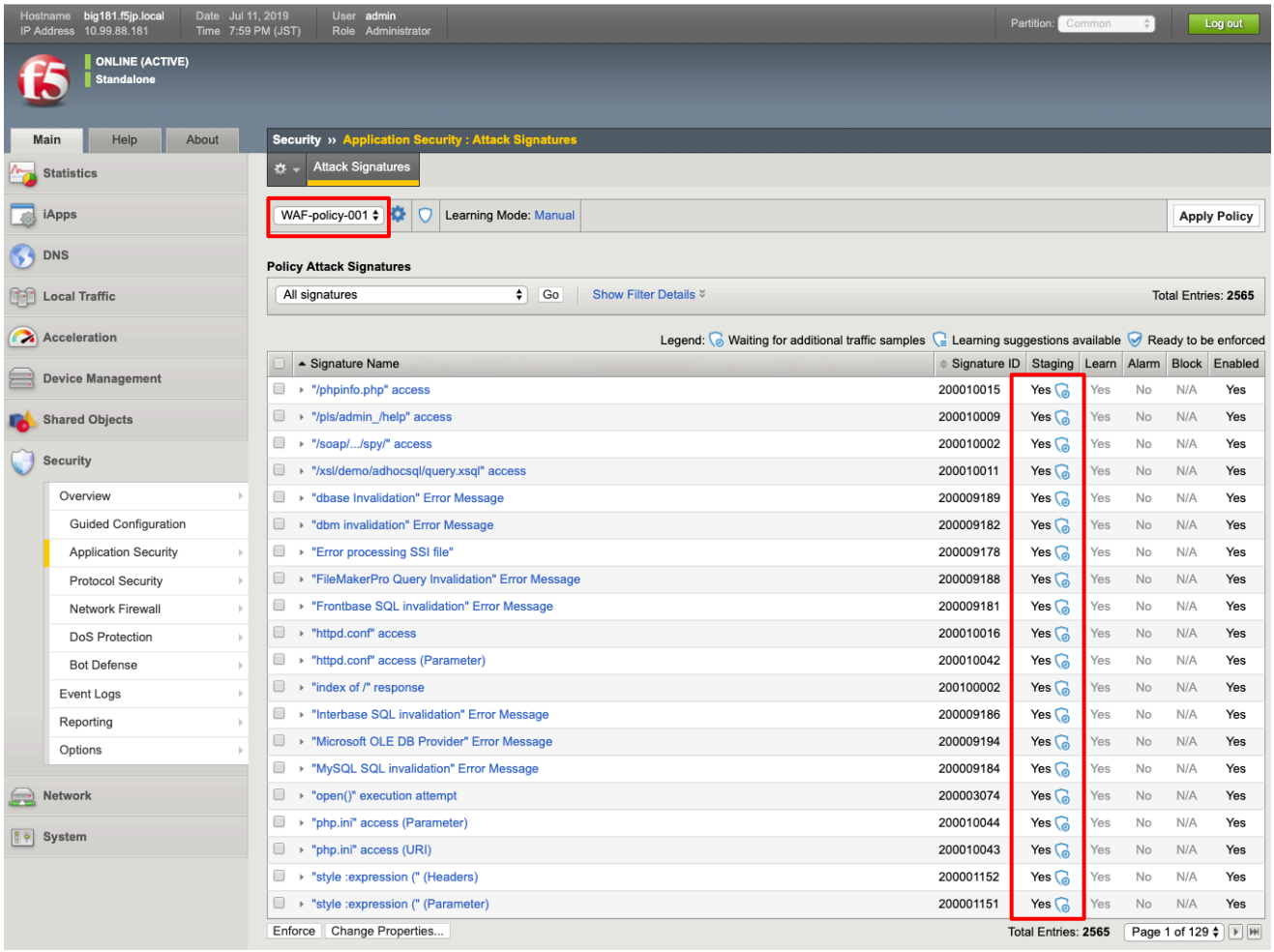
「Enable Signature Staging」にチェックが入っていることを確認します。



5.4.2. Attack signature List

5.4.2.1. 各シグネチャ状態の確認

(1) 「Security」→「Application Security」→「Attack Signatures」をクリックします。
Signature Staging が Enabled なので、Learn/Alarm/Block が無効になり、「In Staging」の状態になっています。



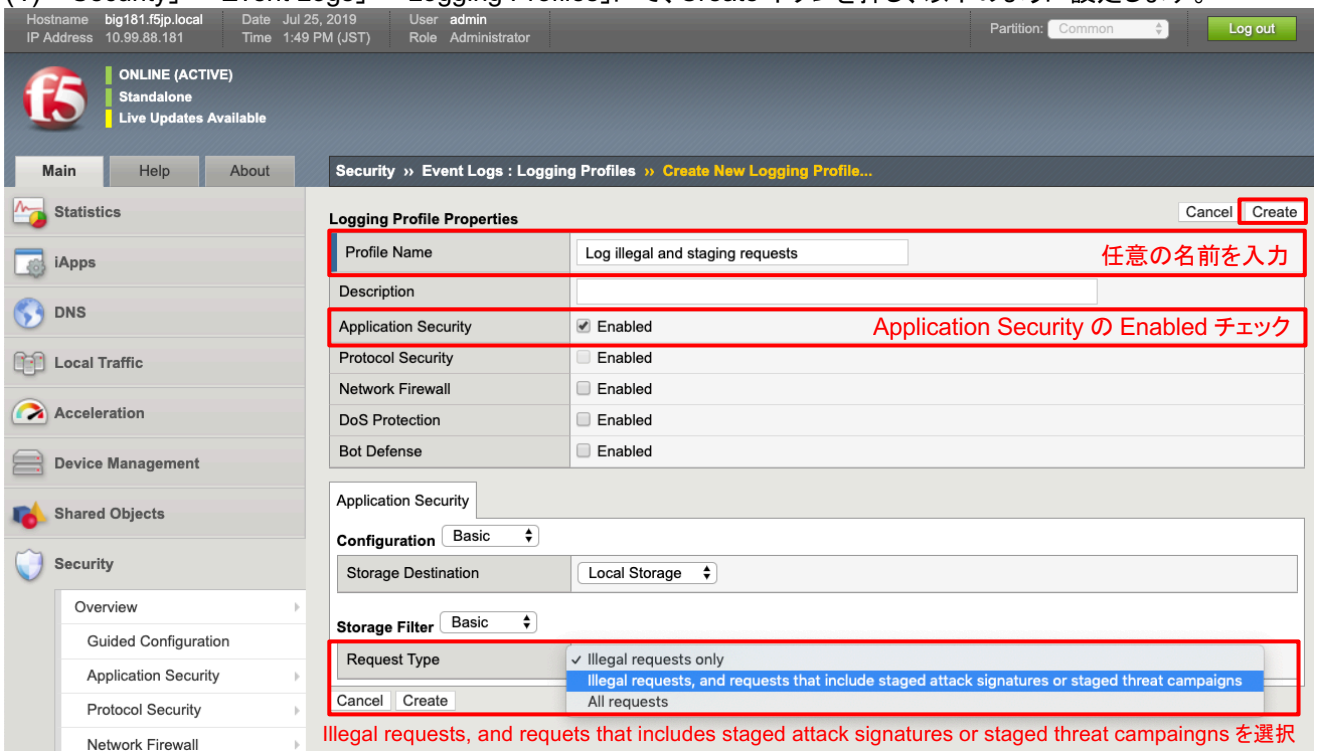
- Staging（ステージング）モードとは：

  Learn/Alarm/Block 設定が無効化され、攻撃を検知した場合には、「Manual Traffic Learning」で学習するだけの動作となるモードです。

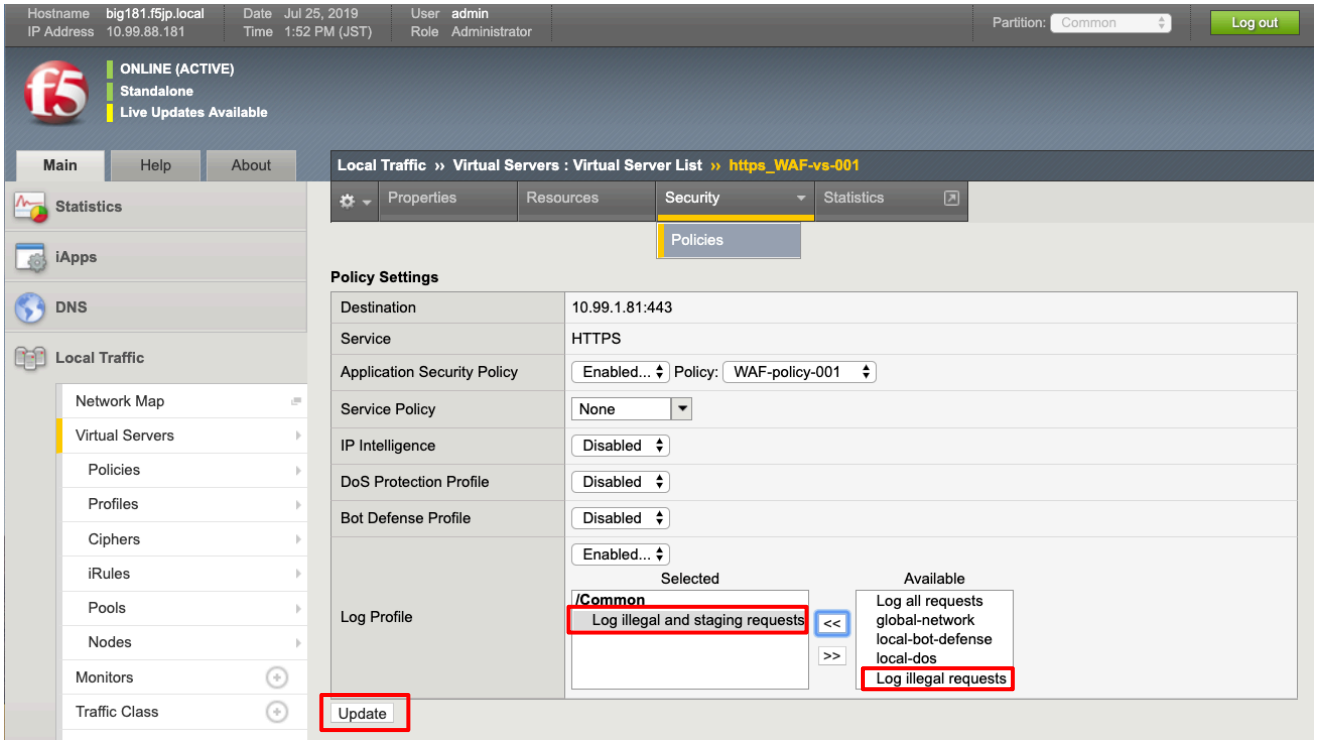
5.4.3. ステージングログの設定

ステージングのログを Event Logs に出力するための設定を行います。

(1) 「Security」→「Event Logs」→「Logging Profiles」にて、Create ボタンを押し、以下のように設定します。



(2) Virtual Server に(1)で作成したロギングプロファイルを追加して、既存のプロファイルをはずします。Update ボタンを押します。



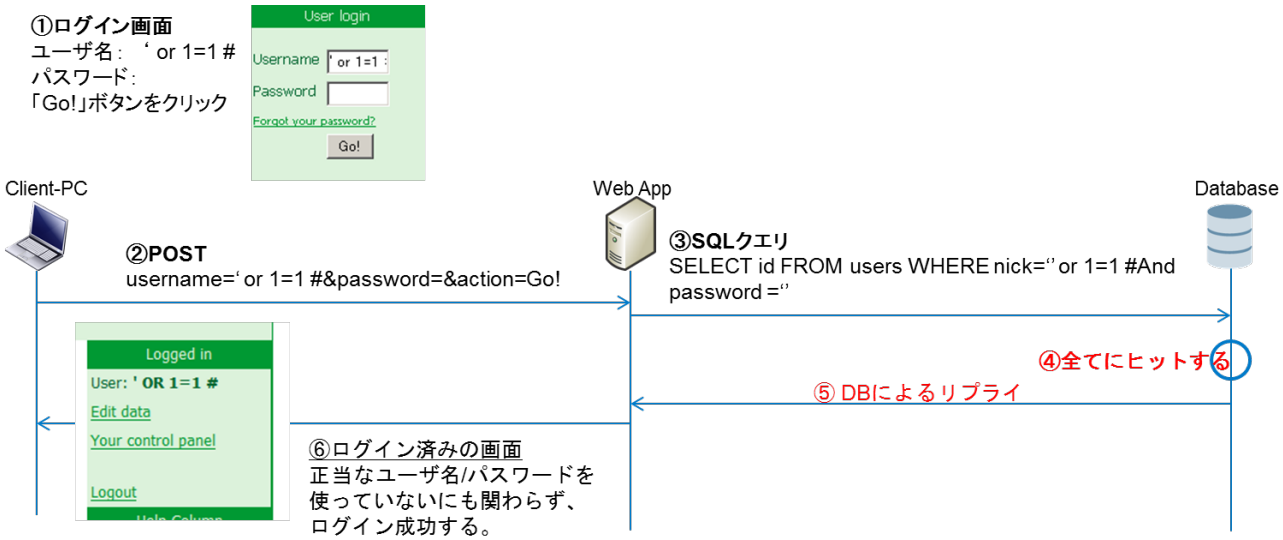
このステージングモードのまま、次項「シグネチャ動作の確認」を実施します。

6. シグネチャ動作の確認

6.1. 例：SQL インジェクション

シグネチャの動作を確認するにあたり、SQL インジェクション攻撃を例に取ります。

SQL インジェクションは、以下のような攻撃です。



① ユーザ名に「’ or 1=1 #」と入力して「Go!」ボタンを押す。
② 結果、HTTP リクエスト：POST で、「username=‘ or 1=1 #&password=&action=Go!」を送る。
③ Web アプリケーションは、

SELECT id FROM users WHERE nick=‘username 値’ And password =‘password 値’

の SQL 文を生成し、データベースへ送る準備をしている。

ここに、HTTP リクエストで到着した値を埋め込むと、

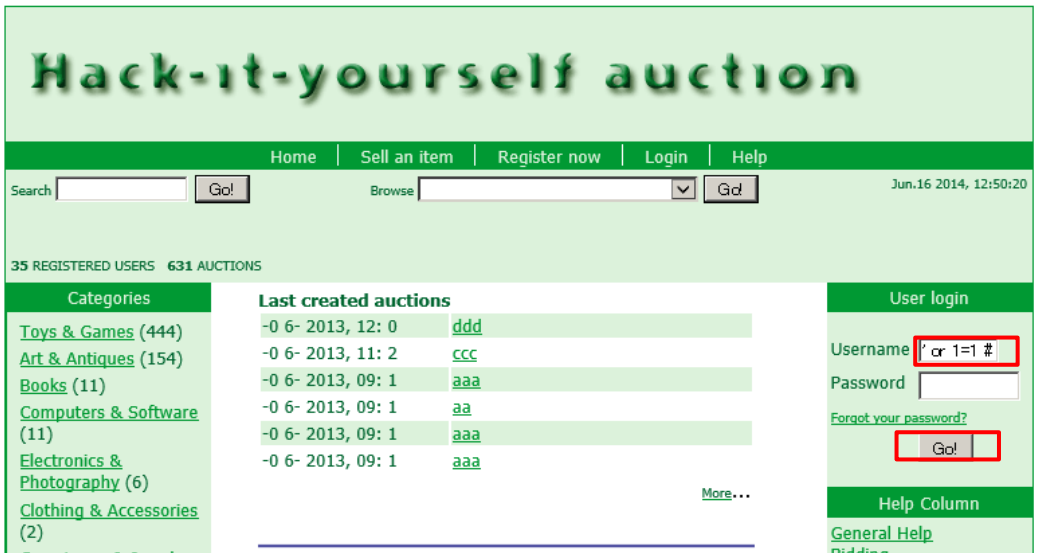
SELECT id FROM users WHERE nick=‘’ or 1=1 #‘ And password =‘’

の形になる。

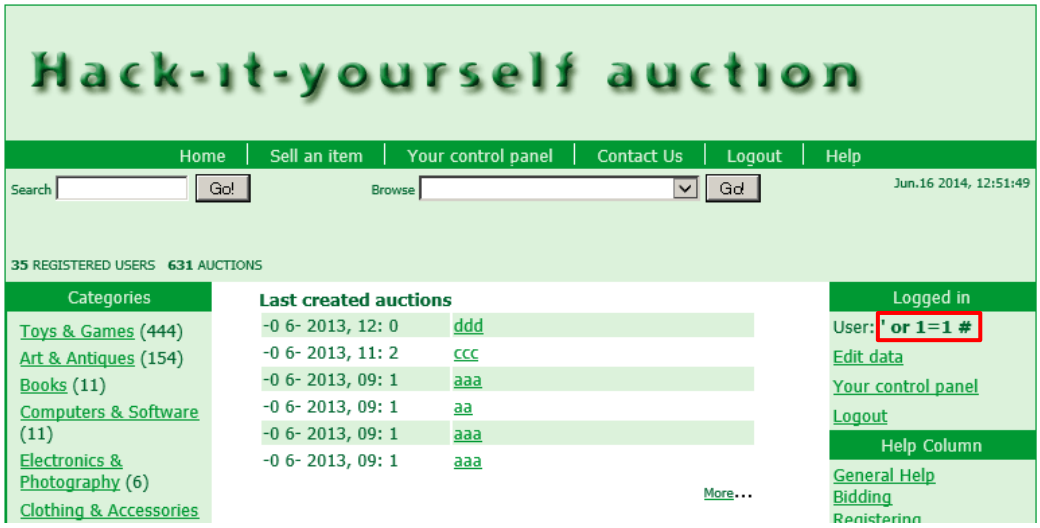
これの意味するところは、
「nick が空 または 1=1 ならば、users テーブルから id を取得する。#以降はコメントアウト。」
となる。
④ この SELECT 文内の”1=1”は正しいので、データベースのすべてにヒットする。
⑤ データベースからリプライを受ける。
⑥ ログインに成功する。

Adv.WAF は、このような脆弱性を持つ Web アプリケーションを防御するためのシグネチャ群を持っています。
この攻撃を例に、シグネチャの動作を確認していきます。

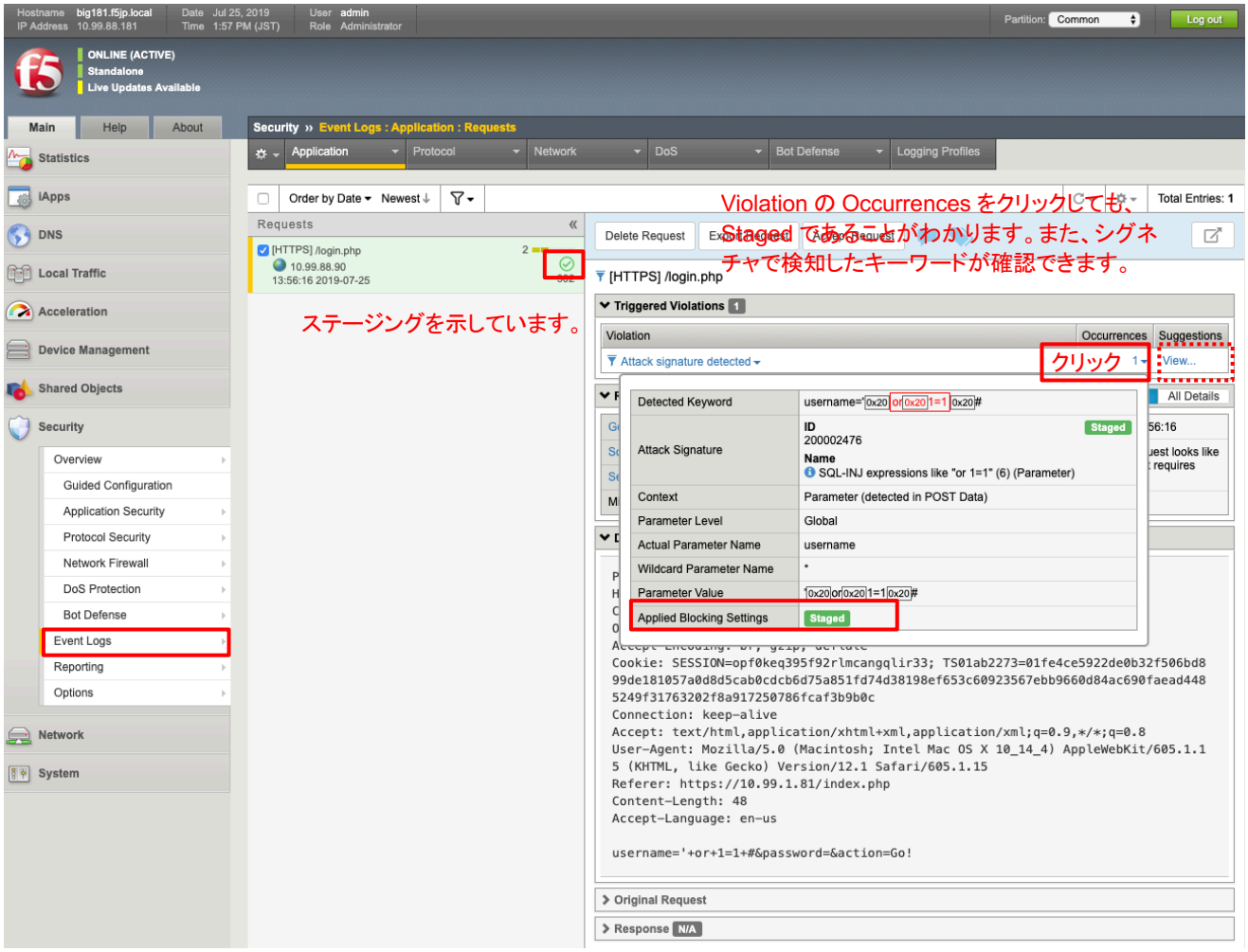
(1) クライアント PC から、https://10.99.1.ZZZ へアクセスします。
ブラウザの Username 欄に、「' or 1=1 #」を入力し、「Go!」ボタンを押します。



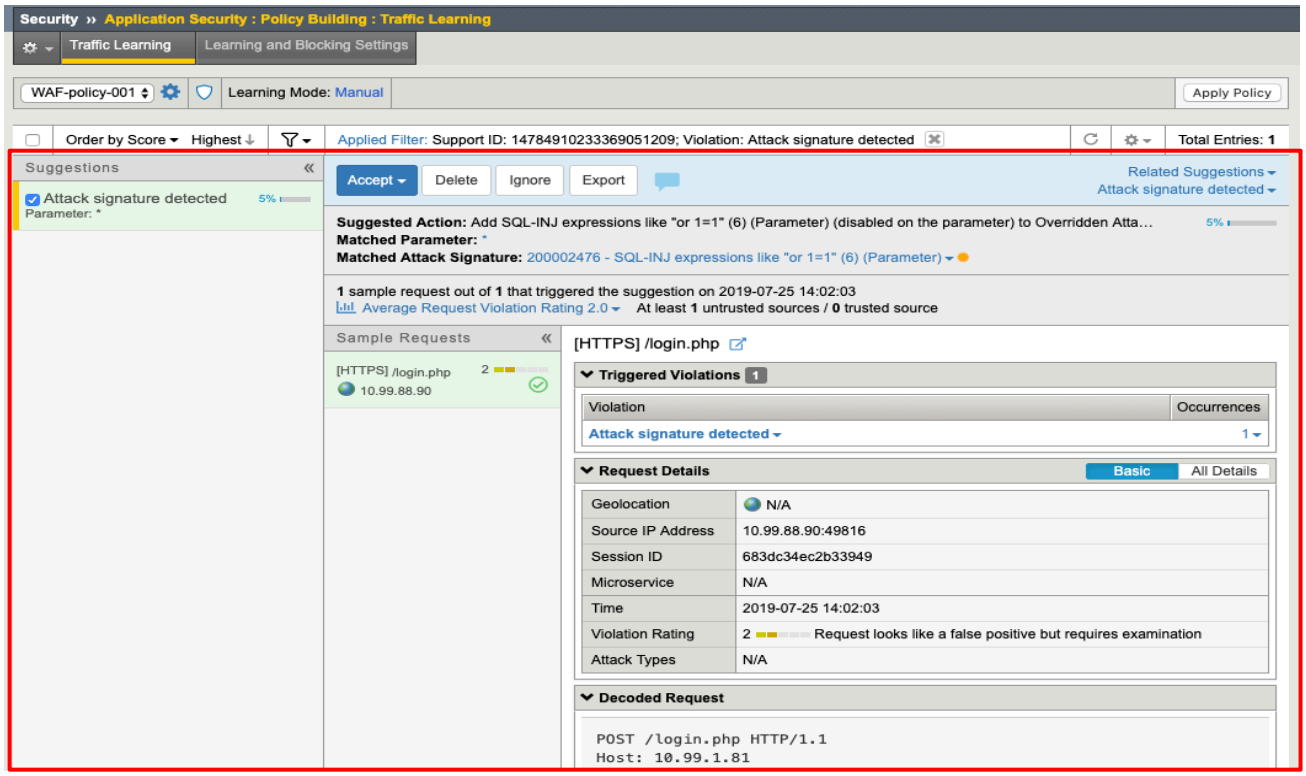
(2) 本 Web アプリケーションはこの脆弱性を持つため、ログインできてしまいます。



(3) 「Security」→「Event Logs」→「Application」→「Requets」で表示された画面で、”Staged”で SQL インジェクションが検出されていることを確認します。



(4) 上記画面の Suggestions の View…をクリックすると、Traffic Learning の画面でも Attack signature detected が確認できます。（「Security」→「Application Security」→「Policy Building」→「Traffic Learning」でもたどれます。）

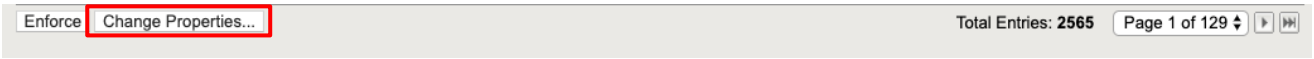


6.2. 各シグネチャの Staging の解除

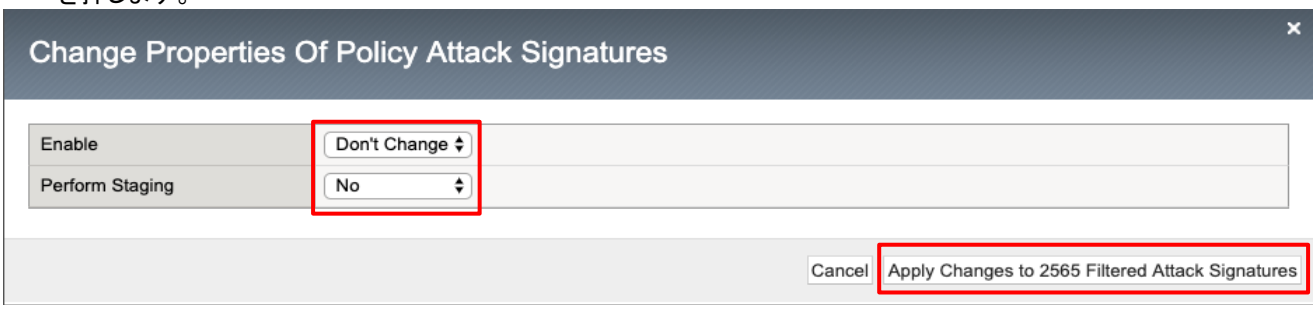
ステージング状態のままでは、攻撃を Event Log として記録してくれません。
よって、全シグネチャのステージングを解除（＝Learn/Alarm/Block 設定を有効に）します。

(1)

(2) 「Security」→「Application Security」→「Attack Signatures」で表示された画面下の「Change Properties」ボタンを押します。

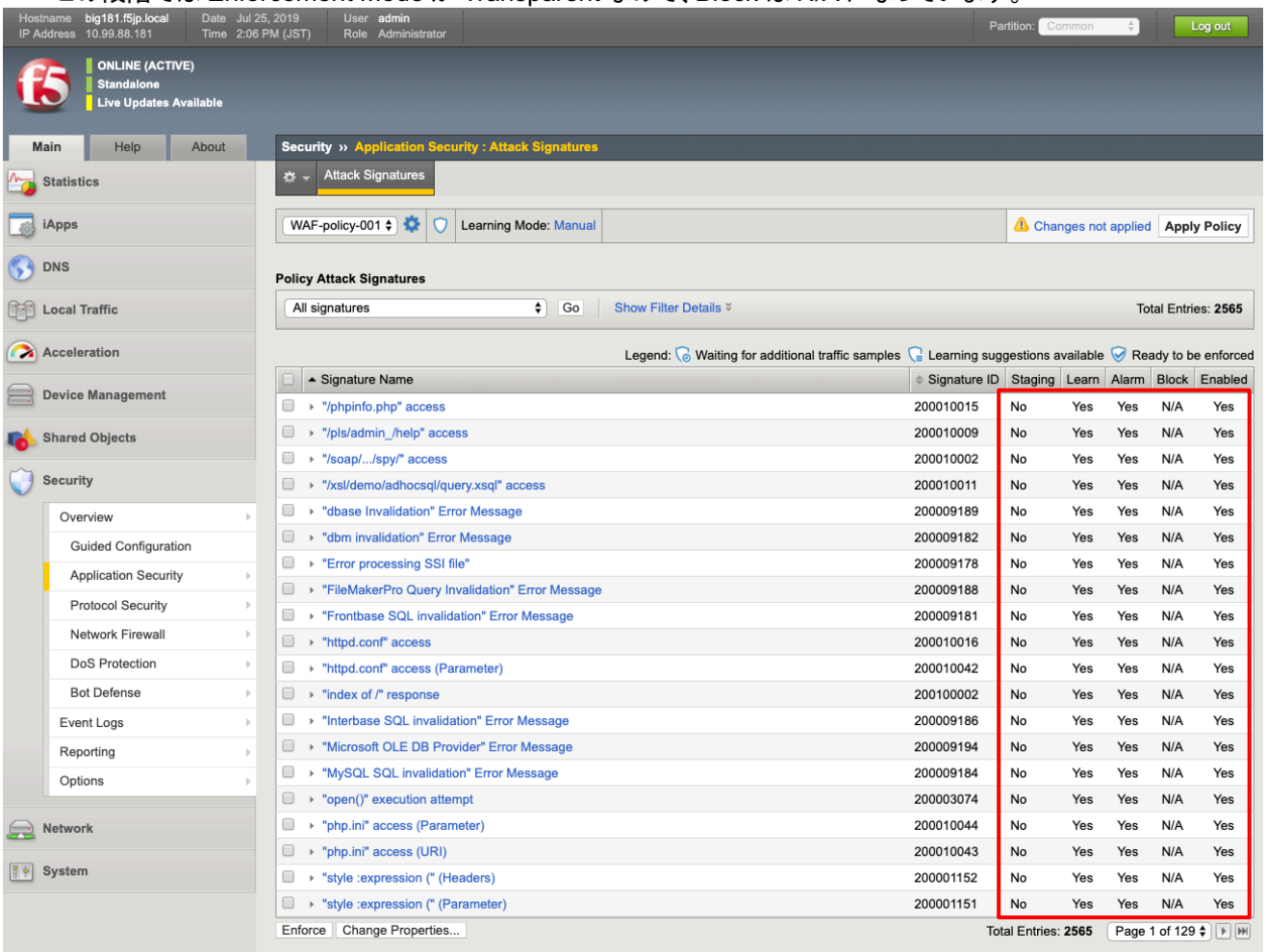


(3) 上図の「Change Properties」ボタンを押すと現れる画面で、以下のように設定し、「Apply Changes to…」ボタンを押します。

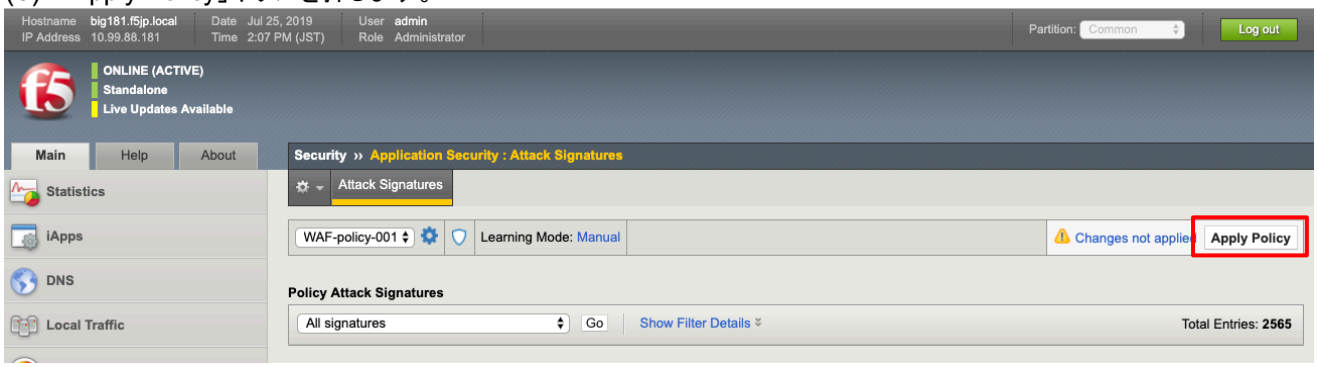


(4) ステージングモード解除が完了するまで、少し時間がかかります。

(5) 以下のイメージは、Staging モードが解除された状態です。
この段階では Enforcement Mode が Transparent なので、Block は N/A になっています。



(6) 「Apply Policy」ボタンを押します。



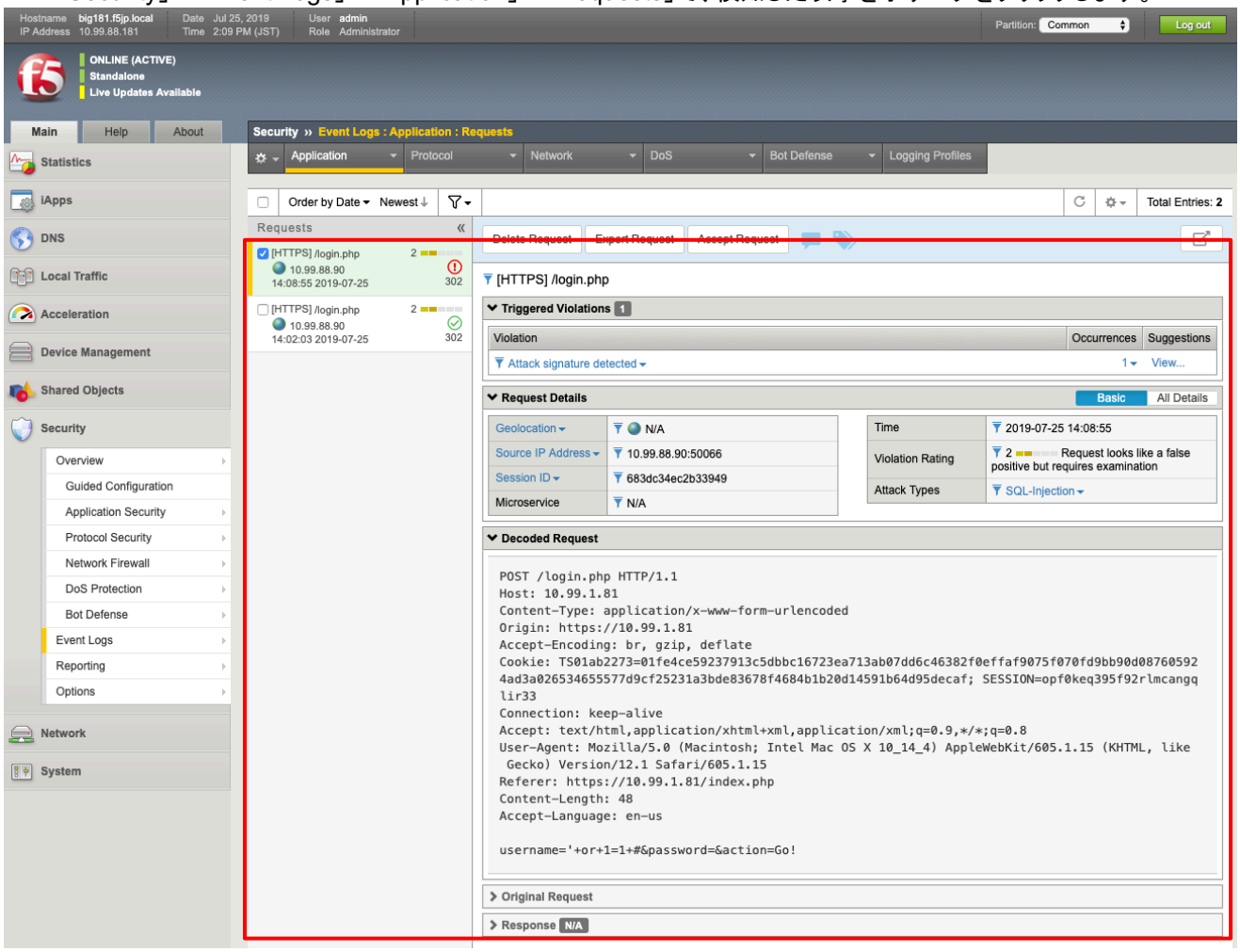
[Note]

「Policy Building Settings」-「Attack Signature Configuration」の「Enable Signature Staging」を OFF にすることでも、全シグネチャの Staging を解除することはできます。しかし、その方法ですと問題になる場合があります。例えば、シグネチャの Update が行われた際には、その Update されたシグネチャだけはしばらく Staging で運用したい場合があります。この場合、この「Signature Staging」が Enabled でなければ、そのように動作してくれません。つまり、この「Attack Signature Configuration」の「Signature Staging」を OFF することで、「Staging 機能は全く使わない」という設定になる、ということです。よって、本ガイドでは、「Enable Signature Staging」は ON にしておき、各シグネチャの Staging を OFF にする、という方法を取ります。

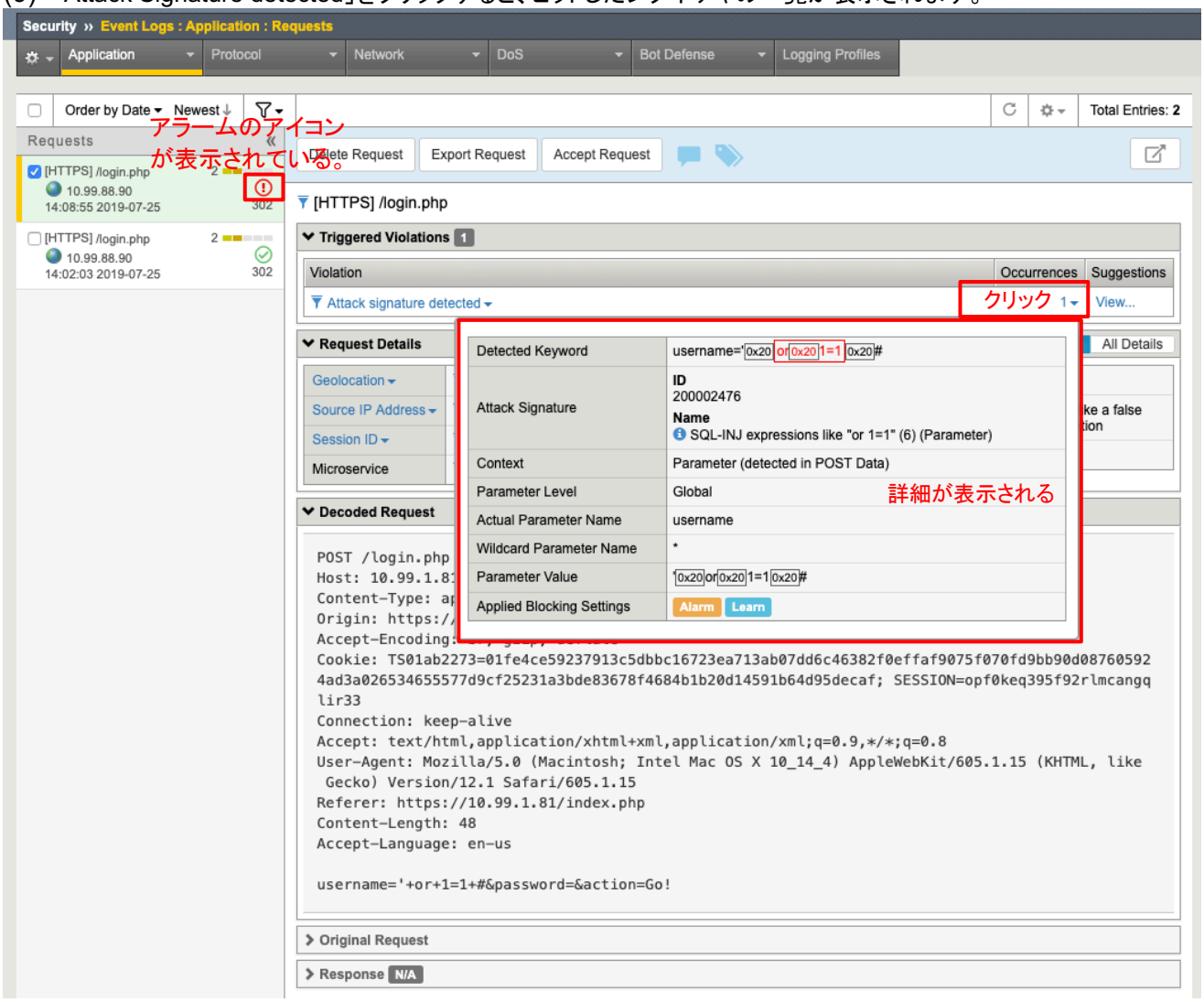
尚、更新されたシグネチャだけをステージングモードとするための設定は、「シグネチャの運用」のセクションで解説します。

(7) もう一度、クライアントのブラウザから、同様の SQL インジェクション攻撃を実施します。

(8) Signature の Alarm が有効になっているので、攻撃を検知したことを示すログが出力されます。
「Security」→「Event Logs」→「Application」→「Requests」で、検知した攻撃を示すログをクリックします。



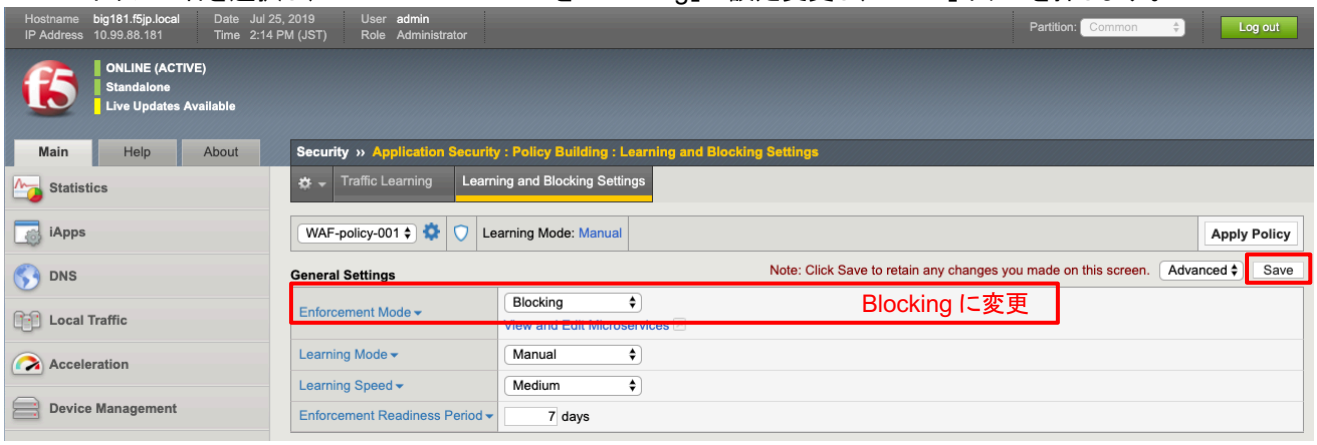
(9) 「Attack Signature detected」をクリックすると、ヒットしたシグネチャの一覧が表示されます。



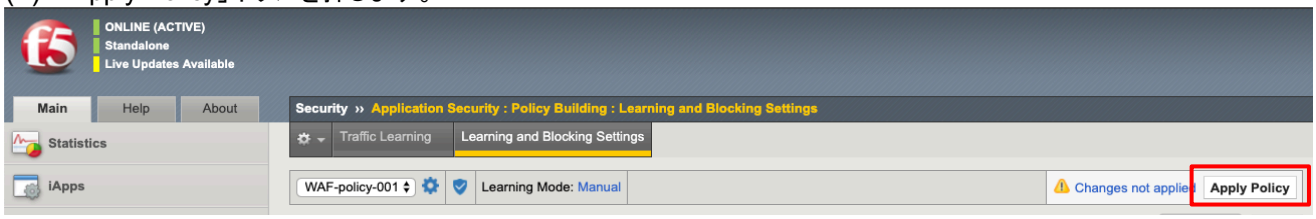
6.3. Enforcement Mode を Blocking に変更

攻撃をブロックする設定方法を示します。

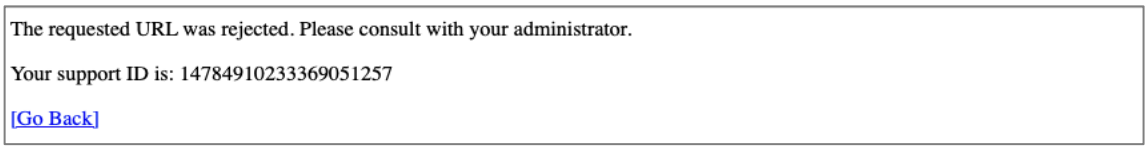
(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」において、該当のポリシー名を選択し、Enforcement Mode を「Blocking」に設定変更し、「Save」ボタンを押します。



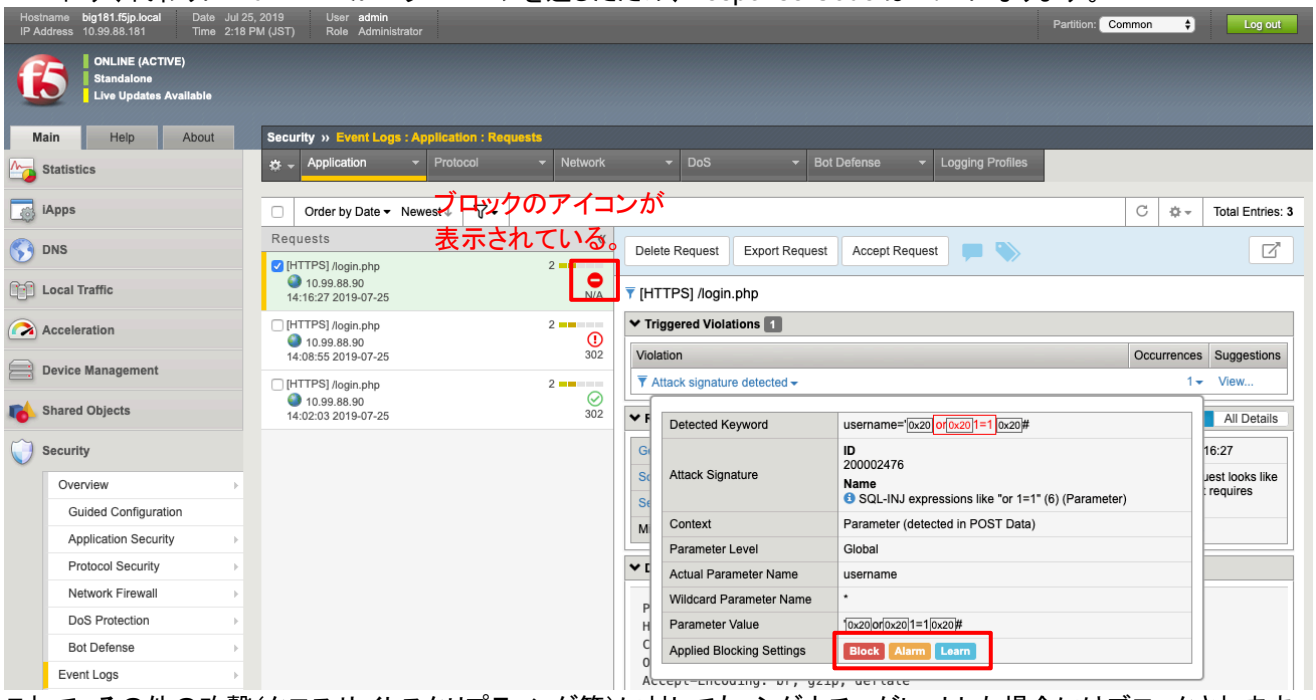
(2) 「Apply Policy」ボタンを押します。



(3) クライアント PC のブラウザからもう一度、同様の SQL インジェクション攻撃を実施します。
Block されたことを示す画面が表示されます。



(4) Event Logs を確認します。Block されているので、Pool Member の Web サーバからの HTTP レスポンスは返されず、代わりに Adv.WAF がエラーページを返したため、Response Code は N/A になります。



これで、その他の攻撃（クロスサイトスクリプティング等）に対しても、シグネチャがヒットした場合にはブロックされます。

7. ［Step2］シグネチャのチューニング（誤検知の対策）

このステップでは、誤検知が発生した場合の対処方法を示します。

このステップで実施する内容は、Web アプリケーションの各パラメータの役割が全て把握できている場合を除き、運用開始前から全てを設定するのは難しいかもしれません。
その場合は、仮運用・本運用に入ってから、再度このチューニングを実施してください。

7.1. 誤検知の例

例えば、BIG-IP をオークションで売る、という場合を想定します。

Web アプリケーションに「f5userX」でログインし、上段の「Sell an item」をクリックしてください。

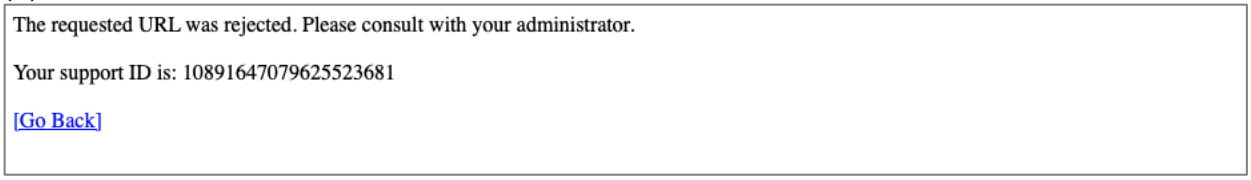
Item の説明に、代表的な例として SQL インジェクション攻撃を防御する機能があることを説明することにし、具体例として「' OR 1=1 #」を用いて説明したとします。

これは攻撃ではなく、ただ単に SQL インジェクションを説明したいだけなのですが、[Step1]で行った、ただシグネチャを適用しただけの状態では、このフォームも対象となるため、攻撃とみなされます。

(1) 例）BIG-IP をオークションで売る例

（アプリケーションの作り上、その他にも必須とされている値があります。
それらについては、本ガイドの手順には直接影響しないので、任意で入力してください。）

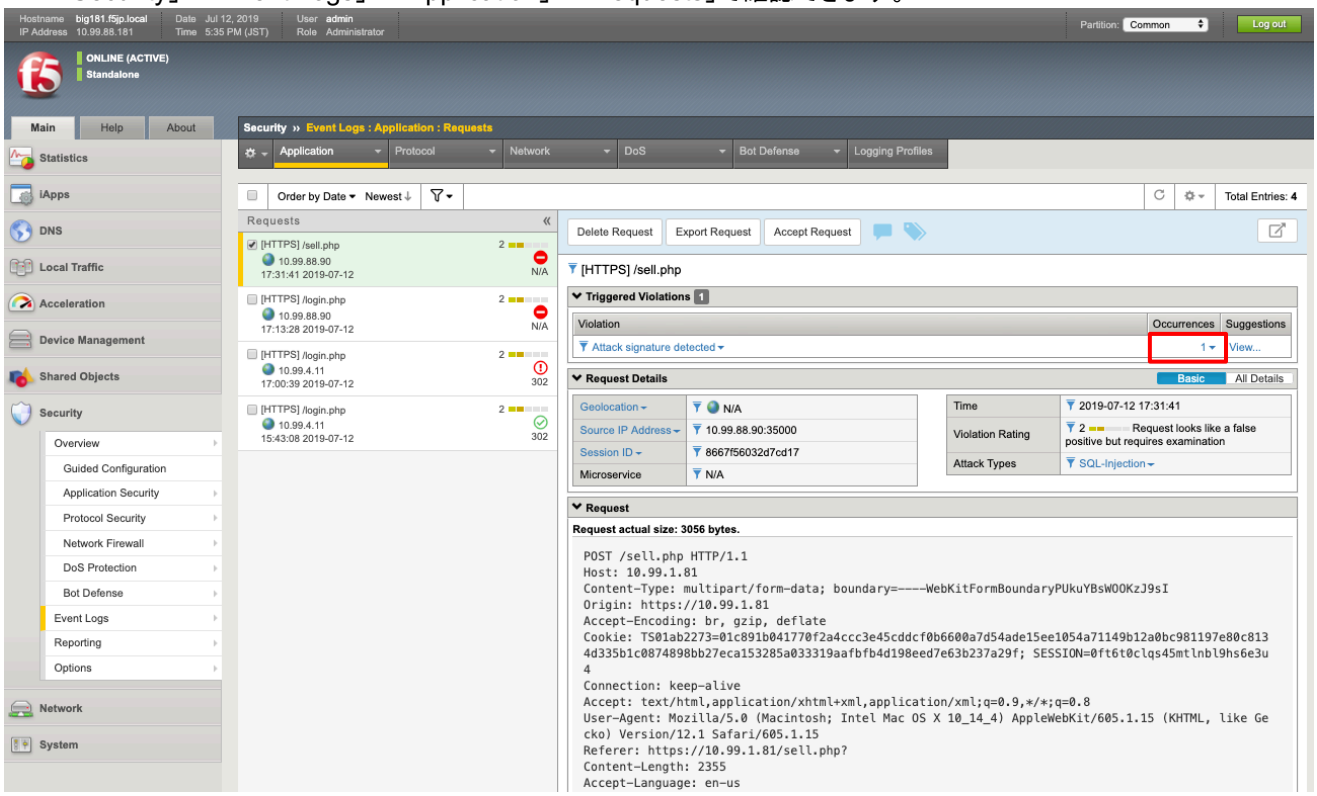
(2) Submit ボタンを押すと、攻撃として検知されてしまいます。



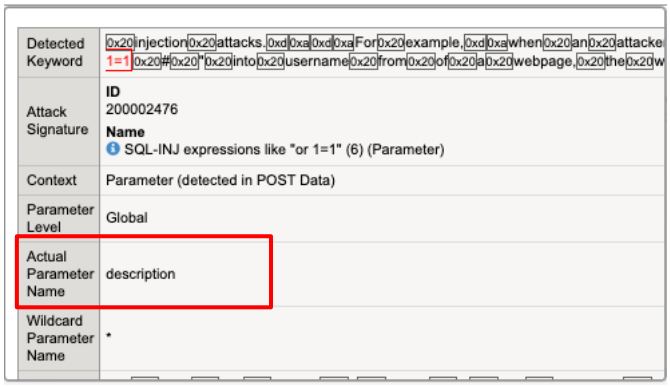
7.2. ［対策 1］誤検知したパラメータをホワイトリスト化する（シグネチャ検知はしない状態にする）

誤検知が発生したパラメータでは、Adv.WAF の全セキュリティ機能を無効化する＝ホワイトリスト化する、という方法を示します。

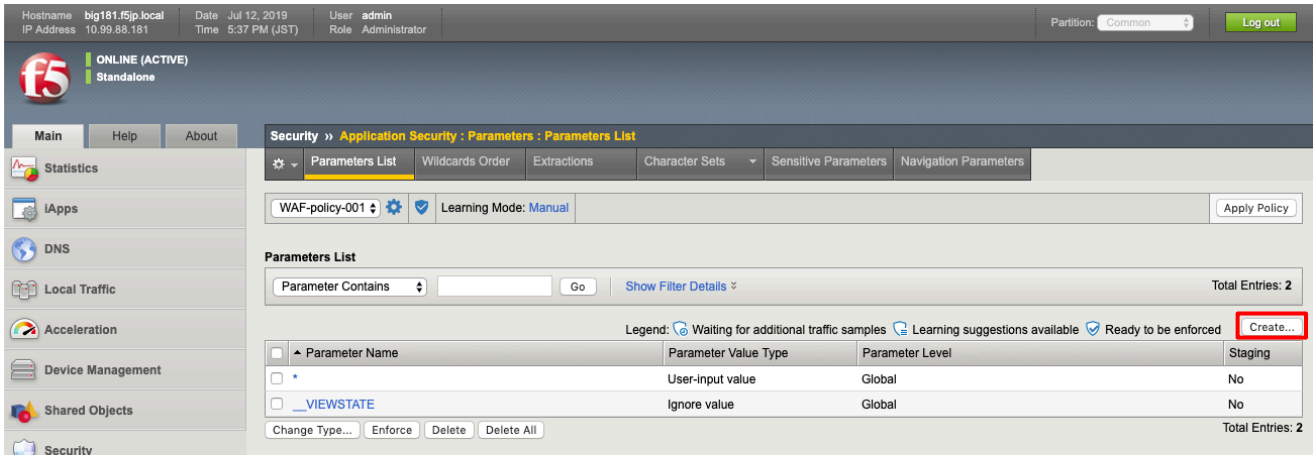
(1) 誤検知が発生した通信の Event ログを表示します。
「Security」→「Event Logs」→「Application」→「Requests」で確認できます。



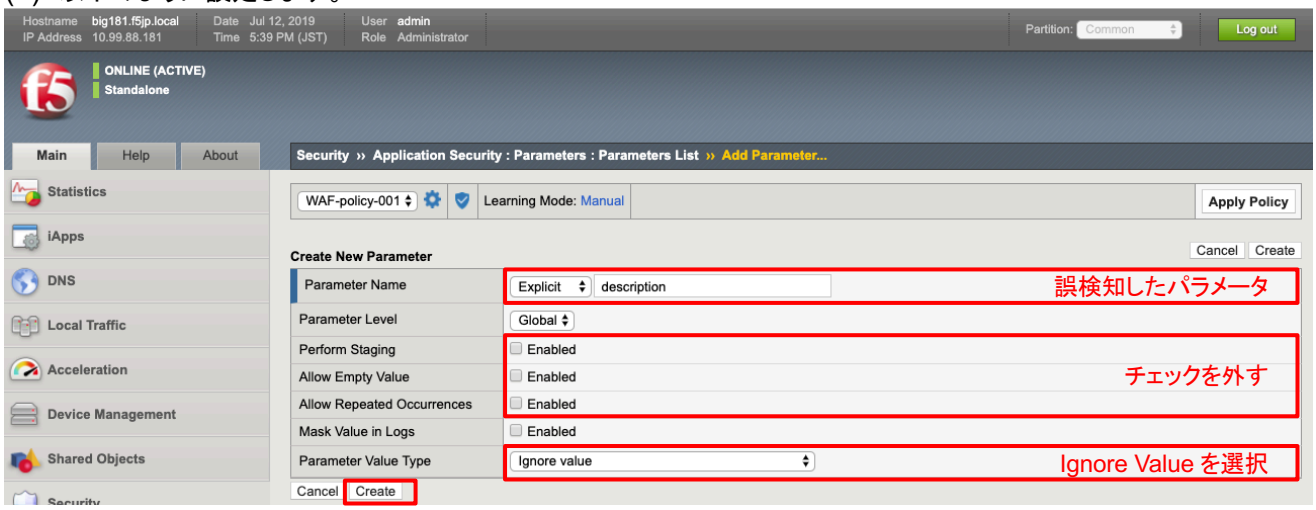
(2) 「Occurrences」をクリックして表示された画面を確認すると、誤検知したパラメータは、「description」であることが分かります。



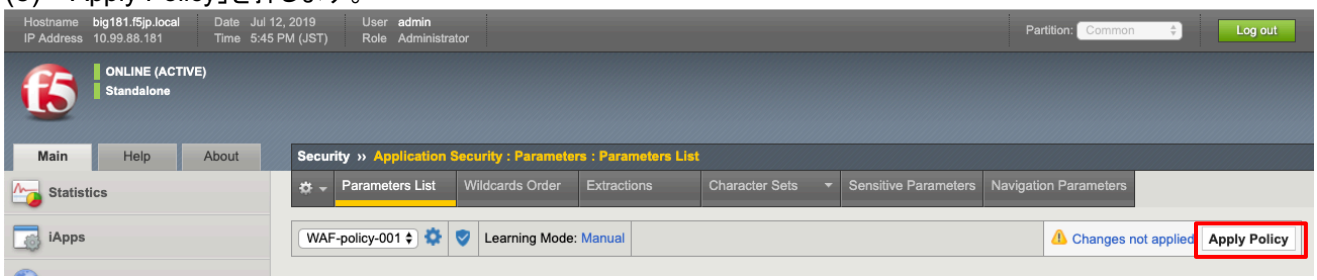
(3) 「Security」→「Application Security」→「Parameters」→「Parameters List」で表示された画面右上の「Create」ボタンを押します。



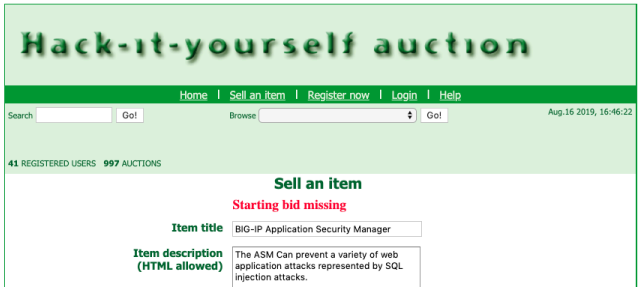
(4) 以下のように設定します。



(5) 「Apply Policy」を押します。



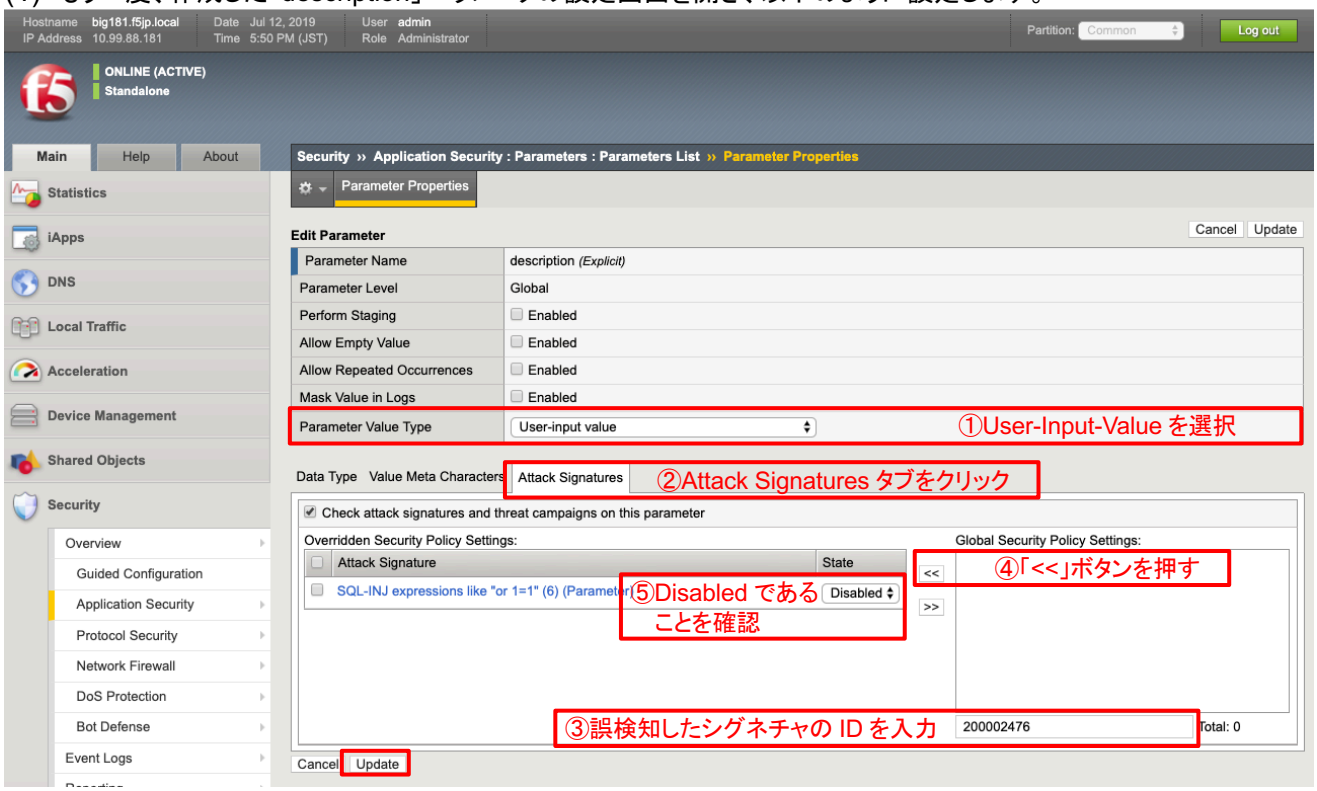
(6) クライアントのブラウザからもう一度、BIG-IP をオークションで売るための入力をしてみます。
今度は検知されず、次の画面に遷移します。
（F5LAB アプリの作りにより、まずは左の画面がされ、全て入力した場合には右のような画面に遷移します。）



7.3. ［対策 2］シグネチャの誤検知が発生したパラメータで、そのシグネチャを無効化する

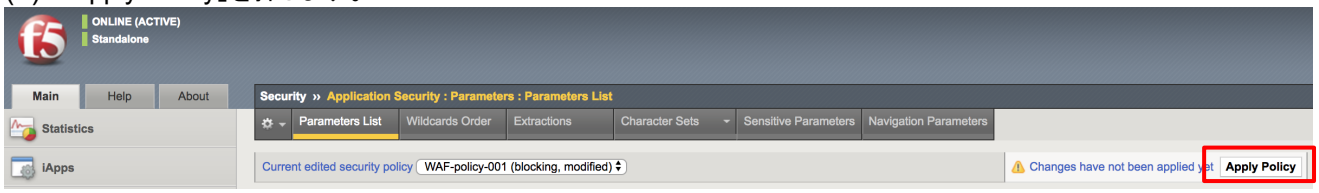
誤検知が発生したパラメータで、その誤検知したシグネチャだけを無効化する方法を示します。

(1) もう一度、作成した「description」パラメータの設定画面を開き、以下のように設定します。

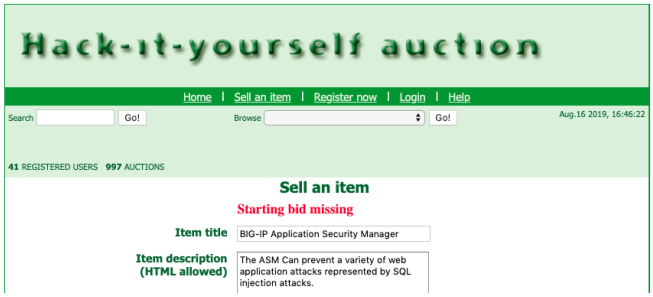


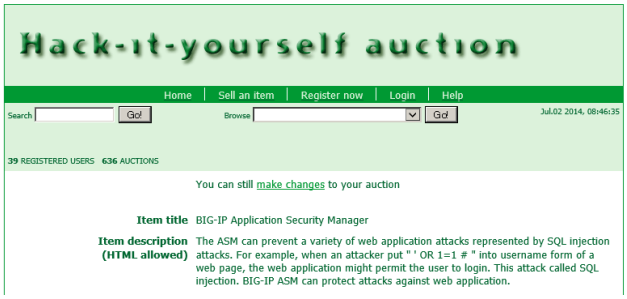
[ご参考]Signature ID は以下になります。

(2) 「Apply Policy」を押します。

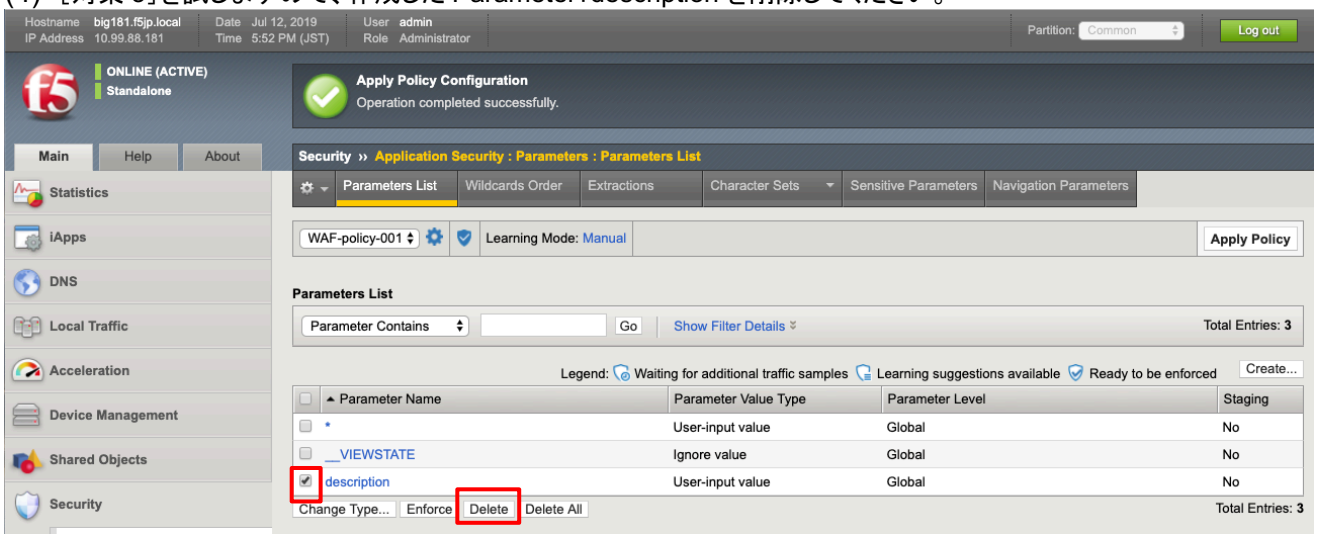


(3) クライアントのブラウザからもう一度、BIG-IP をオークションで売る入力をしてみます。
［対策 1］と同様、攻撃として検知されず、次の画面に遷移します。
（F5LAB アプリの作りにより、まずは左の画面がされ、全て入力した場合には右のような画面に遷移します。）

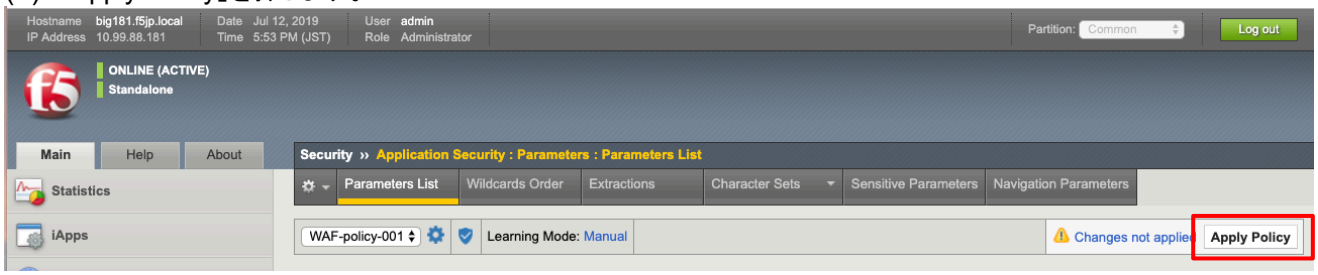




(4) ［対策 3］を試しますので、作成した Parameter：description を削除してください。



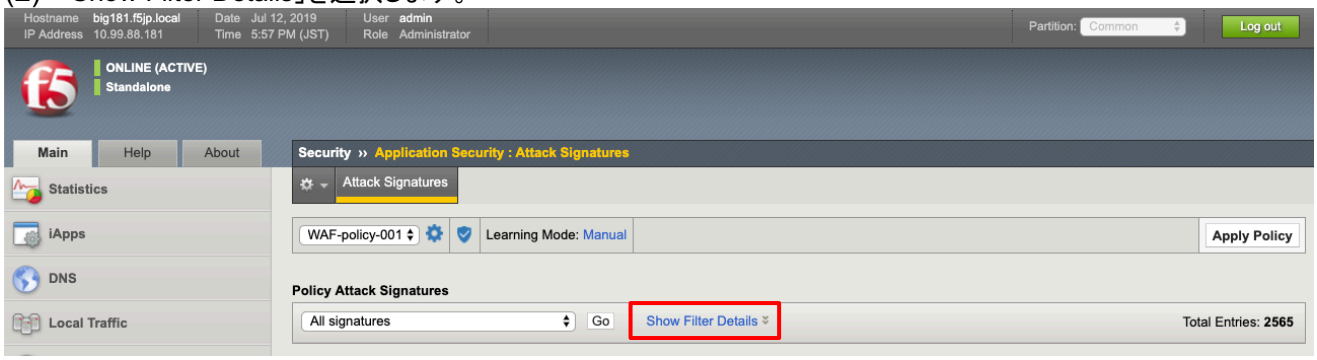
(5) 「Apply Policy」を押します。



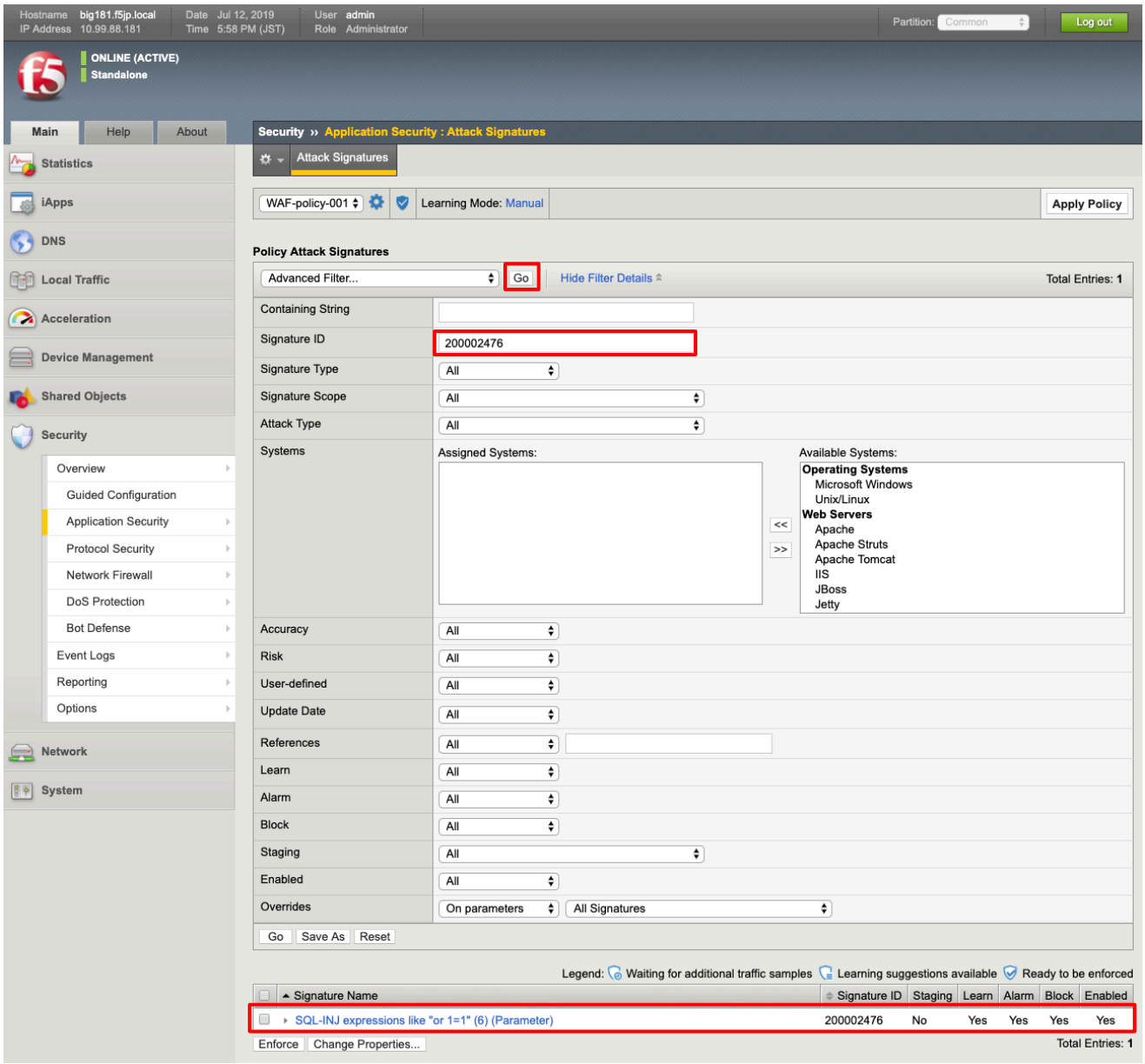
7.4. ［対策 3］誤検知したシグネチャを全体で無効化し、そのシグネチャで守りたいパラメータだけに適用する

誤検知したシグネチャを全体で無効化し、特定のパラメータにだけそのシグネチャを適用する方法を示します。ここでは、「username」にのみ、そのシグネチャを適用します。

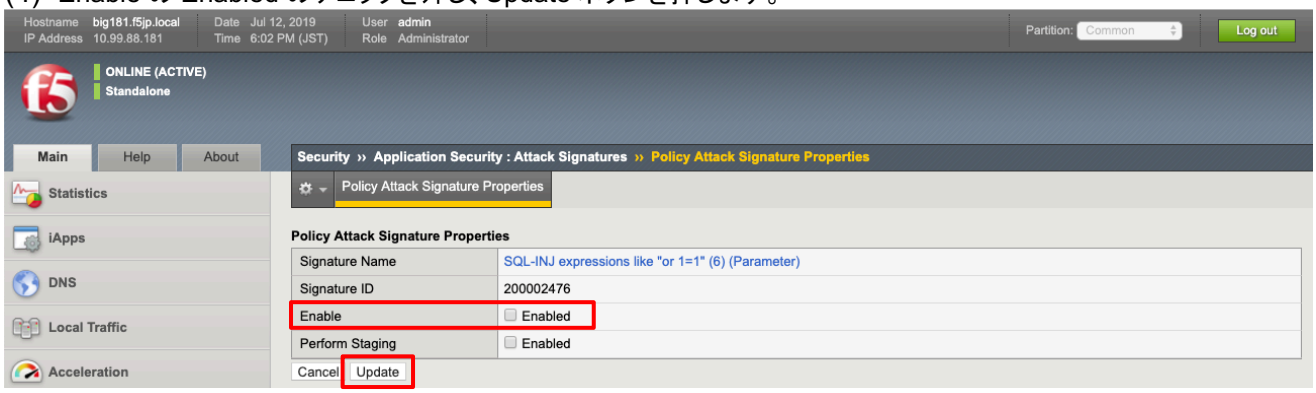
(1) 「Security」→「Application Security」→「Attack Signatures」を選択します。
(2) 「Show Filter Details」を選択します。



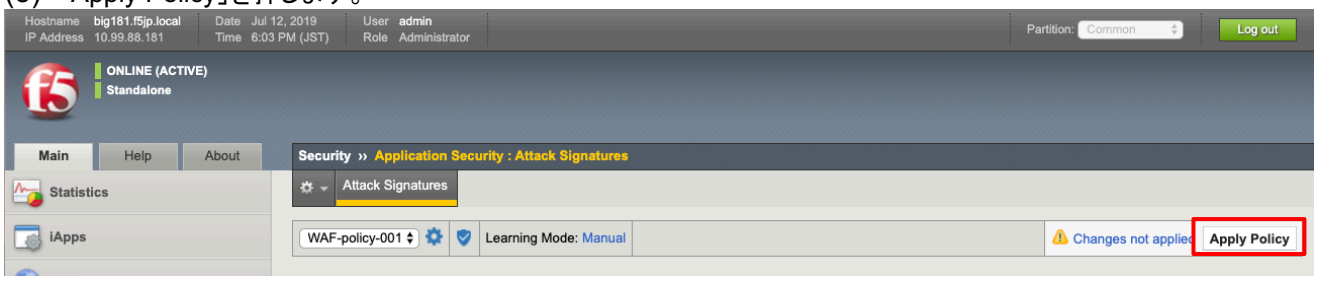
(3) Signature ID に誤検知したシグネチャ ID（200002476）を入力し、Go を押すと、該当のシグネチャが下の方に表示されます。該当シグネチャをクリックします。



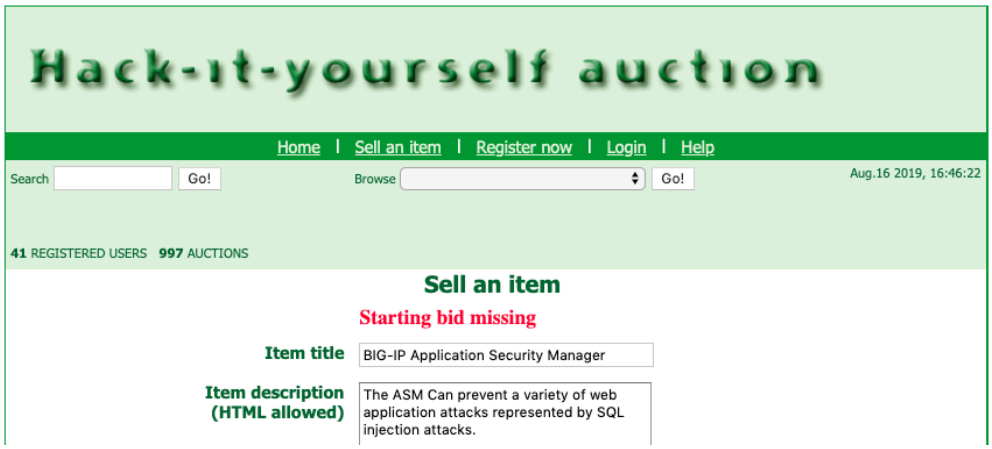
(4) Enable の Enabled のチェックを外し、Update ボタンを押します。



(5) 「Apply Policy」を押します。

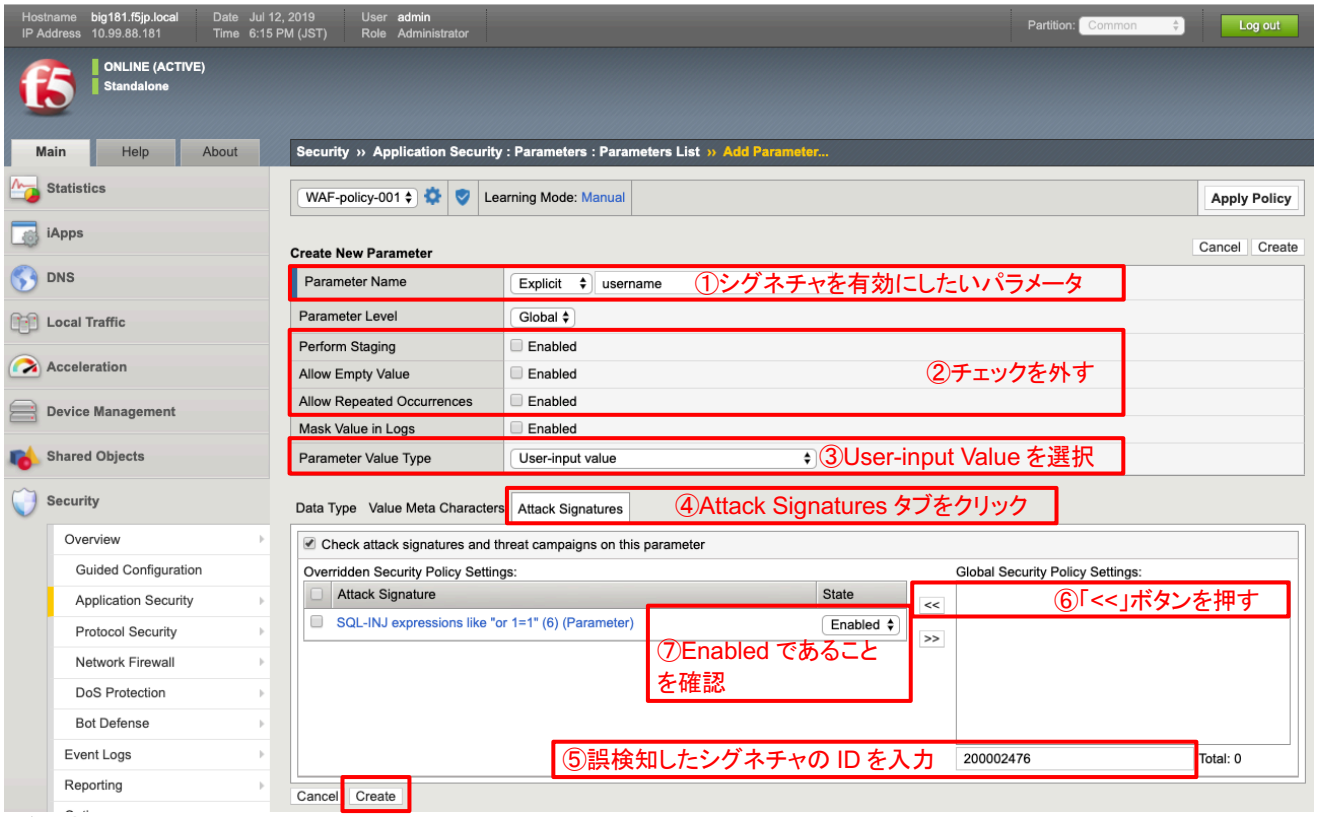


(6) クライアントのブラウザからもう一度、BIG-IP をオークションで売るための入力をしてみます。[対策 1][対策 2]と同様、攻撃として検知されず、次の画面に遷移します。

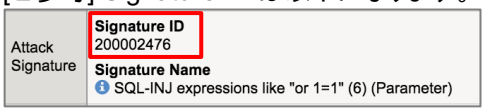


(7) しかし、username に「' OR 1=1 #」を入力して SQL インジェクション攻撃を実施すると、今度は通過してしまいます。以降、この対策を実施します。

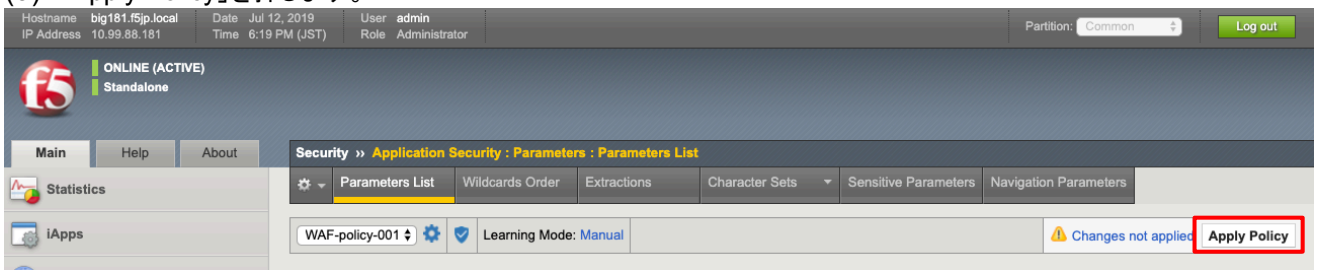
(8) 「Security」→「Application Security」→「Parameters」→「Parameters List」で表示された画面右上の「Create」ボタンを押すと、以下の画面が現れます。
「username」パラメータを作り、以下のように設定します。



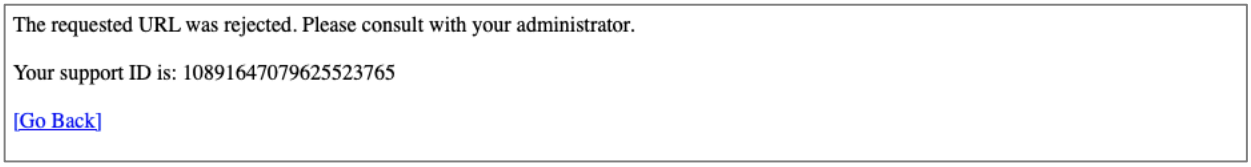
[ご参考] Signature ID は以下になります。



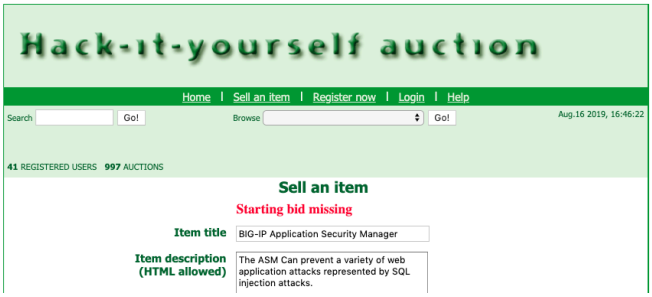
(9) 「Apply Policy」を押します。

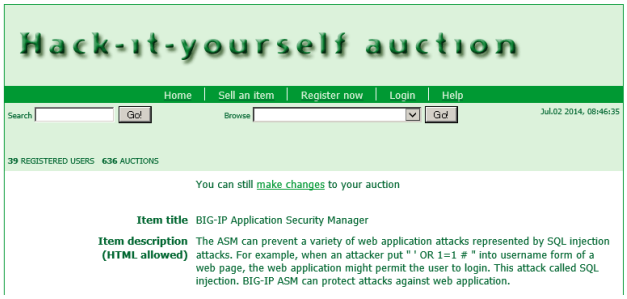


(10)クライアントのブラウザからもう一度、username に「' OR 1=1 #」を入力して SQL インジェクション攻撃を実施します。今度は SQL インジェクション攻撃を検知します。



(11) クライアントのブラウザからもう一度、BIG-IP をオークションで売る入力をしてみます。
攻撃として検知されず、画面が遷移します。
（F5LAB アプリの作りにより、まずは左の画面がされ、全て入力した場合には右のような画面に遷移します。）





8. ［Step3］File Type の設定を行う（強制ブラウジング対策）

強制ブラウジングは、ユーザに開示するつもりのないファイル（例えば、バックアップ用ファイル（“XX.bak”と名づけたもの）やシステム用ファイル（XX.conf）など）を攻撃者が取得する、という攻撃です。

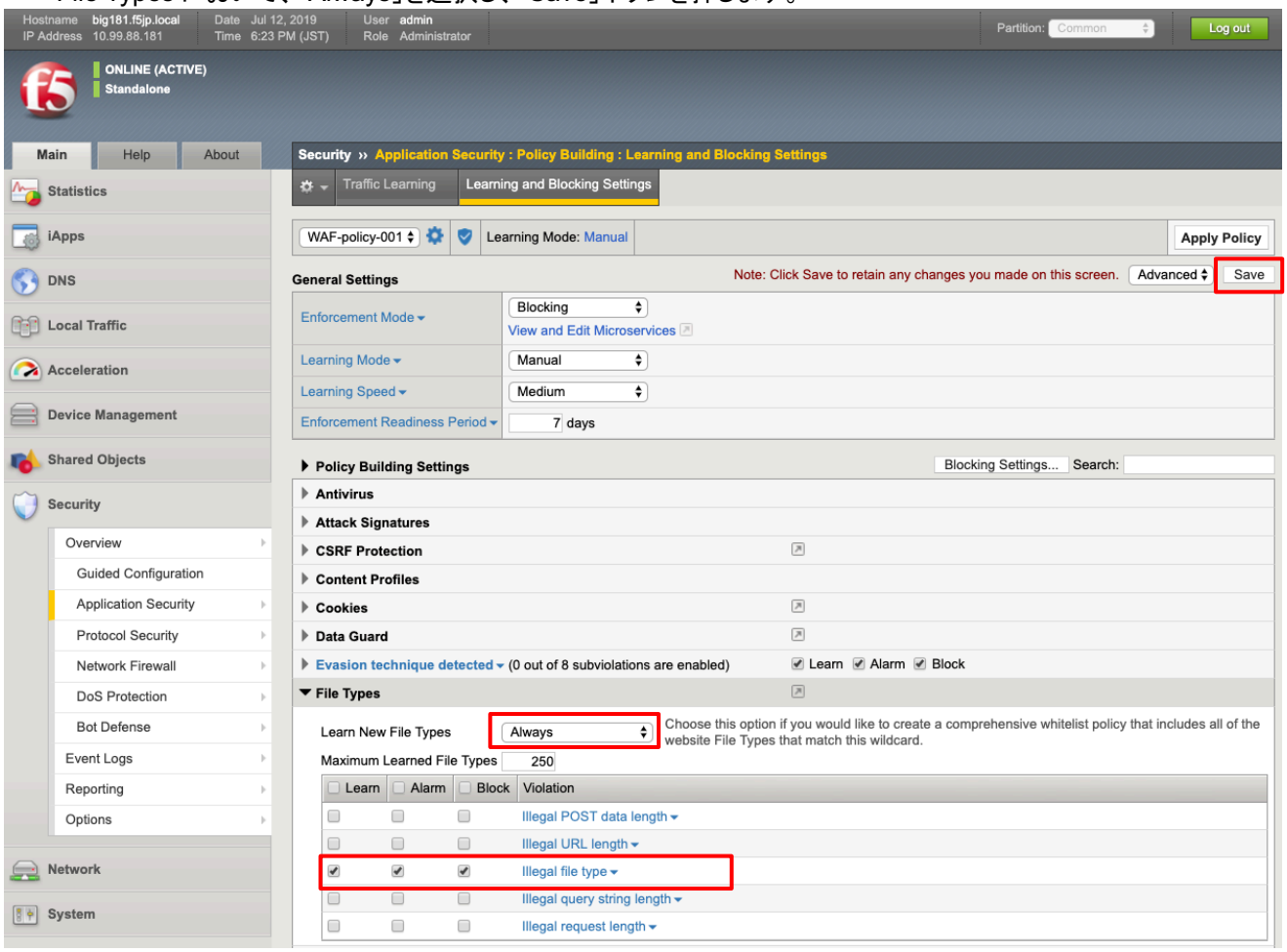
この攻撃は、開示するファイルの種類を限定しておくことで防御できます。

以下にその設定方法を示します。

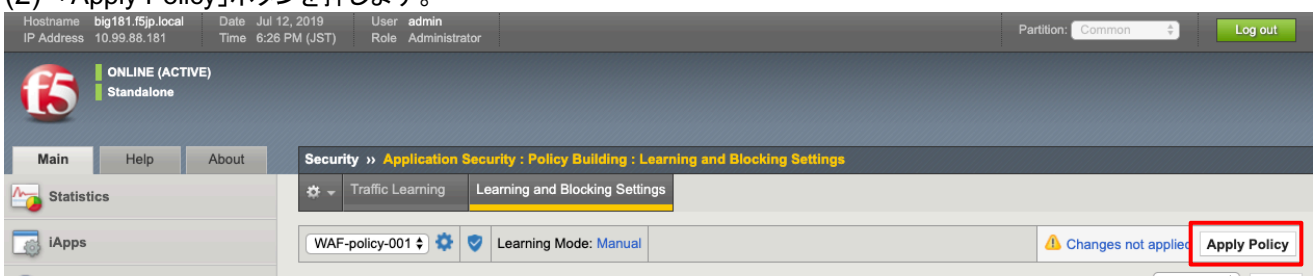
8.1. File Types の設定

8.1.1. File Types の学習

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「Illegal file Type」を探します。以下のように、Learn / Alarm / Block を有効にします。また、Learn New File Types において、「Always」を選択し、「Save」ボタンを押します。

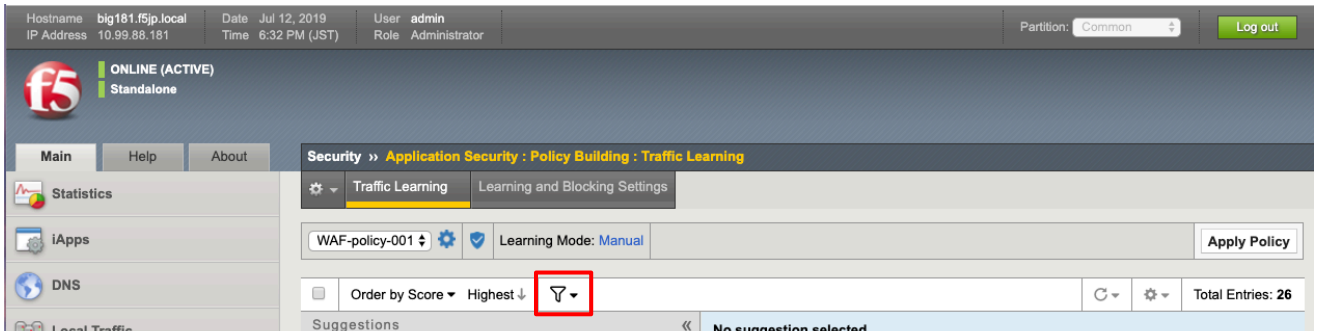


(2) 「Apply Policy」ボタンを押します。

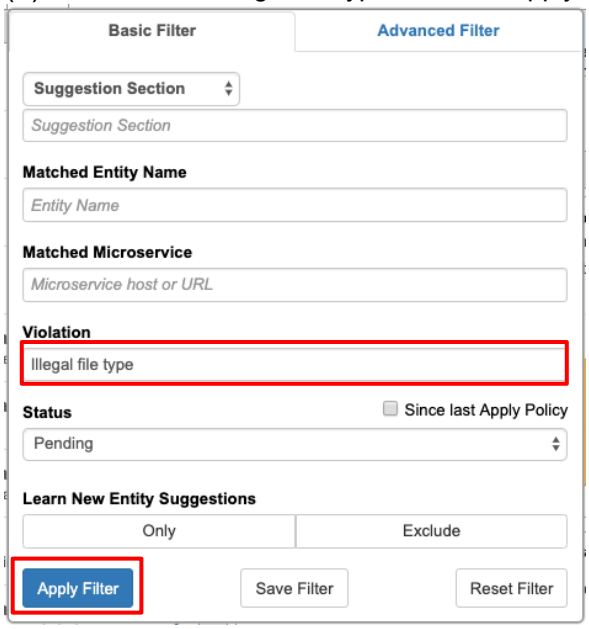


(3) クライアントのブラウザのキャッシュを全部消してください。
(4) クライアントのブラウザで、正当なユーザ(f5userX)でログインして、Web ページを巡回してください。

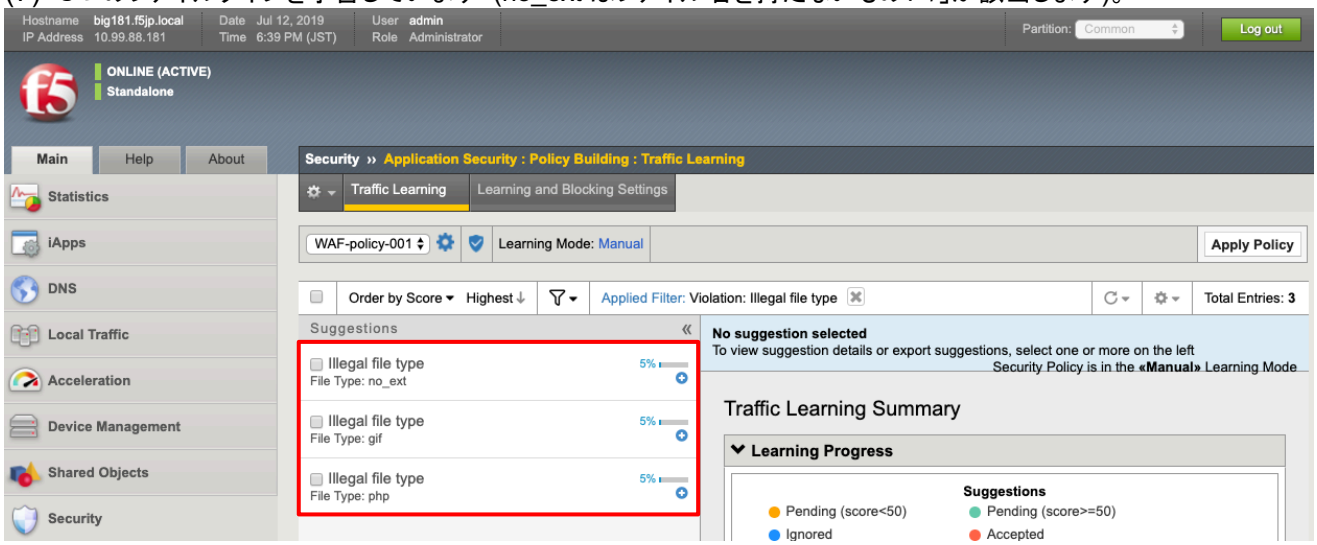
(5) 「Security」→「Application Security」→「Policy Building」→「Traffic Learning」で表示された画面で、Filtering アイコンをクリックします。



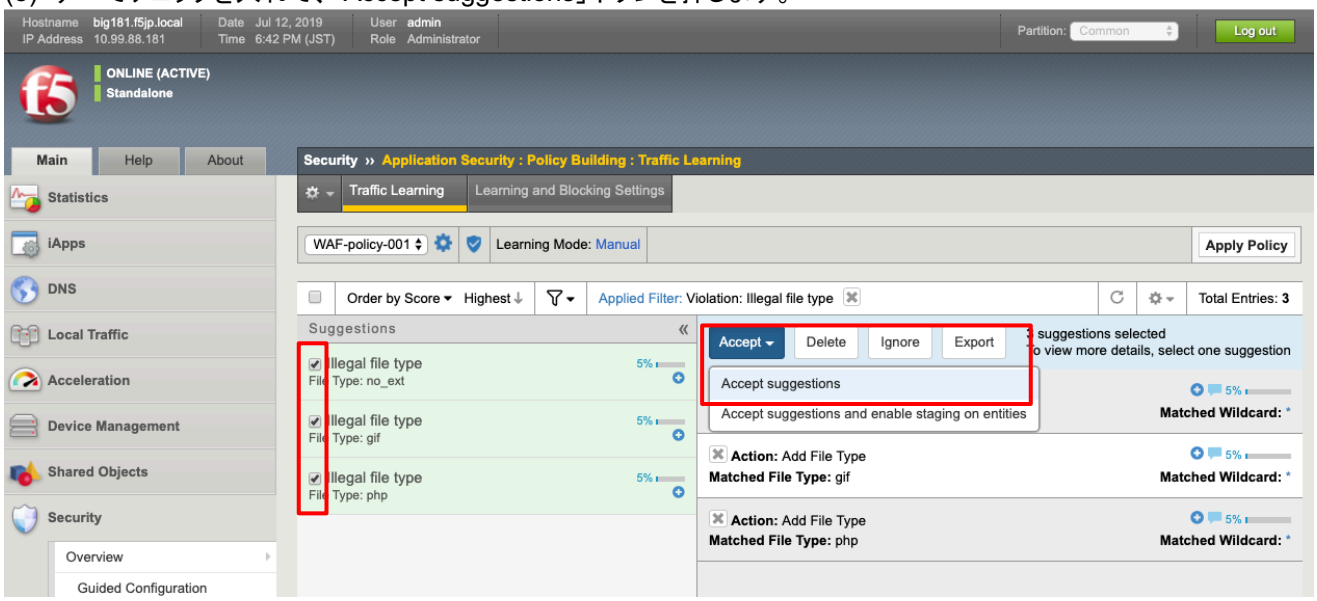
(6) Violation から illegal file type を選択し、Apply Filter を押します。



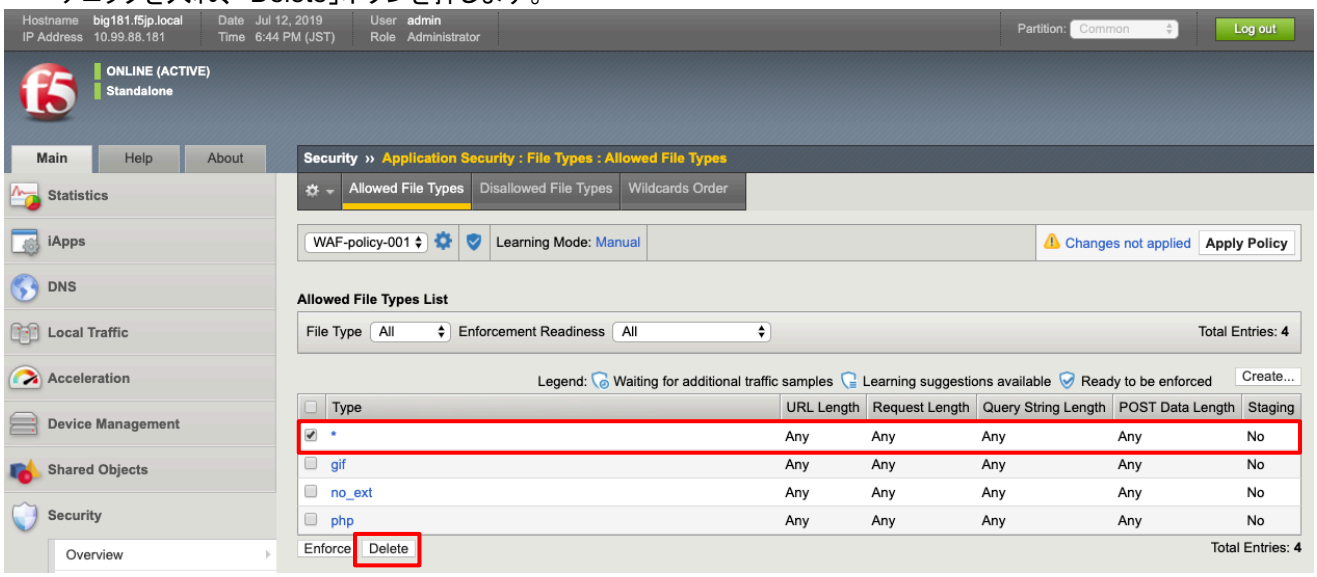
(7) 3つのファイルタイプを学習しています (no_ext はファイル名を持たないもの:「/」が該当します)。



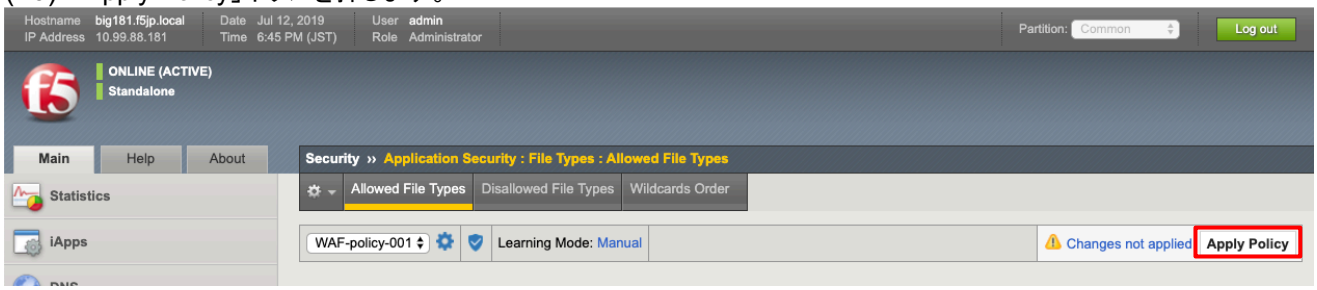
(8) すべてチェックを入れて、「Accept suggestions」ボタンを押します。



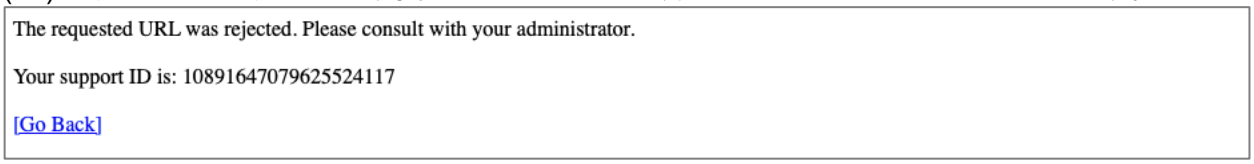
(9) 「Security」→「Application Security」→「File Types」→「Allowed File Types」で、Accept したファイルがここに表示されます。この学習済みのファイルだけを通過させる、という設定にしますので、「*」を削除します。「*」の先頭にチェックを入れ、「Delete」ボタンを押します。



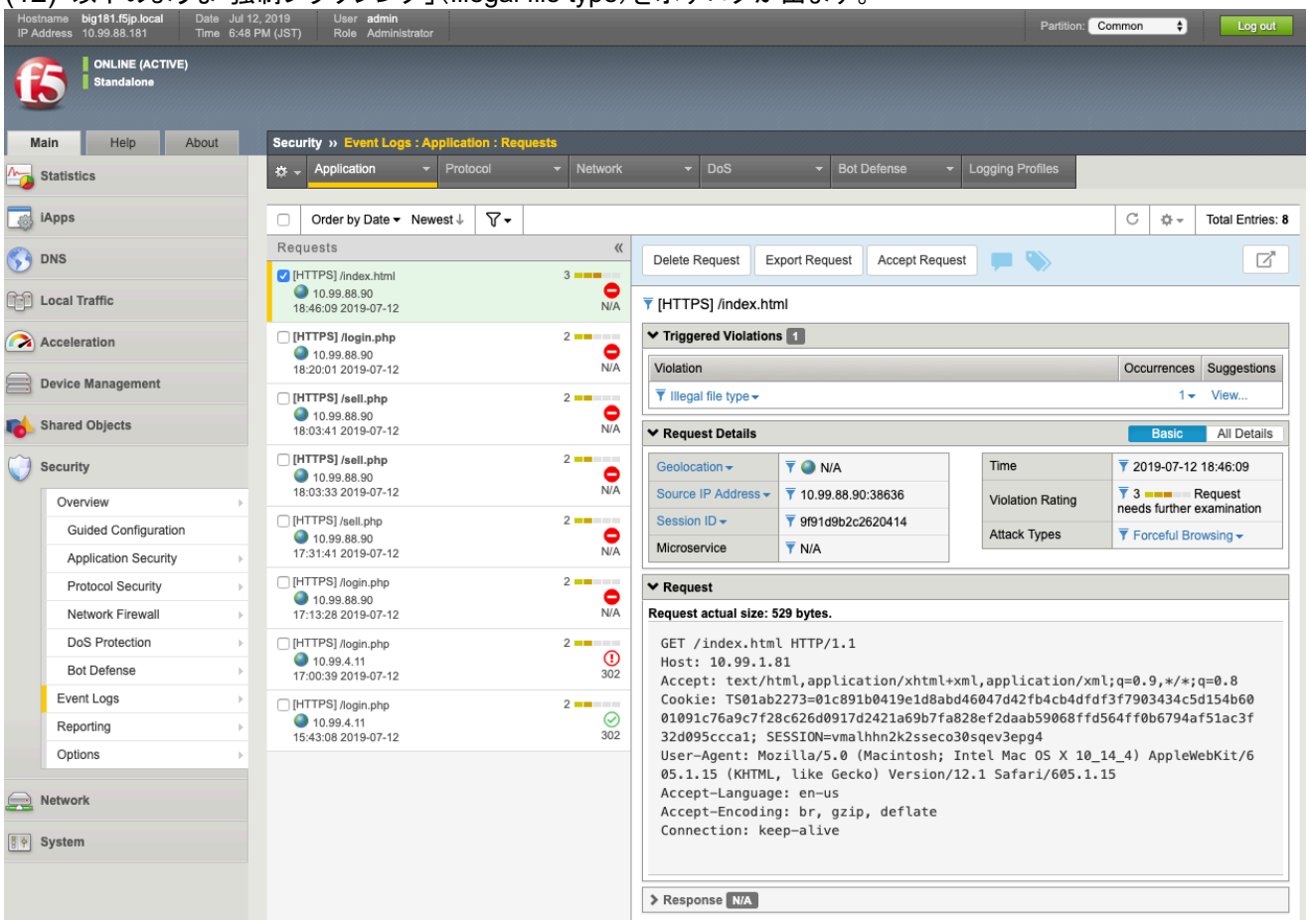
(10) 「Apply Policy」ボタンを押します。



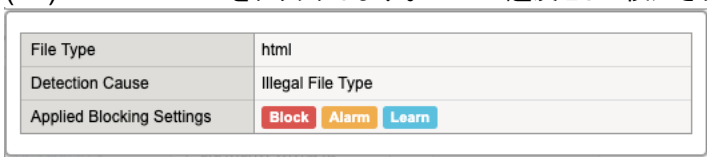
(11) クライアントのブラウザから、学習した以外のファイル、例えば Index.html へアクセスしてみます。



(12) 以下のような「強制ブラウジング」（Illegal file type）を示すログが出ます。



(13) Occurrences をクリックします。html が違反として検知されていることが分かります。



(14) 本ガイドでは、学習したファイルのみ通過させる設定を例に取りましたが、実際の環境においては、以下 2 つのどちらかで対応してください。

① この学習した File Types だけを設定（ホワイトリスト化）する。（本ガイドと同じ方法）

② 利用しないファイルを設定（ブラックリスト化）する。

(ア) 学習したファイルがあまりに多すぎるから
(イ) 利用しないファイルは明確になっているから、など

この場合は、「Security」→「Application Security」→「File Types」→「Disallowed File Types」で指定します。

9. ［Step4］追加機能の要否検討

防御対象の Web アプリケーションに対して、「Step1」～「Step3」で設定した機能に加え、以下のセキュリティ機能が必要かどうかを判断します。

① Data Guard（クレジットカード番号のマスキング）

② L7 DoS への対策

③ Brute Force への対策

④ CSRF への対策

⑤ Threat Campaings シグネチャ

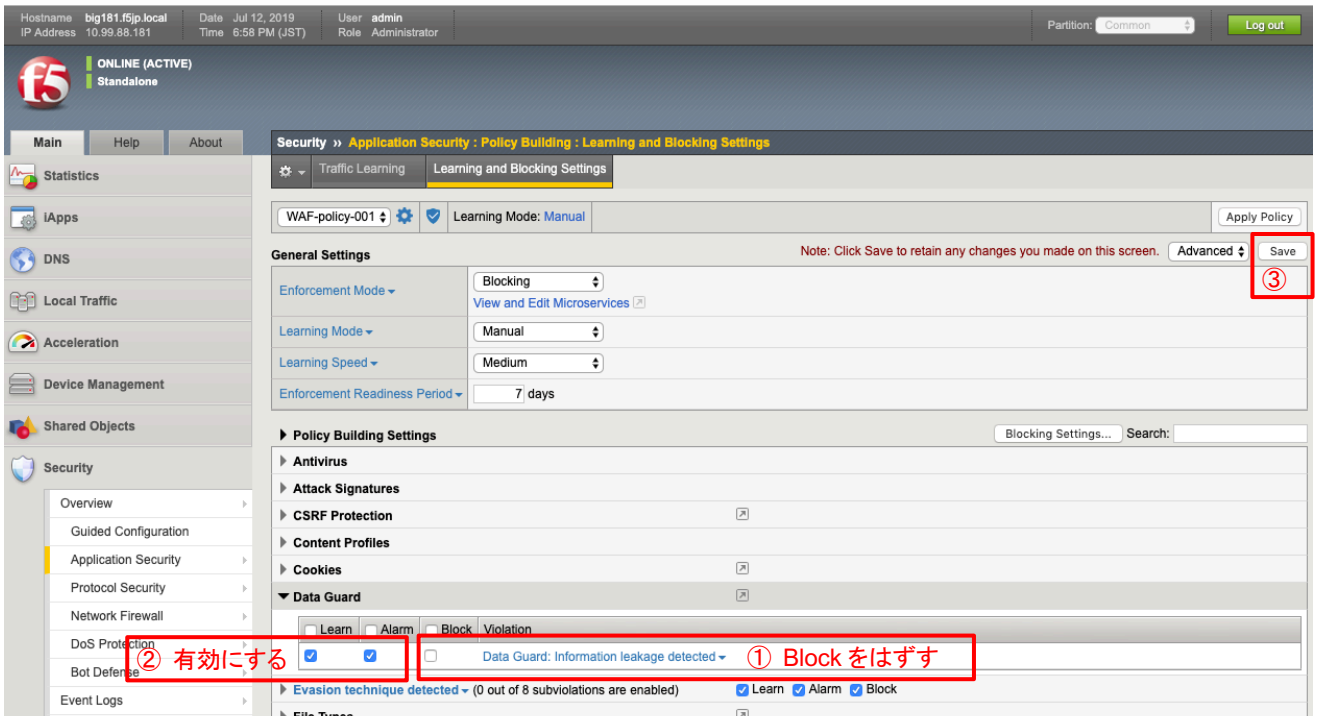
必要に応じて、追加で設定を実施してください。

以降、ぞれぞれの設定方法を記載します。

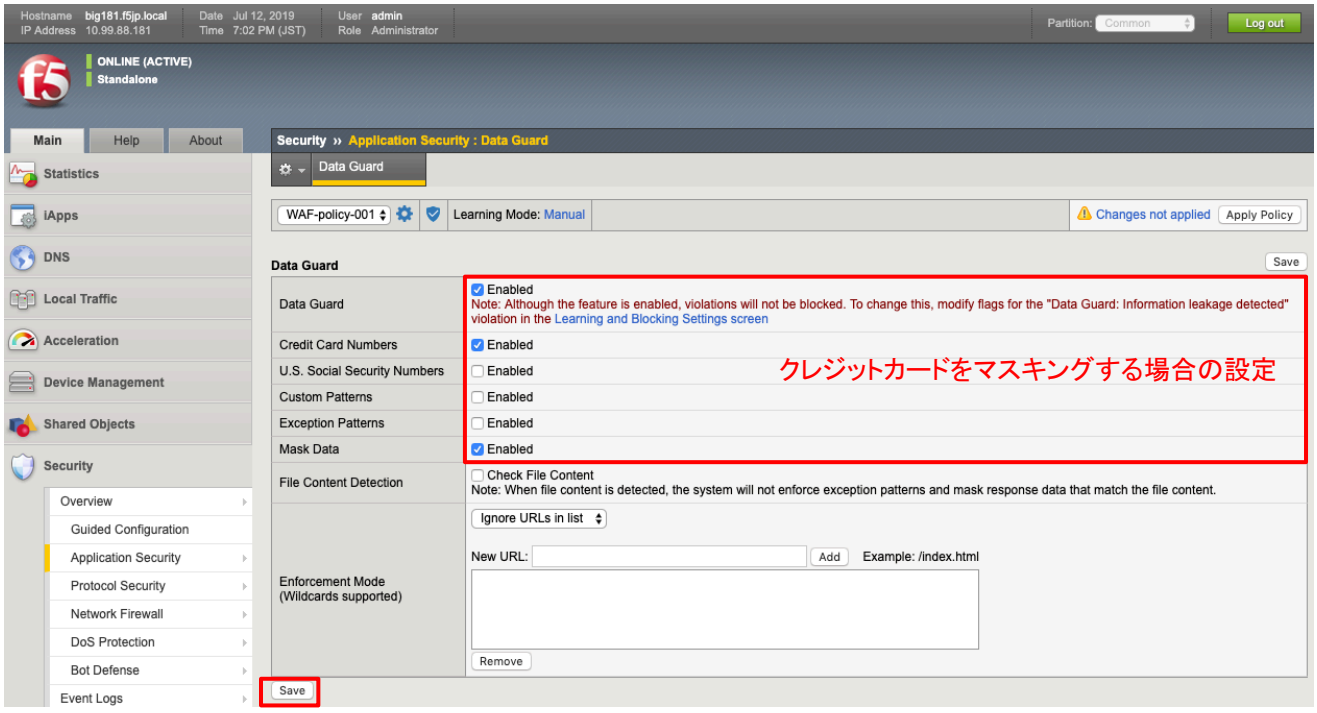
9.1. Data Guard

クレジットカード番号の漏洩の危険性がある Web アプリケーションの場合、本機能により防御が可能です。
HTTP レスポンスにクレジットカード番号が存在する場合、Adv.WAF はその番号をマスキングすることができます。

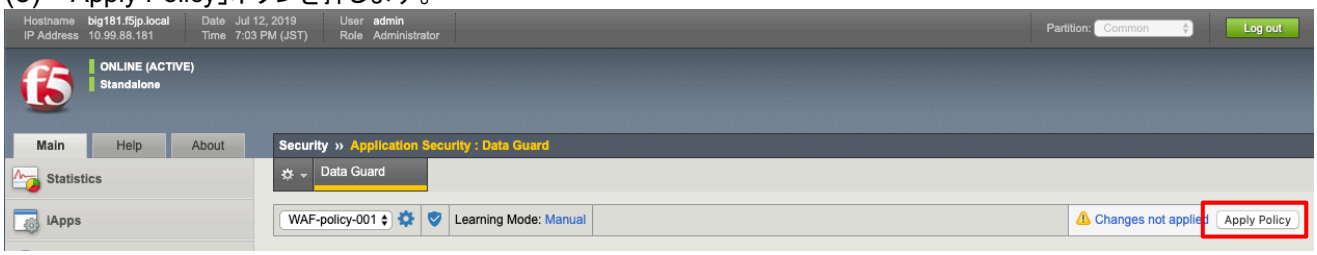
(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「Data Guard: Information leakage detected」を探します。
以下のように、Learn / Alarm を有効にし、Block をはずし、「Save」ボタンを押します。



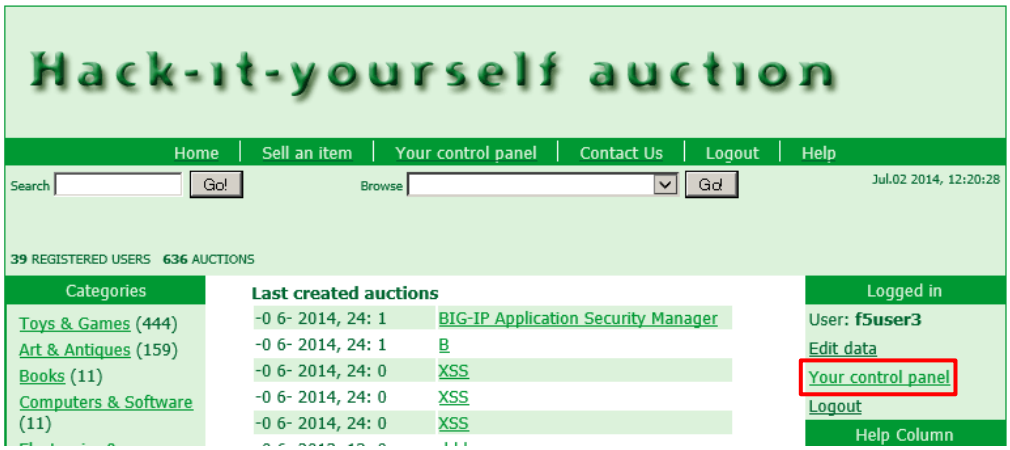
(2) 「Security」→「Application Security」→「Data Guard」で表示された画面で、以下のように設定し、「Save」ボタンを押します。



(3) 「Apply Policy」ボタンを押します。



(4) クレジットカード番号が登録されたユーザでログインします。
「Your Control Panel」をクリックします。



(5) 以下のように、クレジットカード番号がマスキングされます。

※F5 ハンズオンでは実施しません

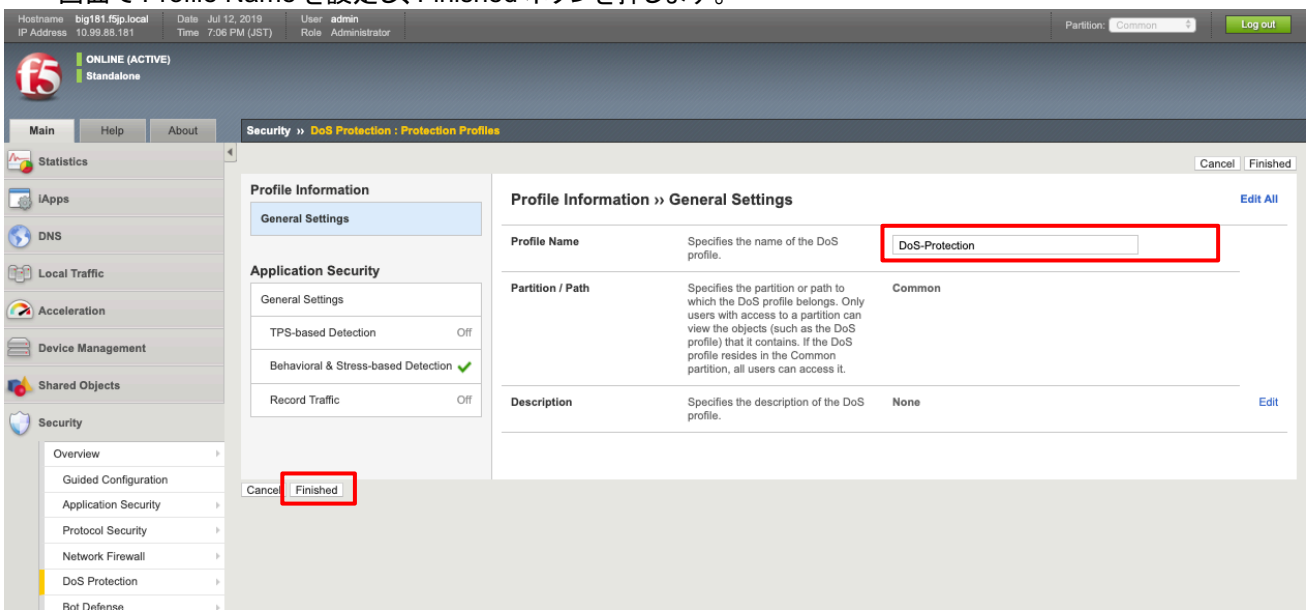
9.2. L7 DoS への対策 ＜ご参考＞

DoS Protection 設定を行うことで、大量の HTTP リクエストが送り込まれてきていることを検知し、緩和することが可能です。

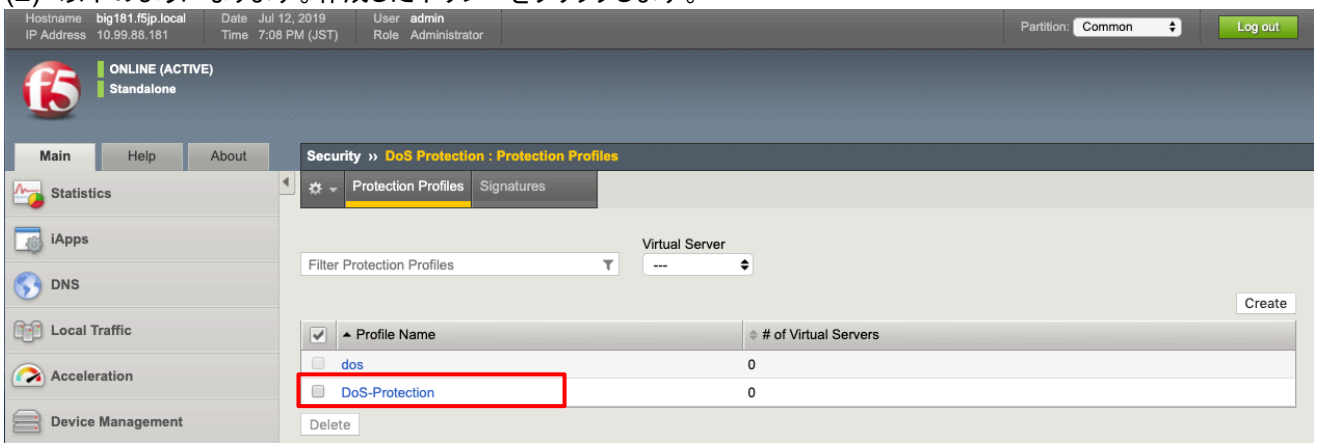
9.2.1. TPS-Based DoS

秒間の HTTP トランザクション数の変化で検知を行います。

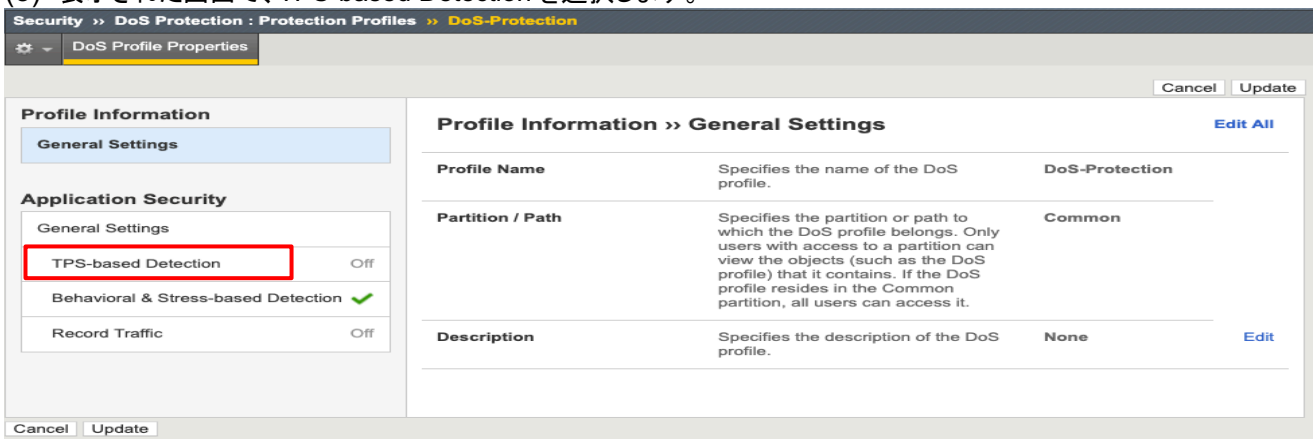
(1) 「Security」→「DoS Protection」→「DoS Profiles」で表示された画面の右上にある「Create」ボタンを押し、現れた画面で Profile Name を設定し、Finished ボタンを押します。



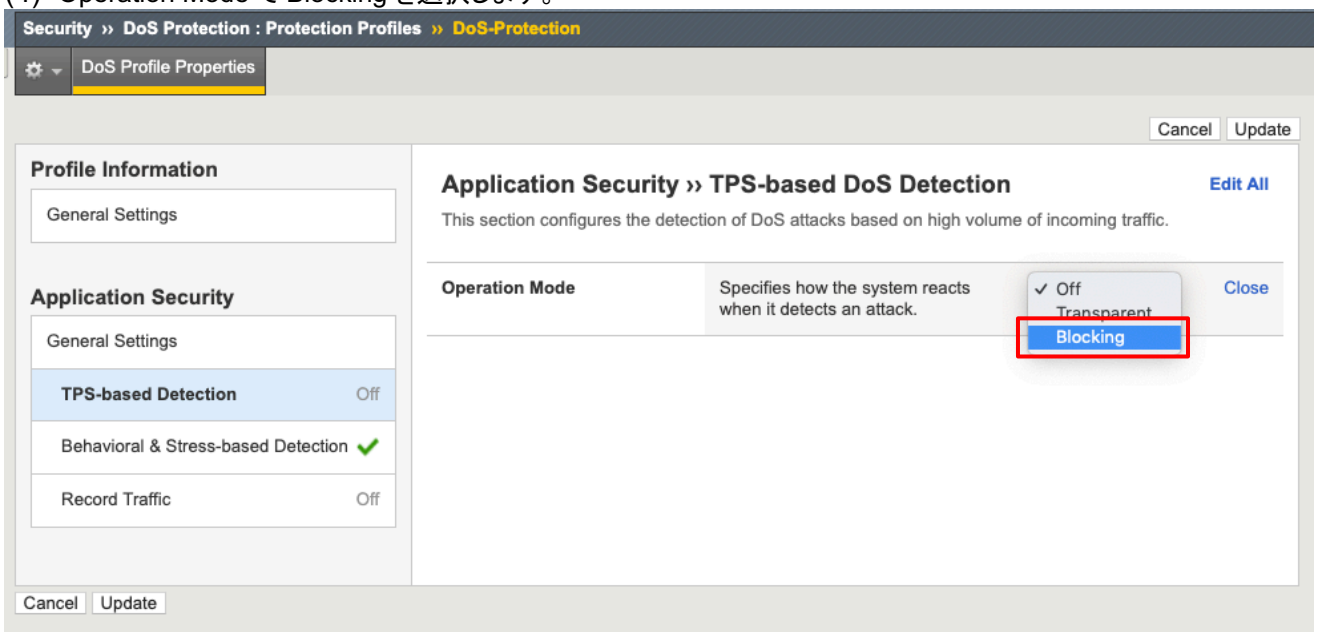
(2) 以下のようになります。作成したポリシーをクリックします。



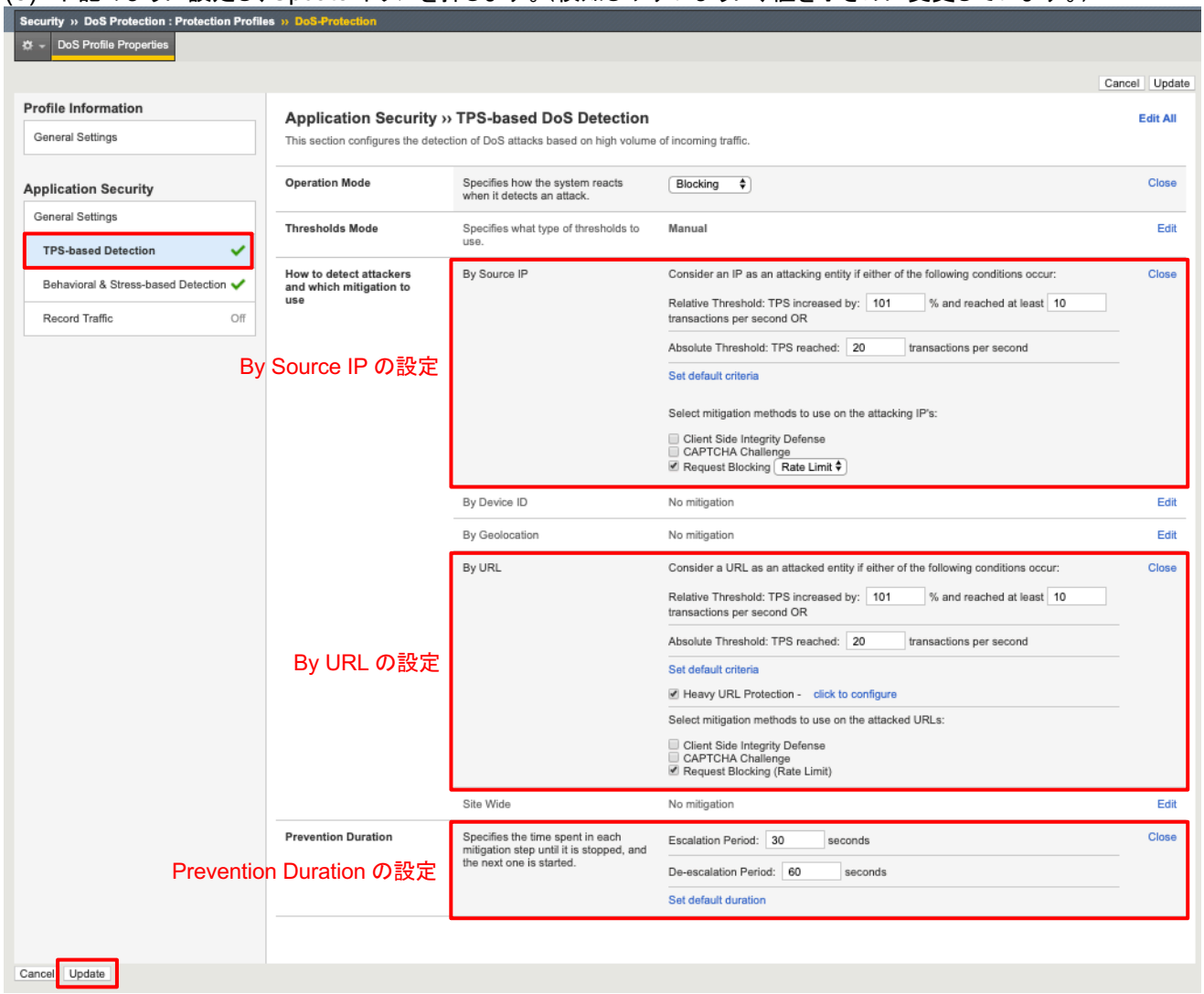
(3) 表示された画面で、TPS-based Detection を選択します。



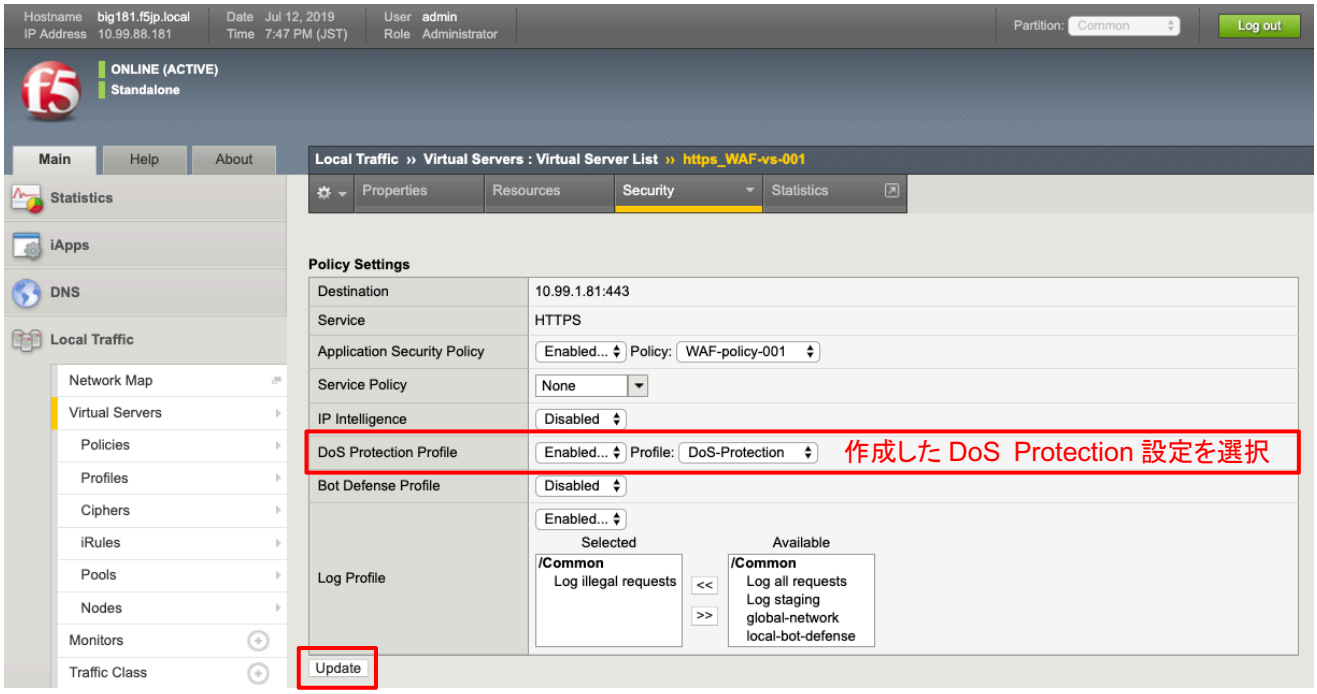
(4) Operation Mode で Blocking を選択します。



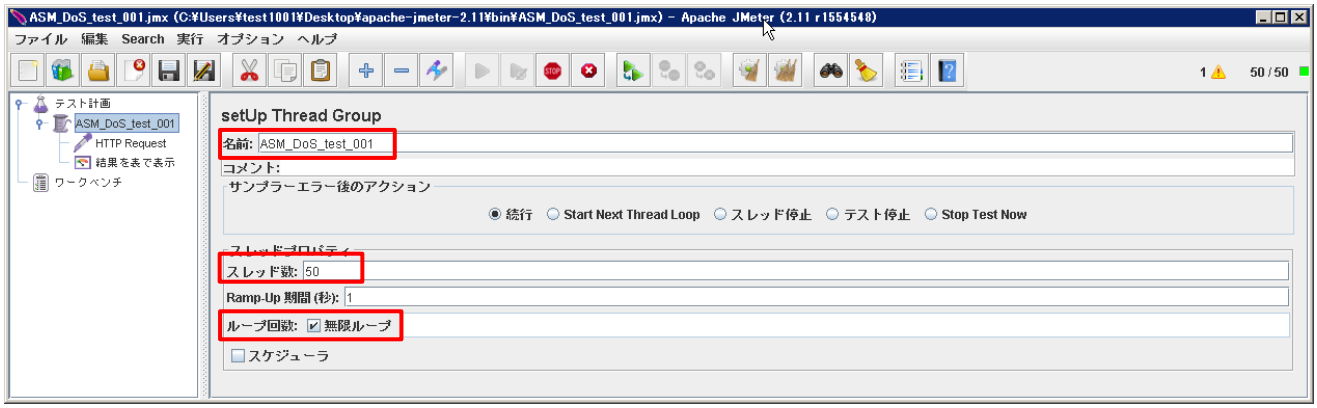
(5) 下記のように設定し、Update ボタンを押します。（検知しやすいように、値を小さめに変更しています。）



(6) 「Local Traffic」→「Virtual Servers」で表示された Virtual Servers のうち、DoS プロテクションを実施したい Virtual Server を選択します。
「Security」タブ→「Policies」で現れた画面で作成した DoS プロテクションポリシーを選択し、アップデートボタンを押します。



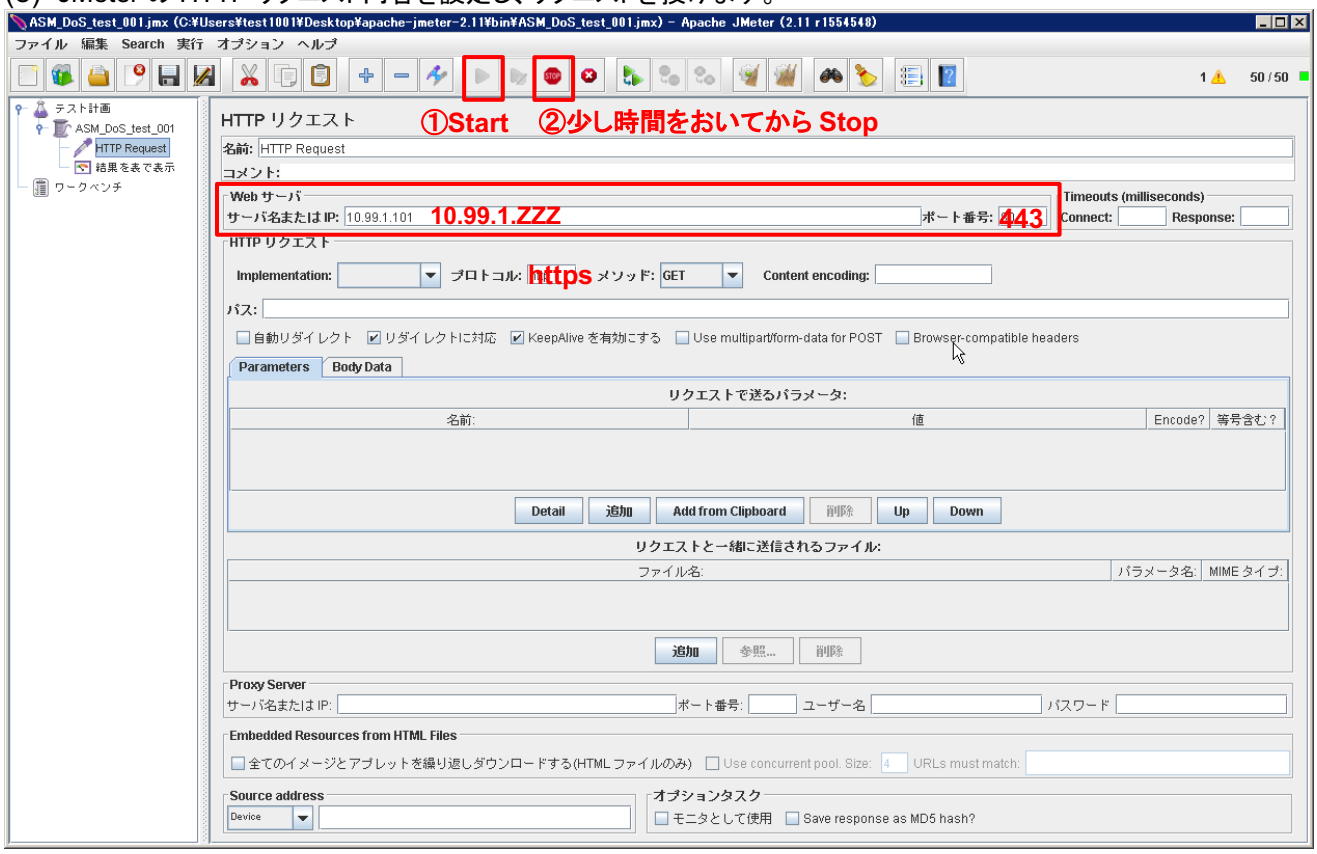
(7) 本検証では、クライアント PC 上の JMeter を使って、多量の HTTP リクエストを生成します。
スレッド：50



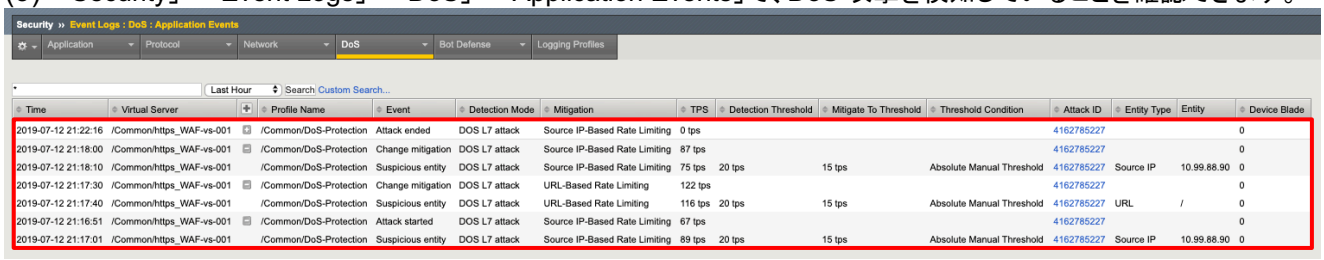
① 「テスト計画」を右クリックし、「Threads(Users)」の「setUp Thread Group」を追加。「無限ループ」にチェックを入れる。

② ①で作成した Thread Group を右クリックして、「サンプラー」の「HTTP リクエスト」を追加します。

(8) JMeter の HTTP リクエスト内容を設定し、リクエストを投げます。



(9) 「Security」→「Event Logs」→「DoS」→「Application Events」で、DoS 攻撃を検知していることを確認できます。



各 Attack ID をクリックすると、DoS Dashboard report を確認できます。

9.2.2. Stress-Based DoS

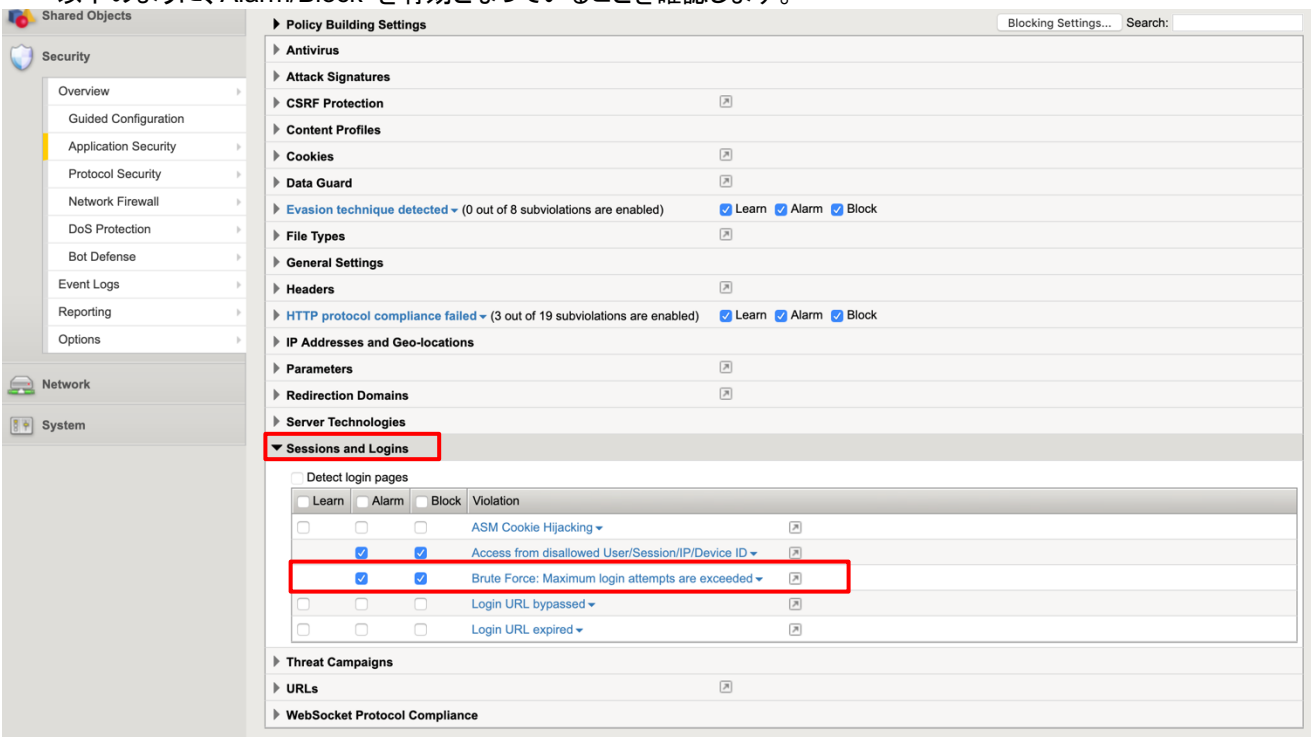
Stress-Based DoS は、TPS ベースと WEB サーバへのレイテンシをチェックする DoS 対策方法で、サーバのリソースをゆっくりと枯渇させるような攻撃の対策に有効です。

9.3. Brute Force の対策

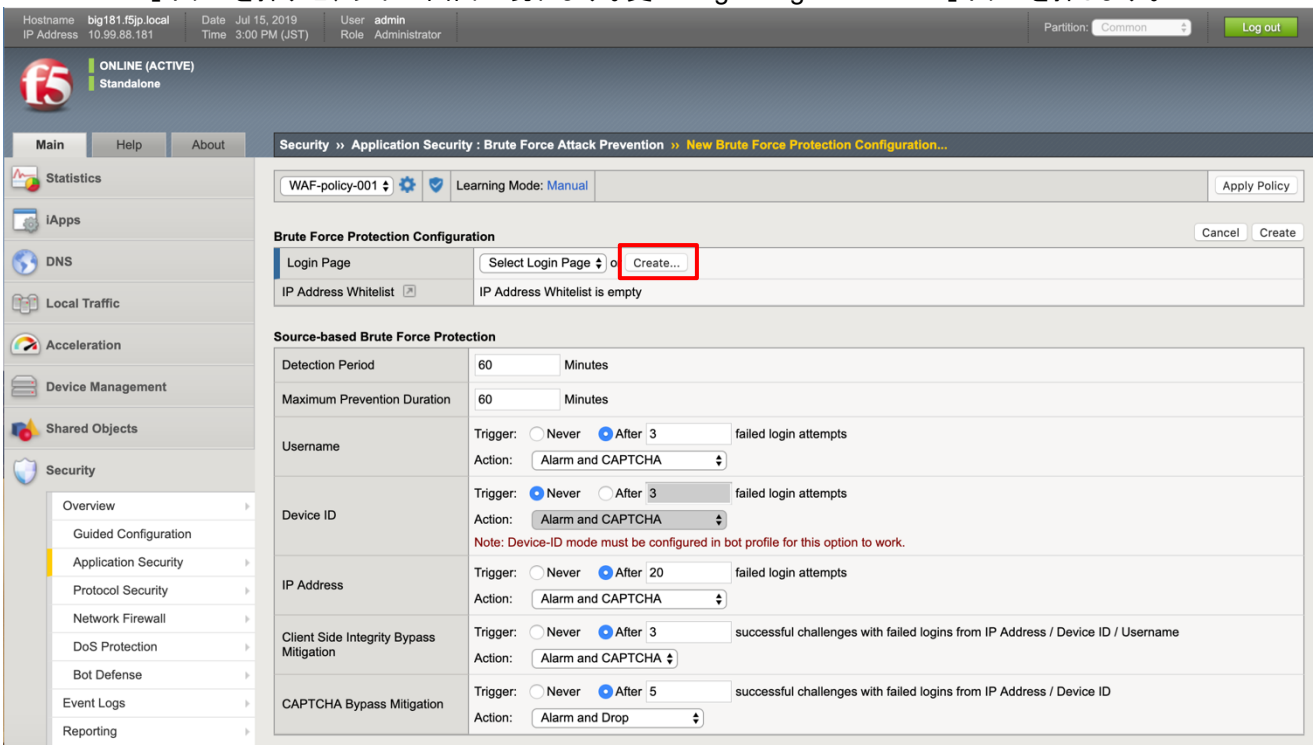
Adv.WAF は、ソース情報（ユーザ名、IP アドレスなど）ベースで Brute Force 対策を行っています。

9.3.1. Brute Force 対策の設定

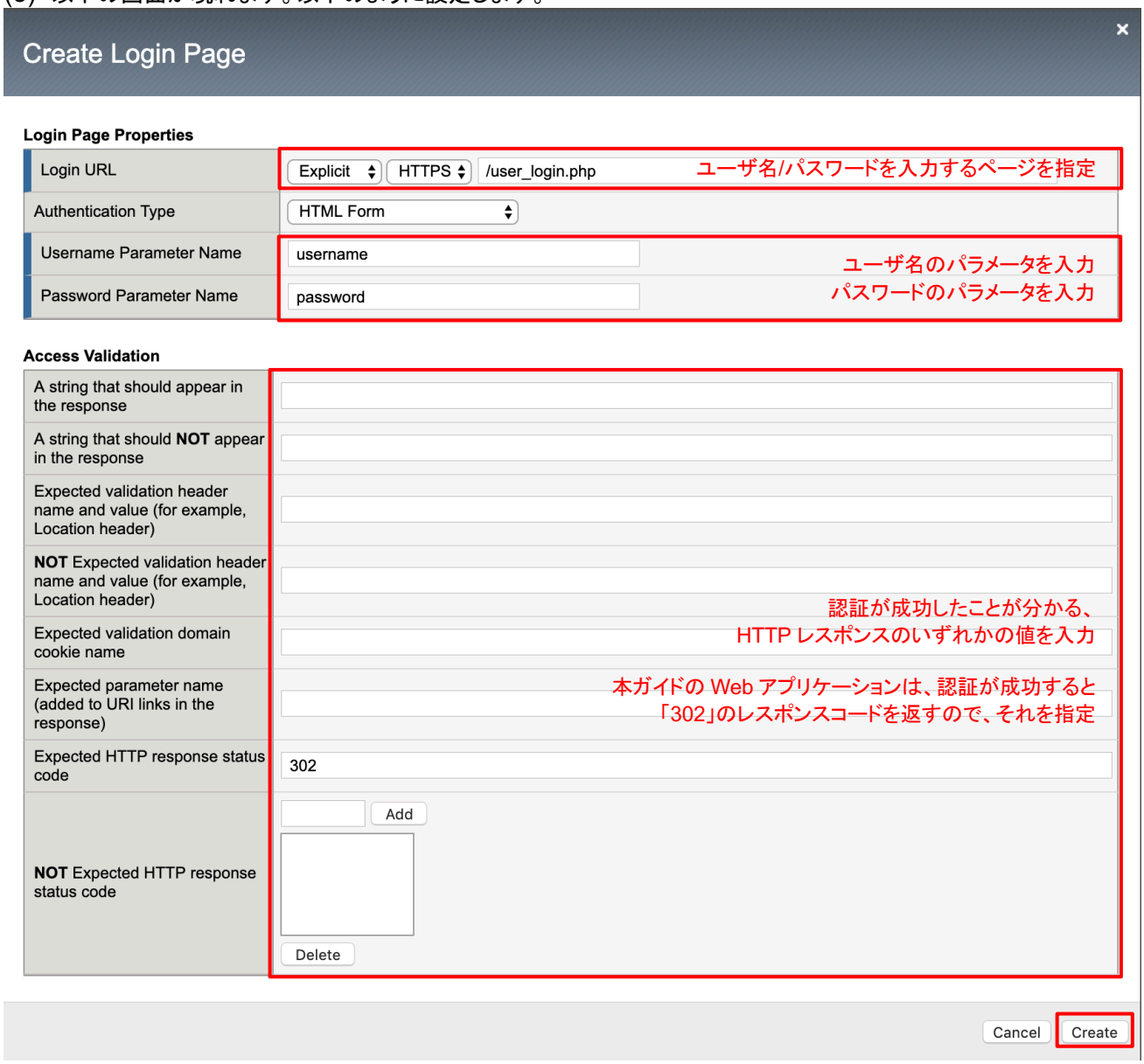
(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「Sessions and Logins」の「Brute Force: Maximum Login attempts are exceeded」を探します。
以下のように、Alarm/Block を有効となっていることを確認します。



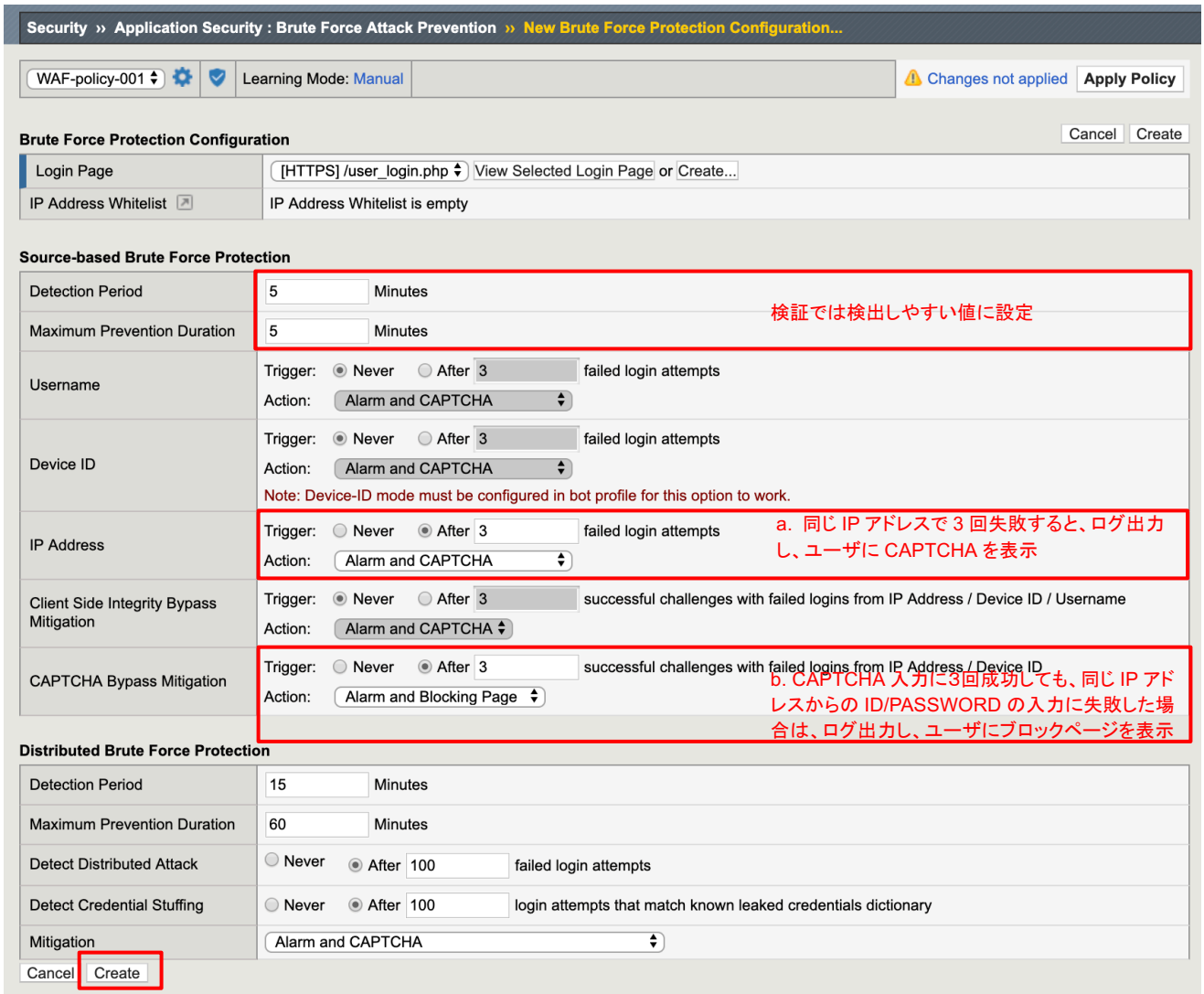
(2) 「Security」→「Application Security」→「Brute Force Attack Prevention」で表示された画面の右上にある「Create」ボタンを押すと、以下の画面が現れます。更に Login Page の「Create」ボタンを押します。



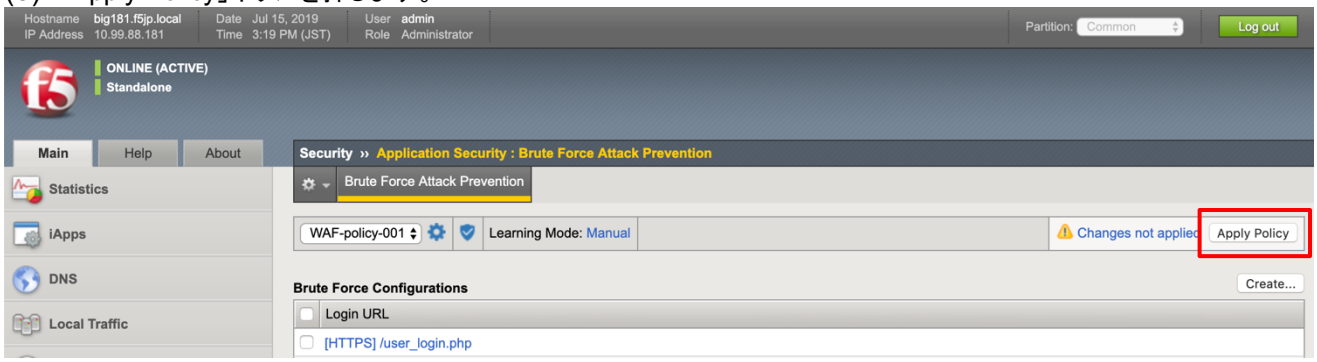
(3) 以下の画面が現れます。以下のように設定します。



(4) 上記画面で「Create」ボタンを押すと、以下の画面に戻ります。
Source-based Brute Force Protection の値を必要に応じて変更します。
下記のように設定し、「Create」ボタンを押します。（下記設定は、検出しやすい値にしています。）

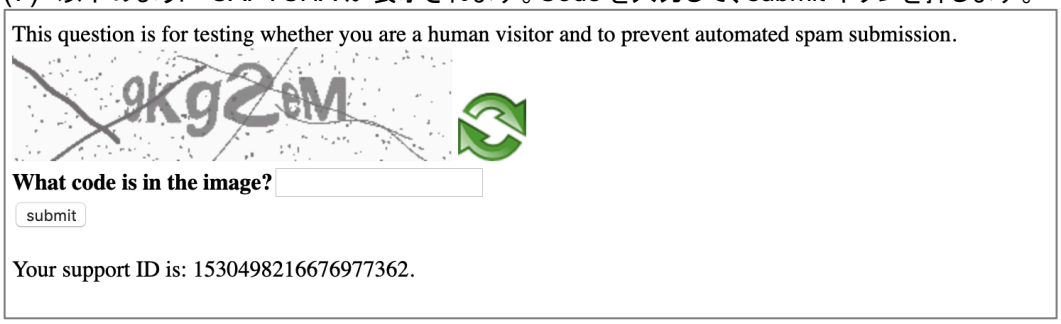


(5) 「Apply Policy」ボタンを押します。



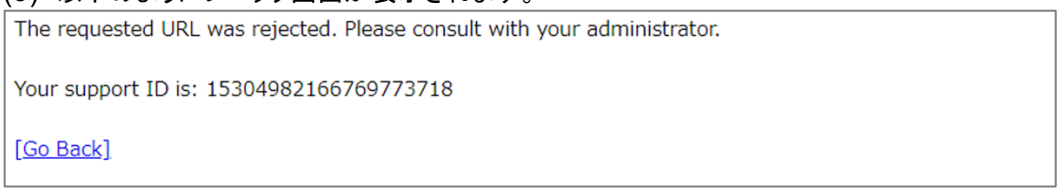
(6) 登録したログインページにアクセスし、クライアント PC から、ユーザ名のみ入力し、「クエリ送信」ボタンを押します。
これを 3 回以上繰り返します。

(7) 以下のように CAPTCHA が表示されます。Code を入力して、submit ボタンを押します。



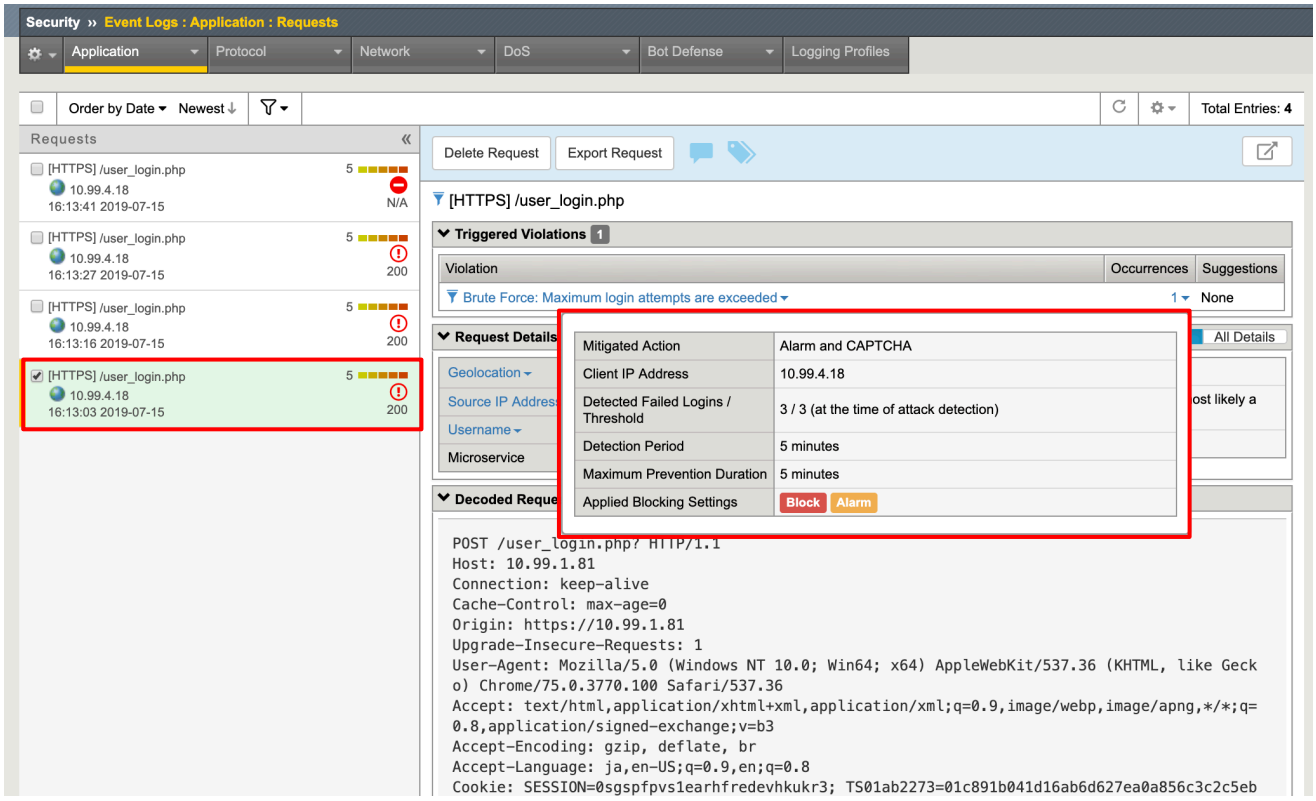
(8) 更に 3 回以上、パスワード入力を誤る操作と CAPTHCHA 対応を返します。

(9) 以下のようにブロック画面が表示されます。

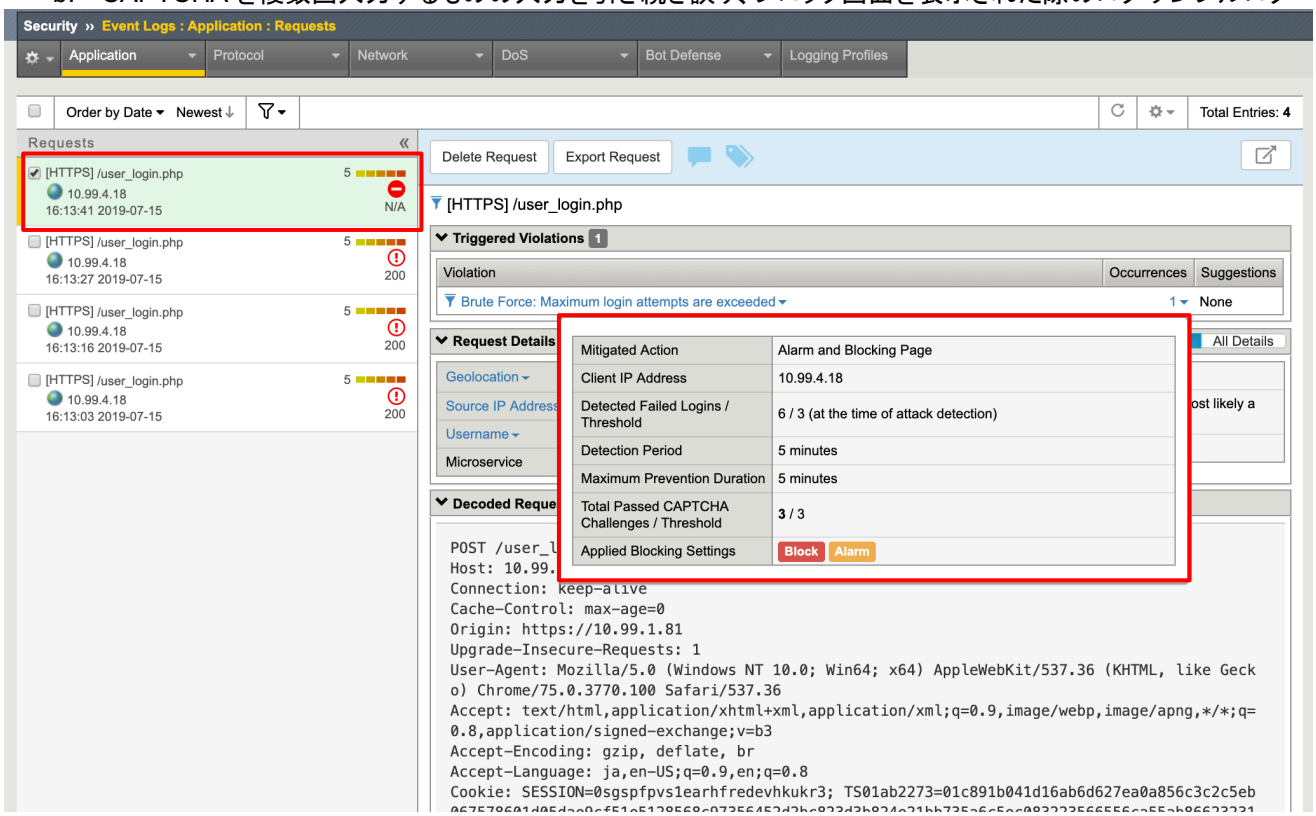


(10)「Security」→「Event Logs」→「Application」→「Requests」で、Brute Force 攻撃の検知の動作を確認します。

a. 同じ IP アドレスからの接続で入力を連続して誤り、CAPTCHA 表示をされた際のログサンプルログ



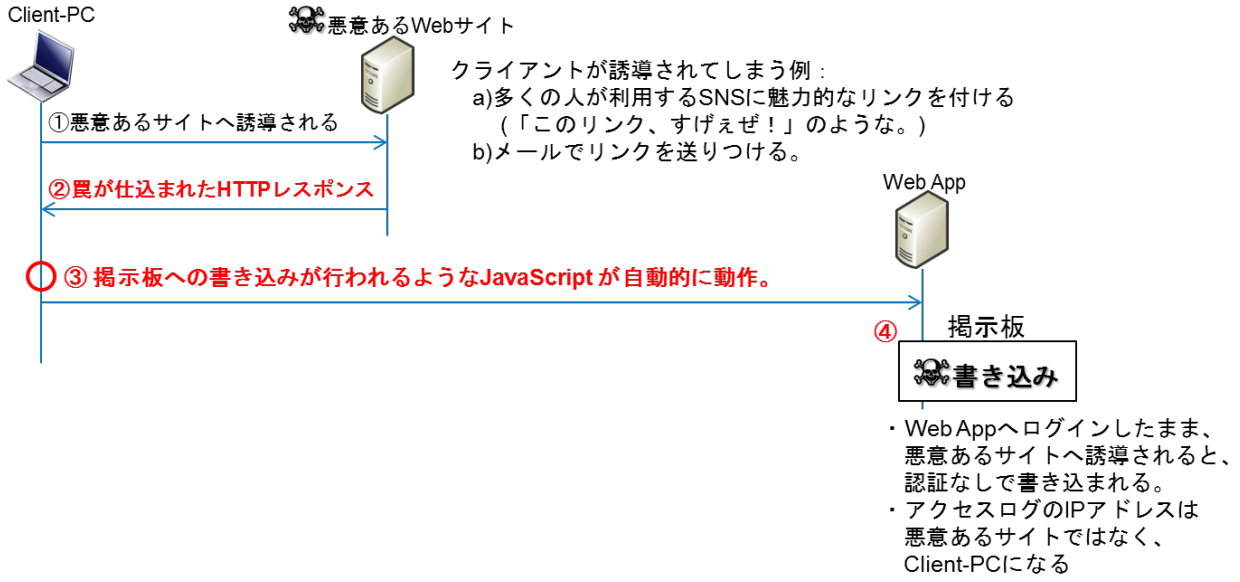
b. CAPTCHA を複数回入力するものの入力を引き続き誤り、ブロック画面を表示された際のログサンプルログ



9.4. CSRF (Cross Site Request Forgery) への対策

9.4.1. CSRF 攻撃とは

CSRF は以下のような攻撃です。



① クライアントが何らかの形で悪意ある Web サイトへ誘導されてしまう。
(ア) SNS 上に、魅力的な文面が書かれたリンク
(イ) 不正なメールのリンク

② 罠（悪意ある JavaScript 等）が仕込まれた HTTP レスポンスを受け取ってしまう。

③ その HTTP レスポンスを受け取ったと同時に、掲示板への書き込みが行われる JavaScript が動作する。

④ Web サイトに、意図しない書込みが行われる。

CSRF は、Web アプリケーションから見ると「正当なユーザからの HTTP リクエスト」と「悪意あるサイトによって作られた HTTP リクエスト」の区別がつかないため、シグネチャによる防御が難しい攻撃です。

よって、「正当なユーザからの HTTP リクエスト」と「悪意あるサイトによって作られた HTTP リクエスト」を区別できる方法を提供できれば、この攻撃を防御することができる、といえます。

Adv.WAF の CSRF 対策イメージ:

例えば、以下のようなリンクが実 Web サーバからの HTTP レスポンスの Body に存在していた、とします。

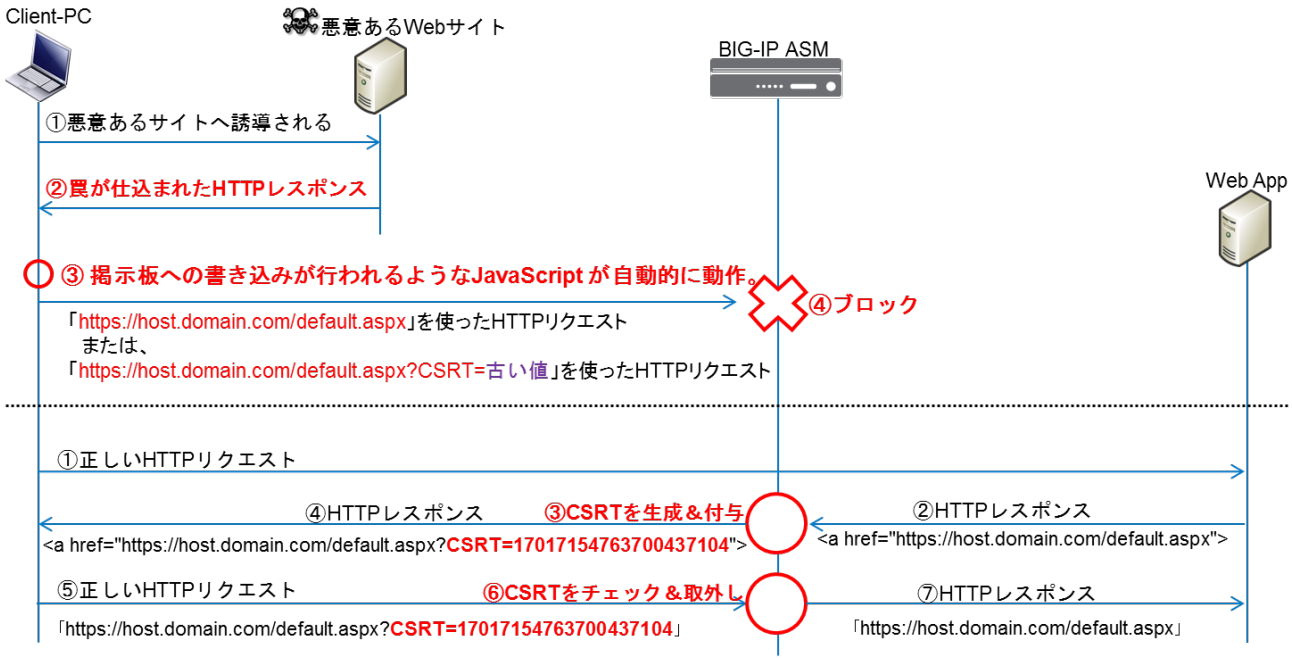
<a href="https://host.domain.com/default.aspx">・・・・(a)

Adv.WAF の CSRF プロテクト機能を有効にすることで、Adv.WAF はこの実 Web サーバからのリンクに、以下のような形で動的な値を Query String として加えます。

<a href="https://host.domain.com/default.aspx?CSRT=17017154763700437104">・・・・(b)

正当なユーザの HTTP リクエストの URI は、動的に変化する CSRT を持つ(b)を使います。

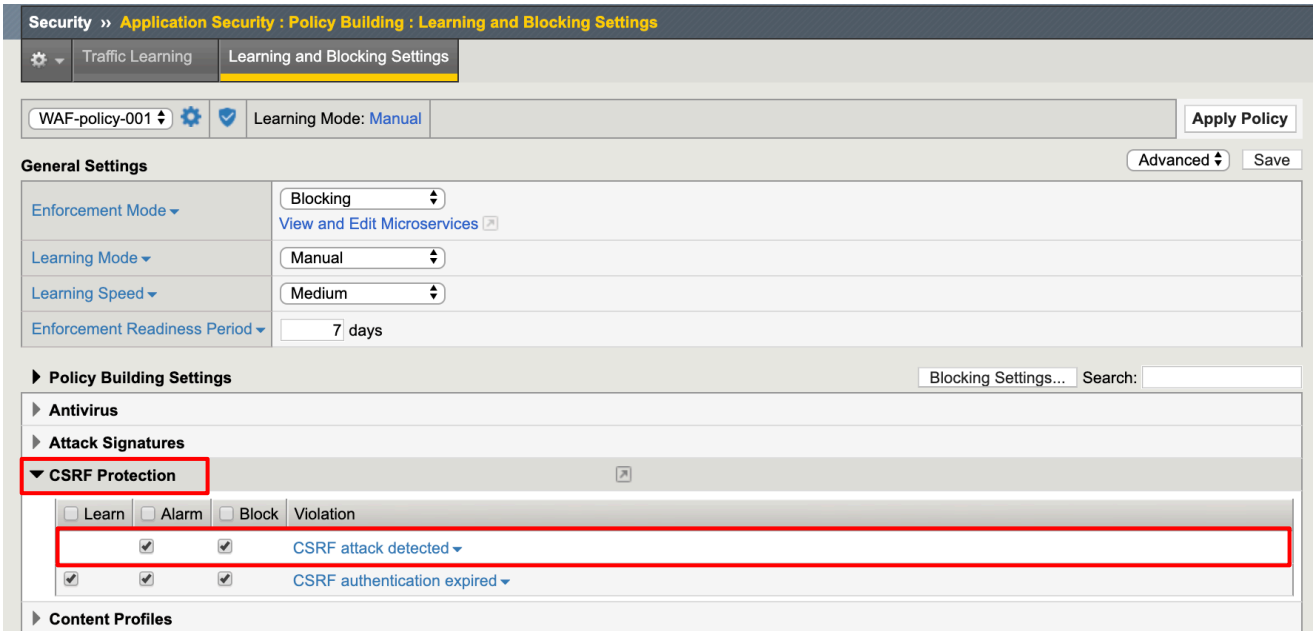
悪意あるサイトによる HTTP リクエストは、(a)を利用する、または、攻撃者が以前に取得した、古い値を持つ(b)を使うので、正当なユーザの HTTP リクエストと区別することができます。



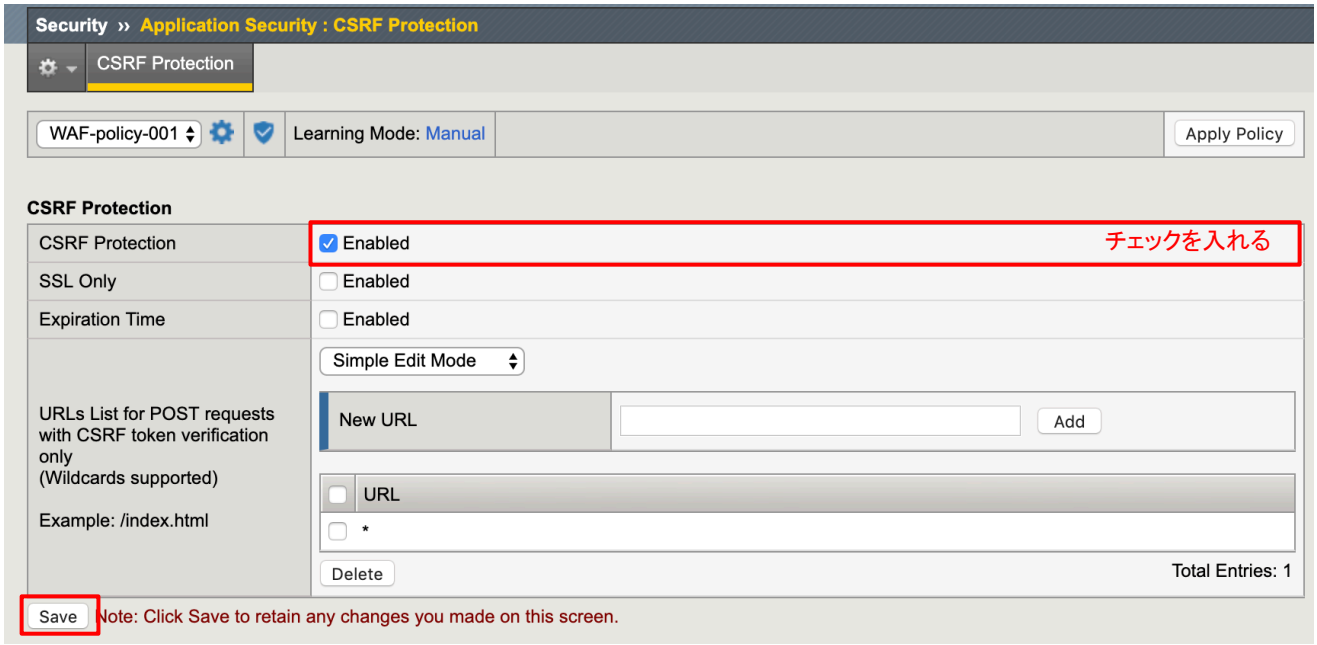
（厳密には、Adv.WAF が HTTP レスポンスに埋め込んだ JavaScript によって、上記の CSRT 値を付与する方式を取っています。）

9.4.2. CSRF 対策の設定

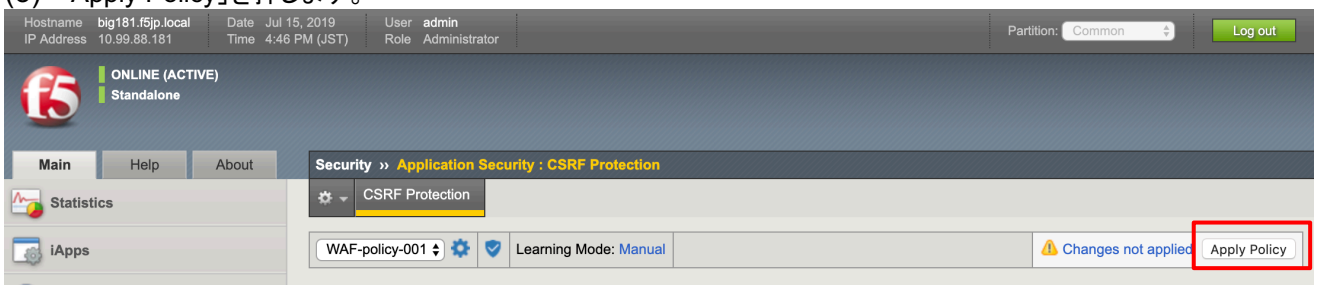
(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「CSRF Protection」を探します。
以下のように、Alarm/Block を有効になっていることを確認します。



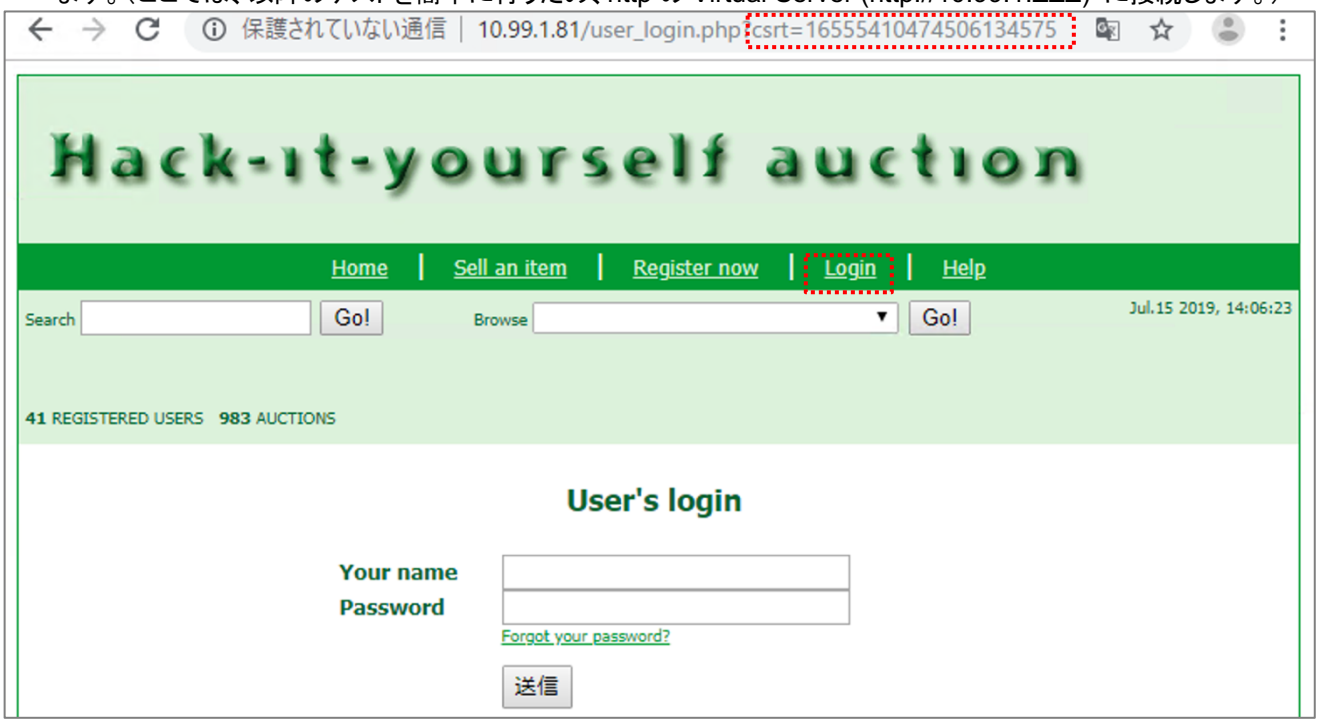
(2) 「Security」→「Application Security」→「CSRF Protection」で表示された画面で、CSRF Protection にチェックをいれ、Save ボタンを押します。



(3) 「Apply Policy」を押します。

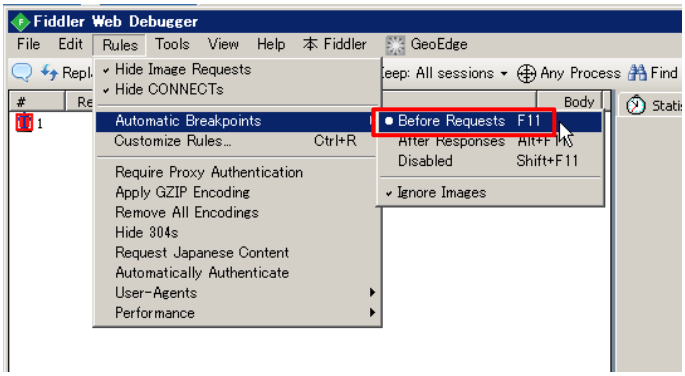


(4) 例えば、「Login」をクリックしてみると、下図のように、csrt 値が Query String として付与されていることが分かります。（ここでは、以降のテストを簡単に行うため、http の Virtual Server (http://10.99.1.ZZZ) に接続します。）



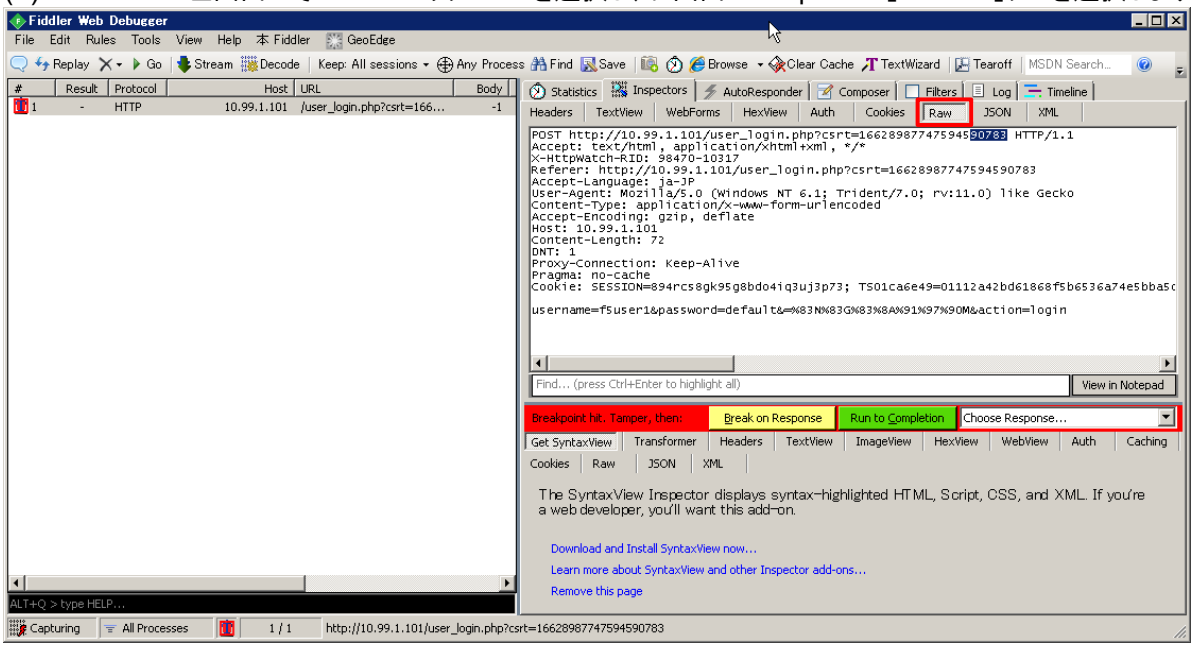
(5) クライアントのブラウザから攻撃を模擬するために、Fiddler2 を利用します。
Fiddler2 を起動し、「Rules」→「Automatic Breakpoints」→「Before Requests」を選択（有効に）します。

(6) （https 通信の場合、「Tools」→「Options」の「HTTPS」タブにて、「Capture HTTPS CONNECTs」と「Decrypt HTTPS traffic」を有効にして、Fiddler が作成するルート証明書を許可し、その証明書をブラウザにインポートする必要があります。）

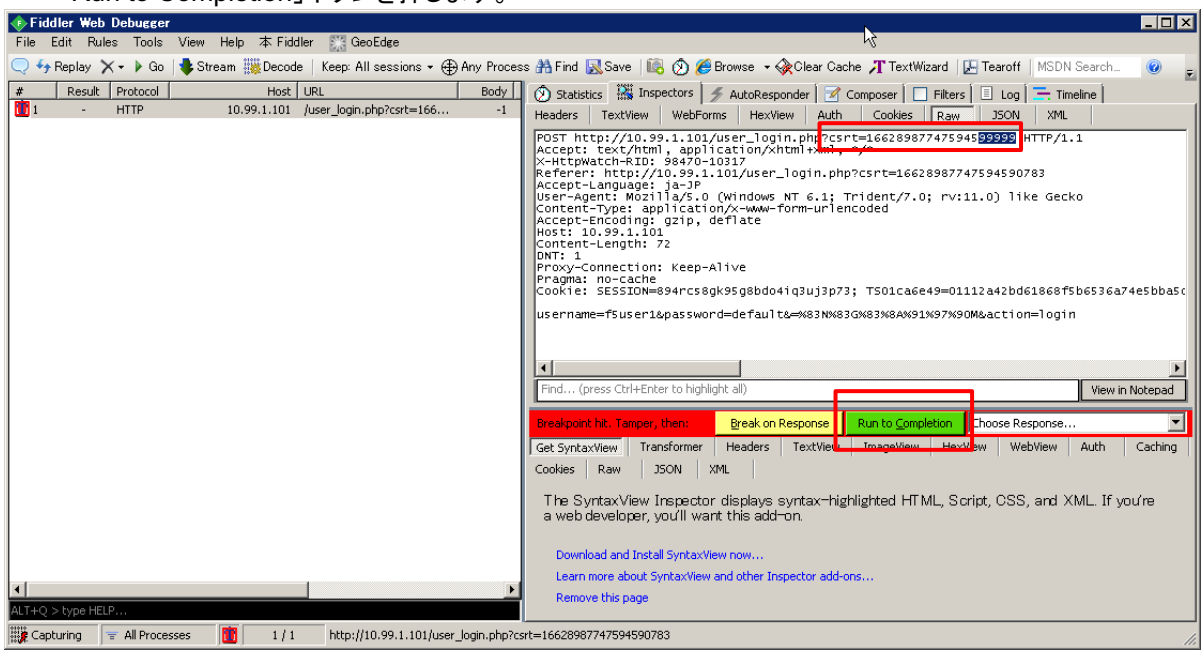


(7) ここでは http の Virtual Server にアクセスし、ブラウザに正当なユーザ名（f5suerX）とパスワードを入力し、「クエリ送信」ボタンを押します。

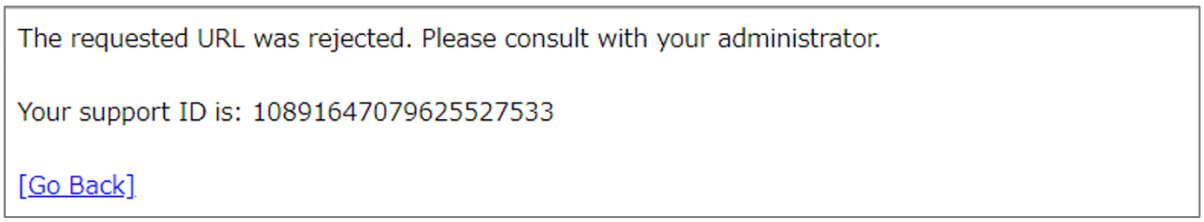
(8) Fiddler2 がこの HTTP リクエストをインターセプトするので、先に進みません。
(9) Fiddler2 の左画面でその HTTP リクエストを選択し、右画面で「Inspectors」→「Raw」タブを選択します。



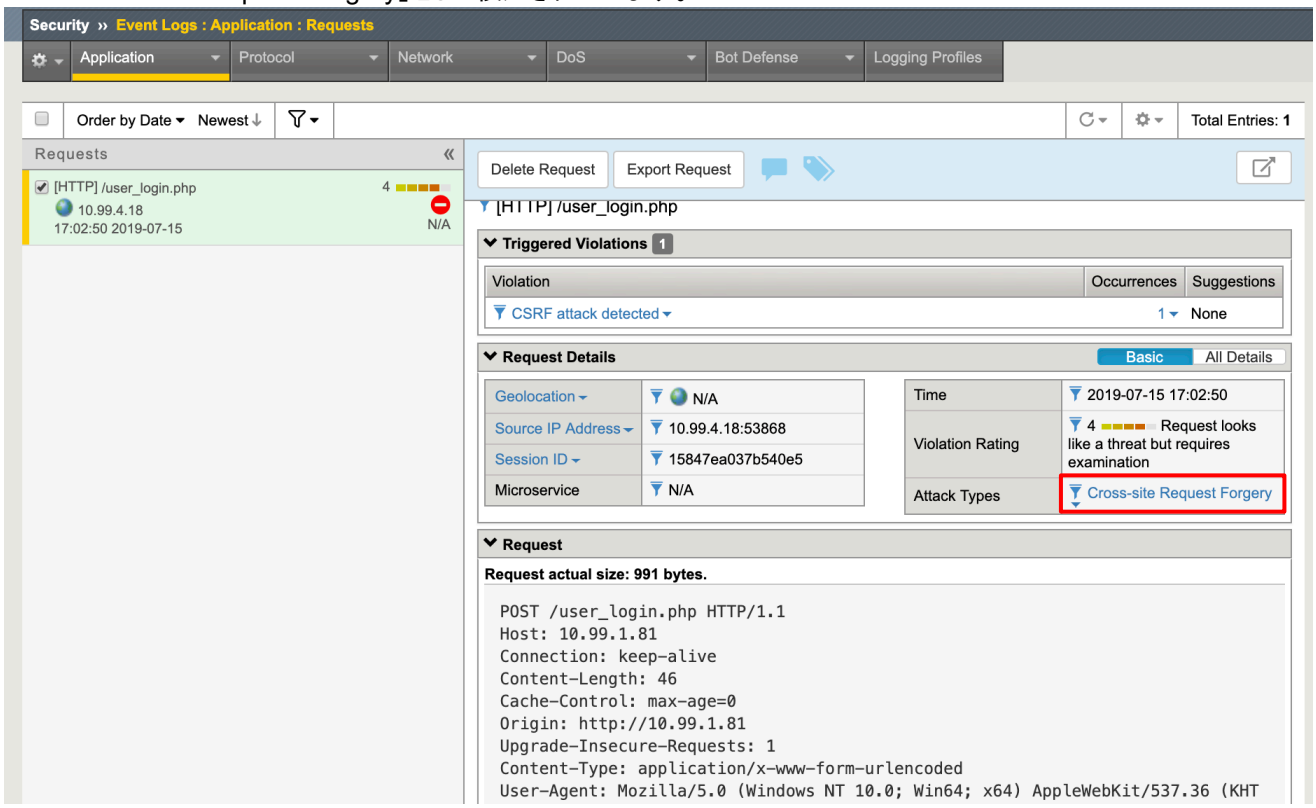
(10)CSRT の値の末尾 5 桁を以下のように「99999」に書き換えてみます。
　＝古い CSRT 値を使って攻撃をしてきている、という想定です。
　「Run to Completion」ボタンを押します。



(11)Adv.WAF が期待する CSRT 値ではない HTTP リクエストが到達したため、攻撃として検知します。



(12) Fiddler2 を終了し、「Security」→「Event Logs」→「Application」→「Requests」でログを確認します。
　「Cross-site Request Forgery」として検知されています。



9.5. Threat Campaigns シグネチャ ＜ご参考＞

※F5 ハンズオンでは実施しません

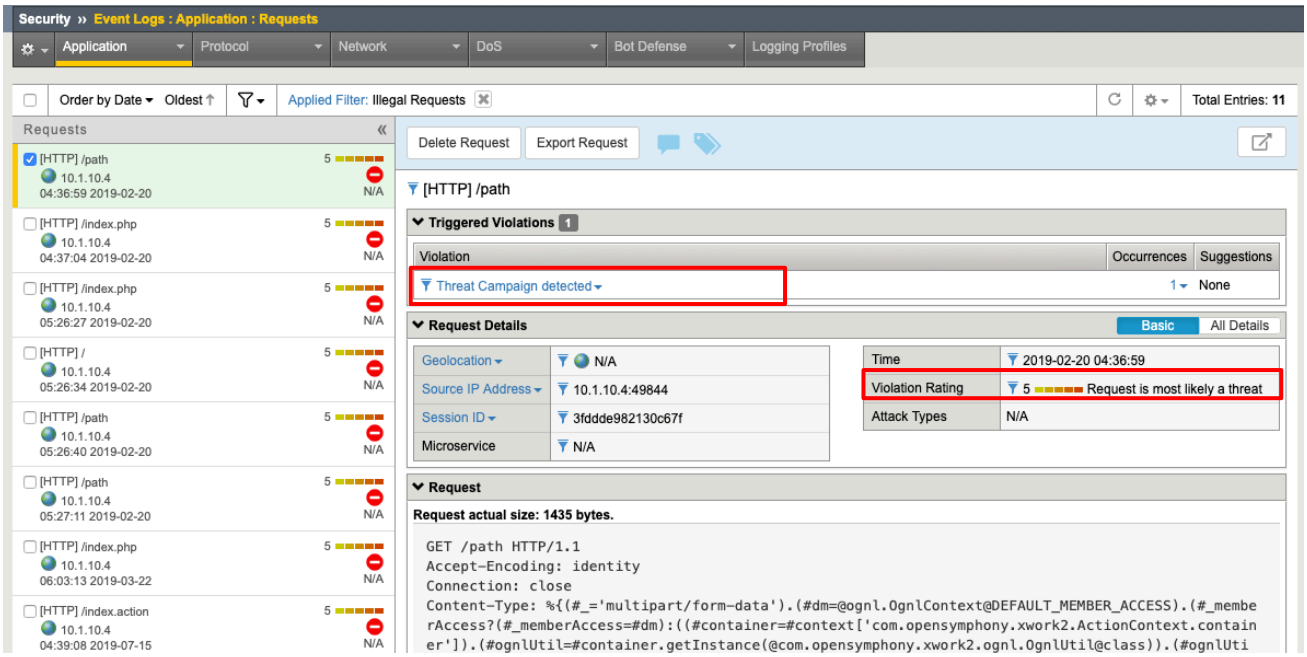
Web アプリケーションの脅威対策を行う上で、実際の攻撃とフォールスポジティブを見極めることに直面することがあります。シグネチャや WAF の様々な機能によって、既知の脆弱性に対する攻撃リクエストを評価すると同時に、悪意のないリクエストも評価されるため、管理者は WAF で検知されたアラートに対し、正常なリクエストか悪意のあるリクエストか判断を迫られることがあります。

この問題は、検査するデータがシングルポイントであることにあります。もっと多くの条件を元にリクエストを検査することによって、この問題を解決することができることがあります。F5 はこの問題を解決する手法として、”Threat Campaigns シグネチャ“という独自のシグネチャを提供しています。Threat Campaigns シグネチャは、実際の攻撃キャンペーンを元に複数の条件をマッピングされたものになっています。

この項では、Threat Campaign の設定確認と、テキスト上にて攻撃サンプルログを確認します。
（注：Threat Campaigns はサブスクリプション形式のライセンスとなります。）

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面にて、Threat Campaigns の Alarm&Block 有効になっていることを確認します。

(2) 攻撃を受けたと仮定して、「Security」→「Event Logs」→「Application」→「Requests」でログを確認します。
下記は攻撃を受けたときのサンプルとなります。



上記のイベントログを見ると、Violation Rating が“5：Request is most likely a threat”となっていることが分かります。これは Threat Campagins シグネチャが実際の攻撃を元に作成されており、ほぼフォールスポジティブではないことを表しています。

9.6. 最終的なチューニング

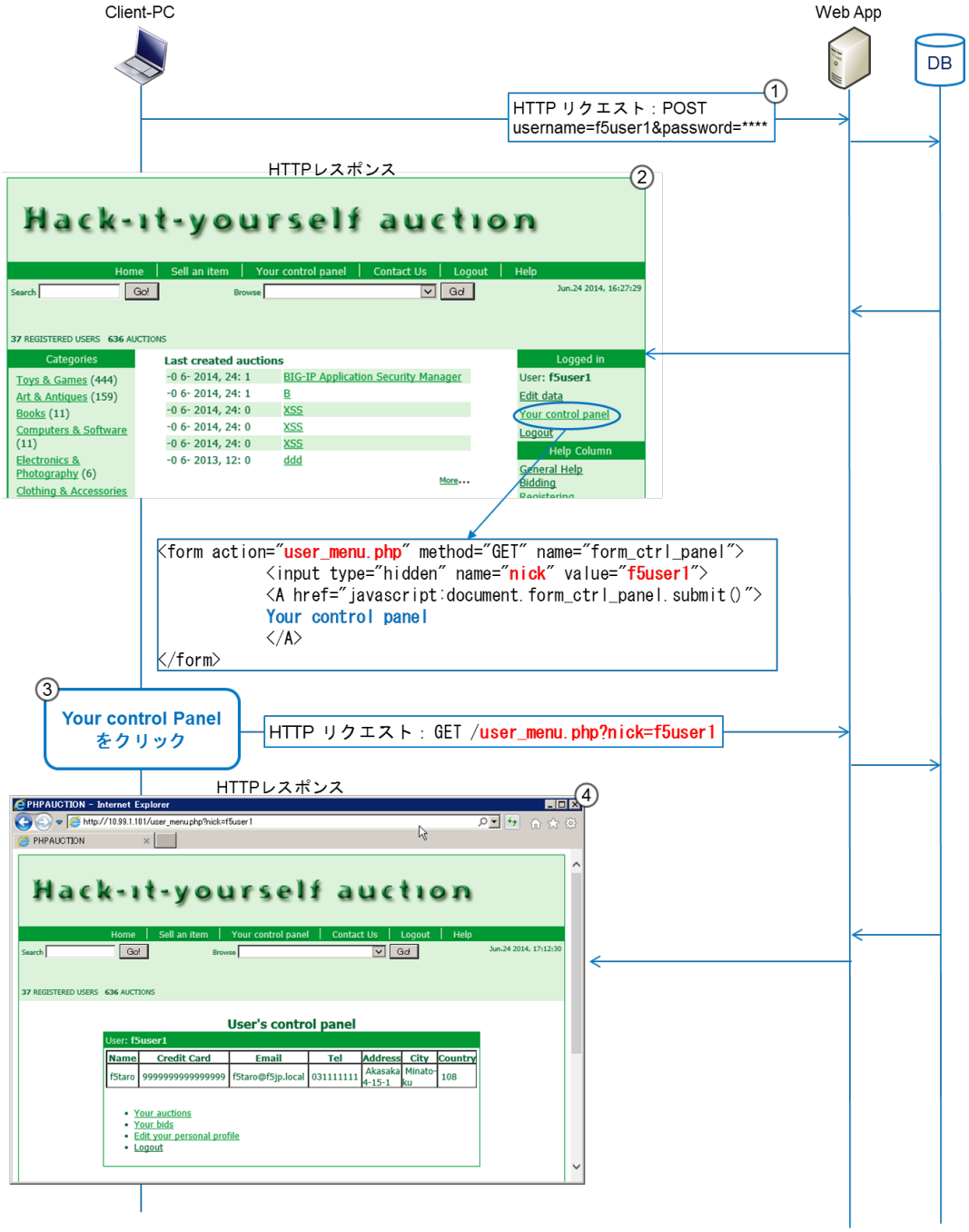
[Step1]～[Step4]では防御できない攻撃への対策として、2 つの攻撃とその設定をサンプルとして、解説します。

9.7. 例：パラメータ・タンパリング

パラメータ・タンパリング攻撃とは、サーバから提供されたパラメータ値を改ざんすることで、Web アプリケーションが期待しない動作を引き起こすような攻撃です。以下に、この攻撃例を示します。

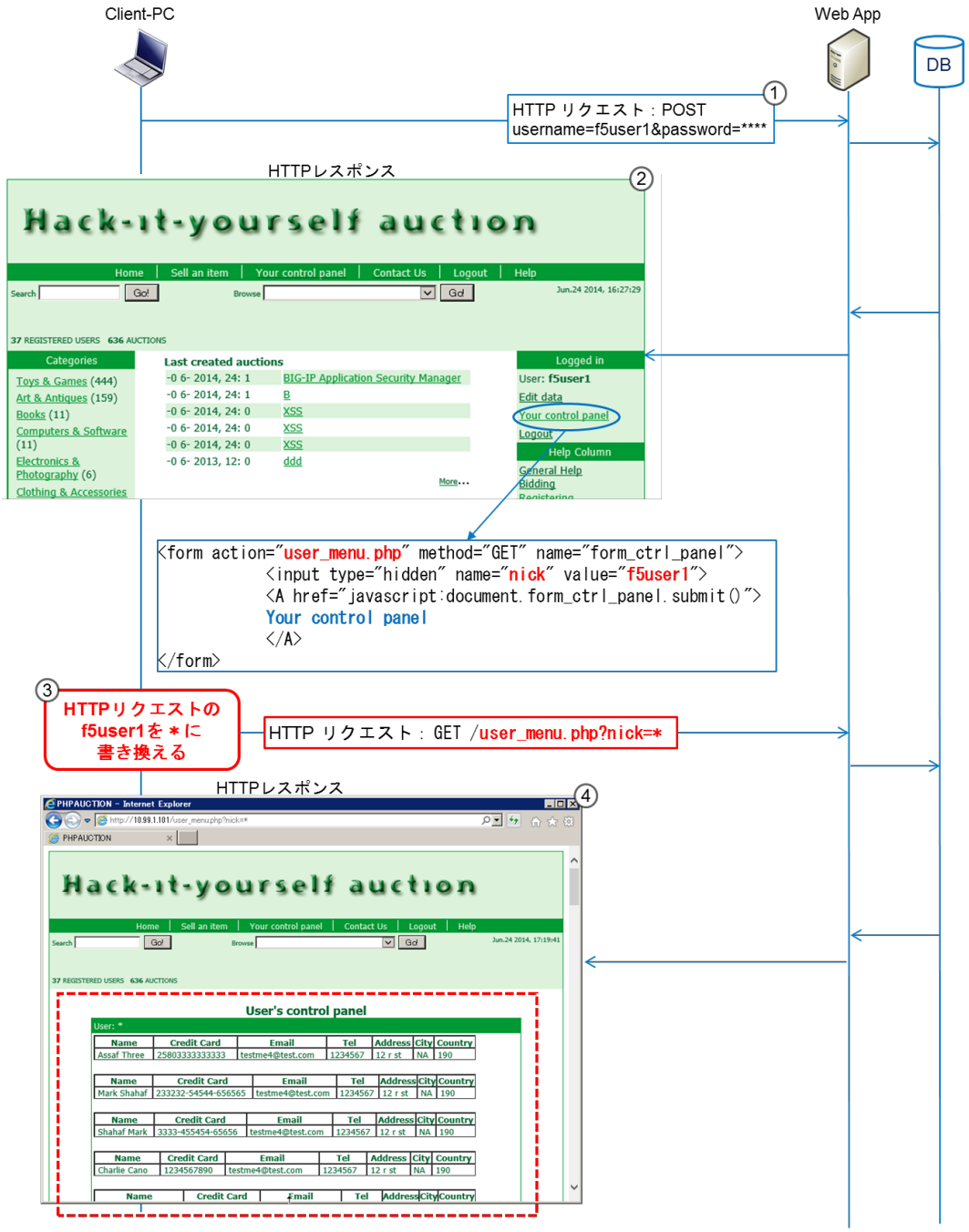
9.7.1. 攻撃例

(1) 正常な動作



① ユーザ名: f5userX でログイン。
② HTTP レスポンスページの「your control Panel」は、図中のような HTML と JavaScript で構成されている。
③ この「your control Panel」をクリックすると、「/user_menu.php?nick=f5userX」の GET リクエストが送信される。
④ f5userX の情報が記載されたページが HTTP レスポンスとして返される。

(2) パラメータ・タンパリング攻撃



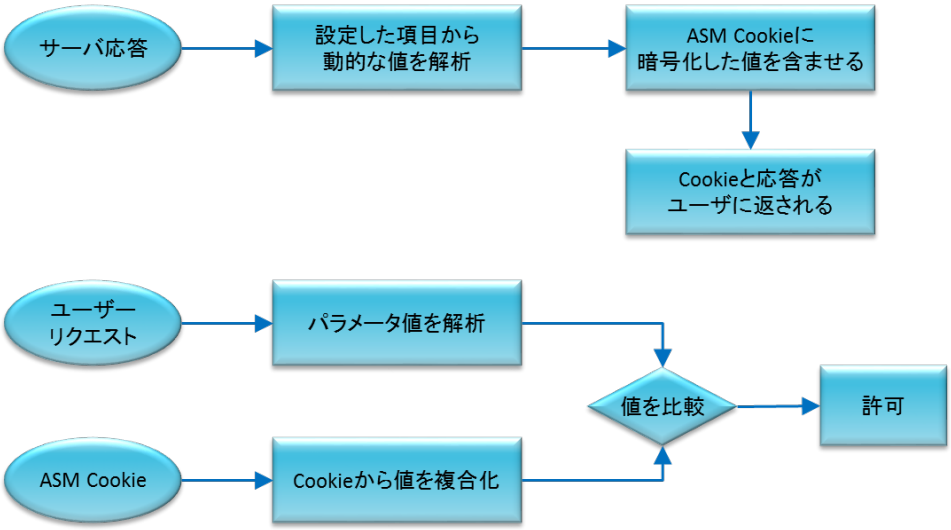
① ユーザ名: f5userX でログイン。
② HTTP レスポンスページの「your control Panel」は、図中のような HTML と JavaScript で構成されている。
③ この「your control Panel」をクリックすると、「/user_menu.php?nick=f5userX」となるところを、攻撃者は f5userX を＊に書き換えて送信。
④ 全ユーザの情報が記載されたページが HTTP レスポンスとして返されてしまう（ようなことがあり得る）。

次に進む前に、Fiddler2 を使って本攻撃を実施してみてください。

9.7.2. パラメータ・タンパリングの防御方法

この攻撃に対しては、Web アプリケーション側から提供されたパラメータ値が、後の HTTP リクエストで改ざんされていないかどうかをチェックすることで防御できます。

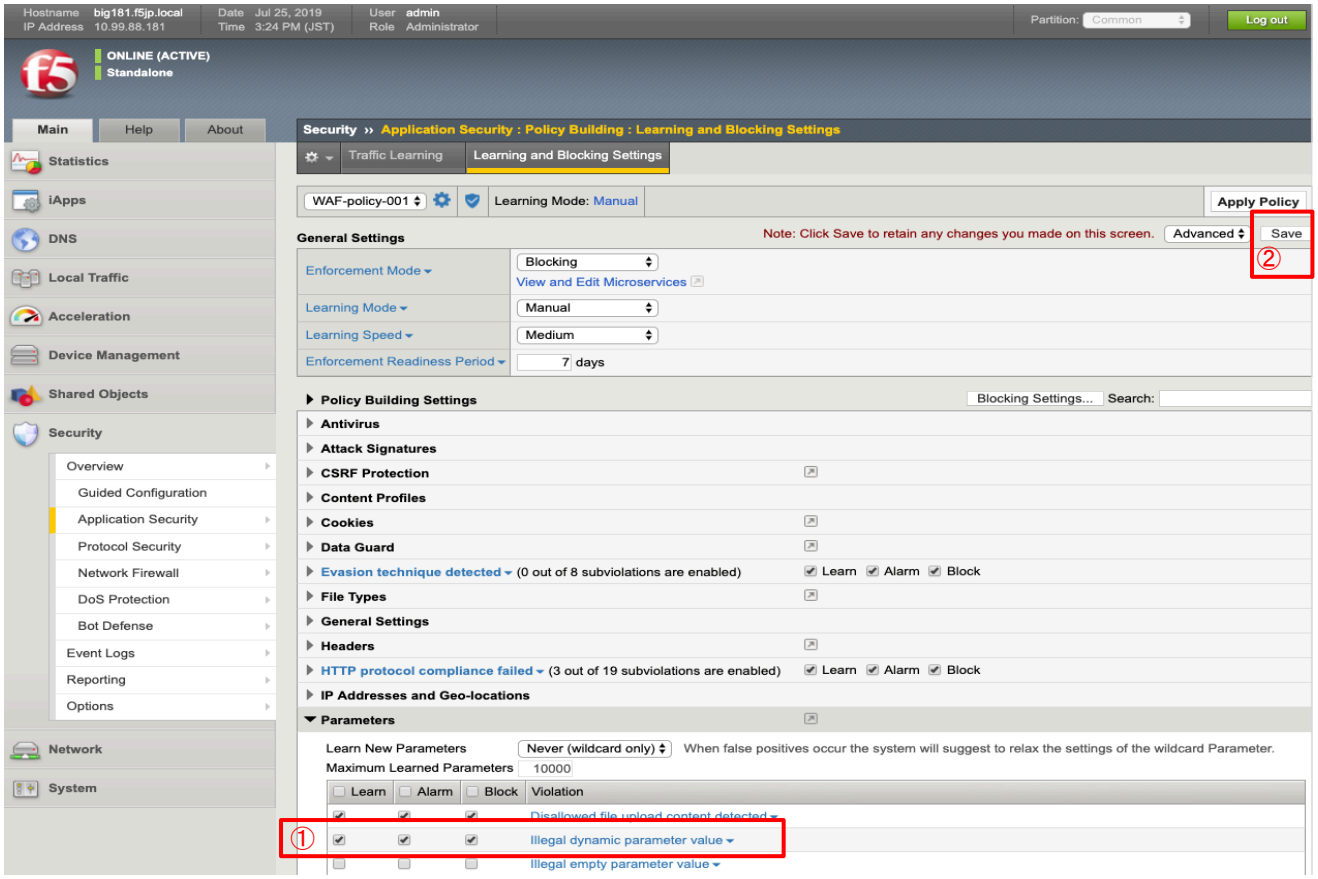
Adv.WAF は以下のようなフローにて、改ざんがなされていないかのチェックを行います。



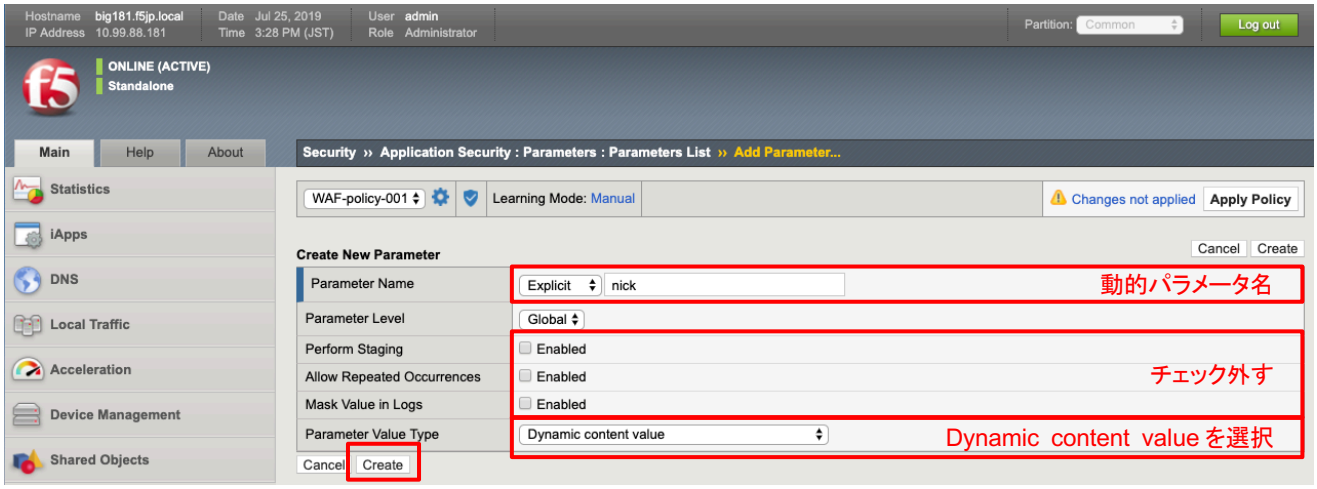
9.7.3. パラメータ・タンパリング対策の設定

このようなパラメータ・タンパリング攻撃を防ぐ Adv.WAF の設定を示します。

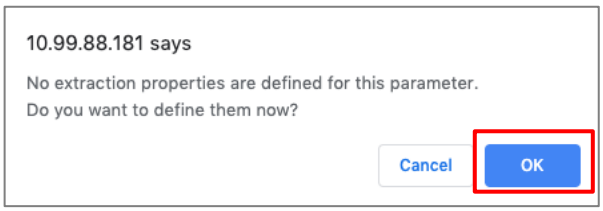
(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「Prameters」カテゴリの「illegal dynamic parameter value」を探します。
以下のように、Learn / Alarm/Block を有効にし、「Save」ボタンを押します。



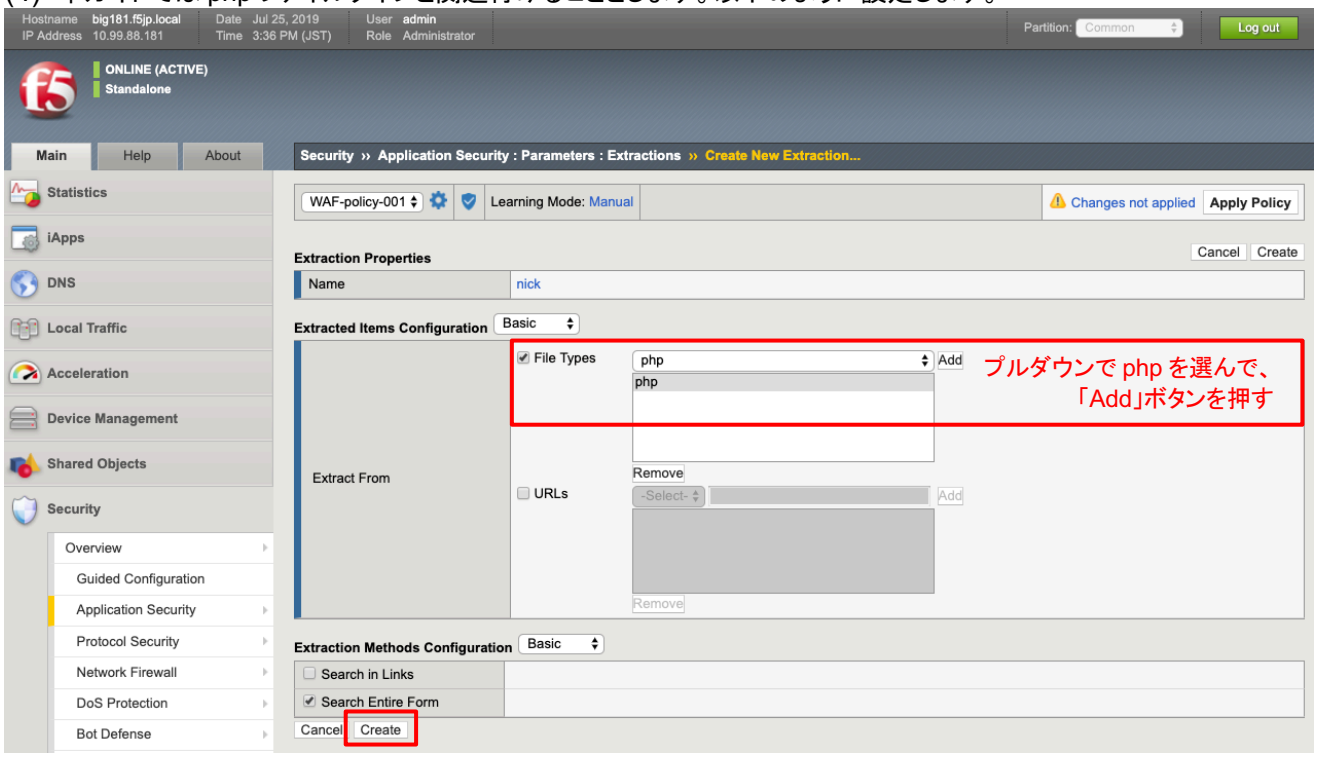
(2) 「Security」→「Application Security」→「Parameters」→「Parameters List」で表示された右側の「Create」ボタンで現れた以下の画面で、以下のように設定します。



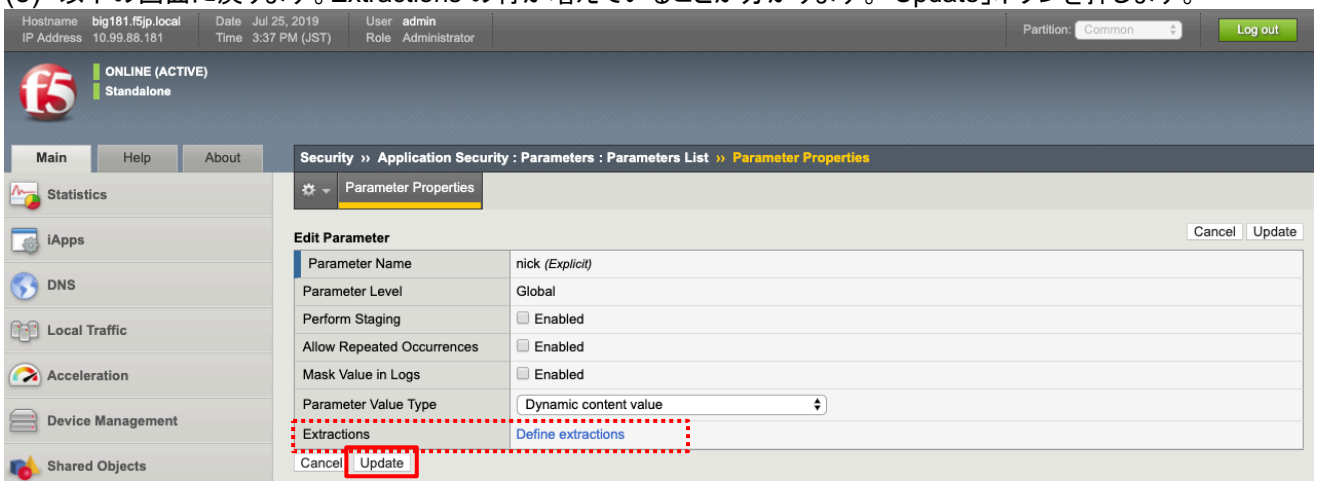
(3) この動的パラメータ（nick）と関連するファイルタイプ（または URL）の指定を促す画面が出ます。「OK」を押します。



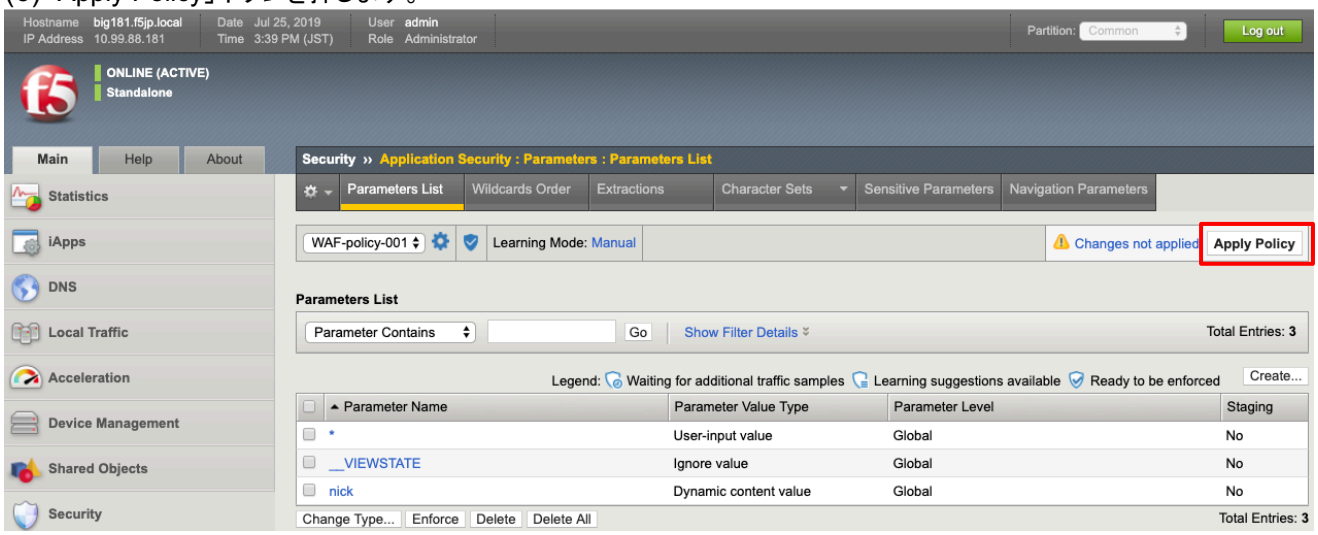
(4) 本ガイドでは php ファイルタイプと関連付けることとします。以下のように設定します。



(5) 以下の画面に戻ります。Extractions の行が増えていることが分かります。「Update」ボタンを押します。

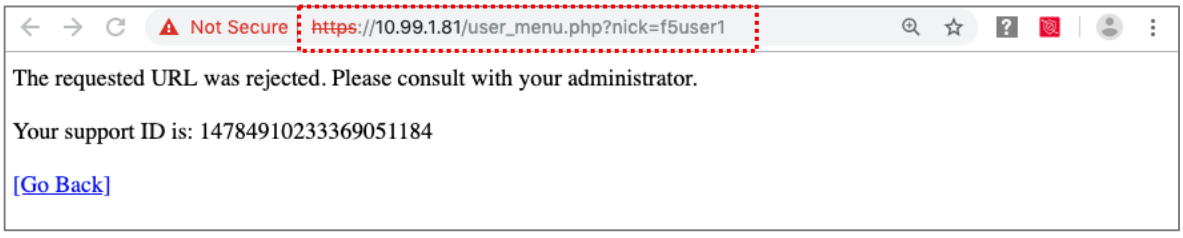


(6) Apply Policy」ボタンを押します。

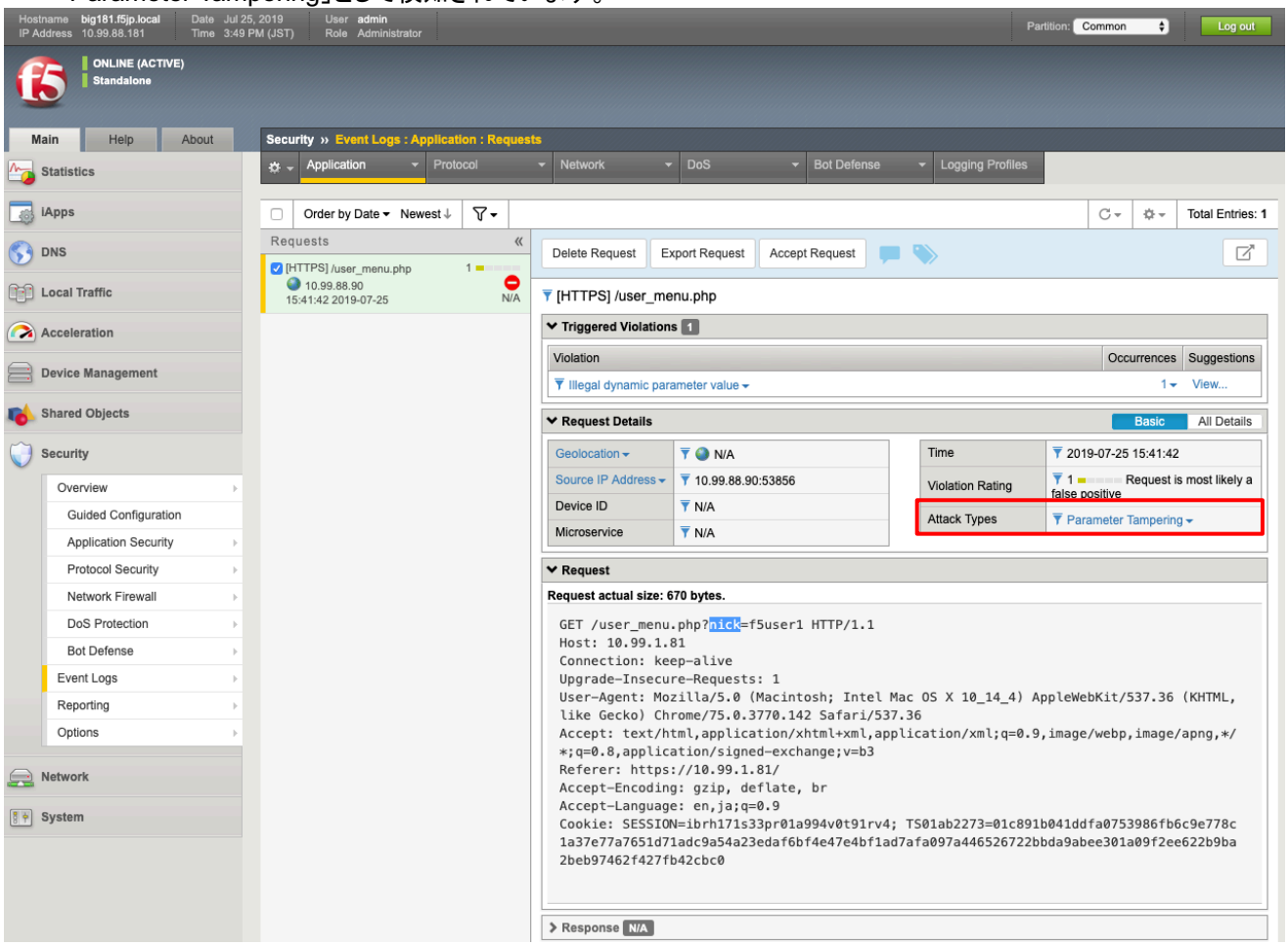


(7) クライアントのブラウザで、正当なユーザ（f5userX）でログインします。

(8) 「Your Control Panel」をクリックし、以下のように「nick=*」に書き換えて、Enter します。
以下のように攻撃として検知されます。



(9) 「Security」→「Event Logs」→「Application」→「Requests」でログを確認します。
「Parameter Tampering」として検知されています。



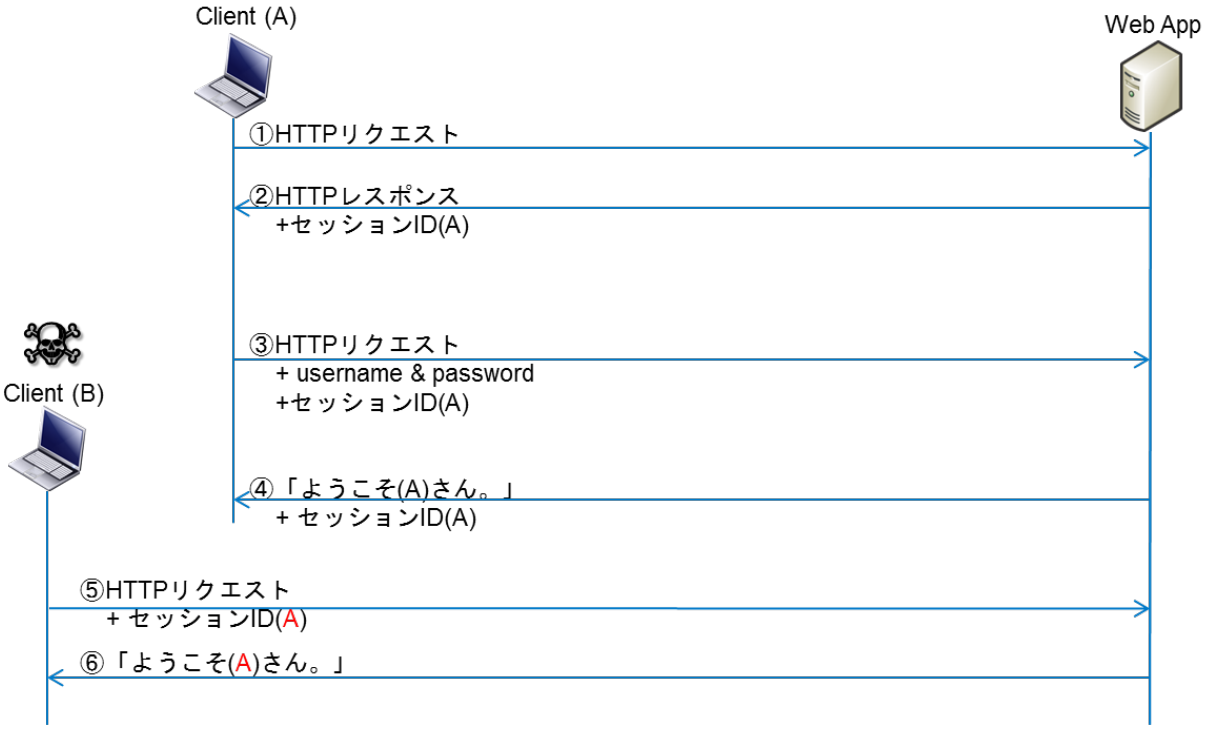
(10) 「Illegal dynamic parameter value」をクリックすると、以下の画面が現れます。
パラメータ:「nick」で検知していることが分かります。

Parameter Location	Query String
Parameter Level	Global
Parameter Name	nick
Parameter Value	f5user1
Applied Blocking Settings	Block Alarm Learn

9.8. 例：セッションハイジャック

他人のセッション ID（Cookie）を模倣することで、そのユーザとしてログインする、という攻撃です。
以下に、この攻撃例を示します。

9.8.1. 攻撃例



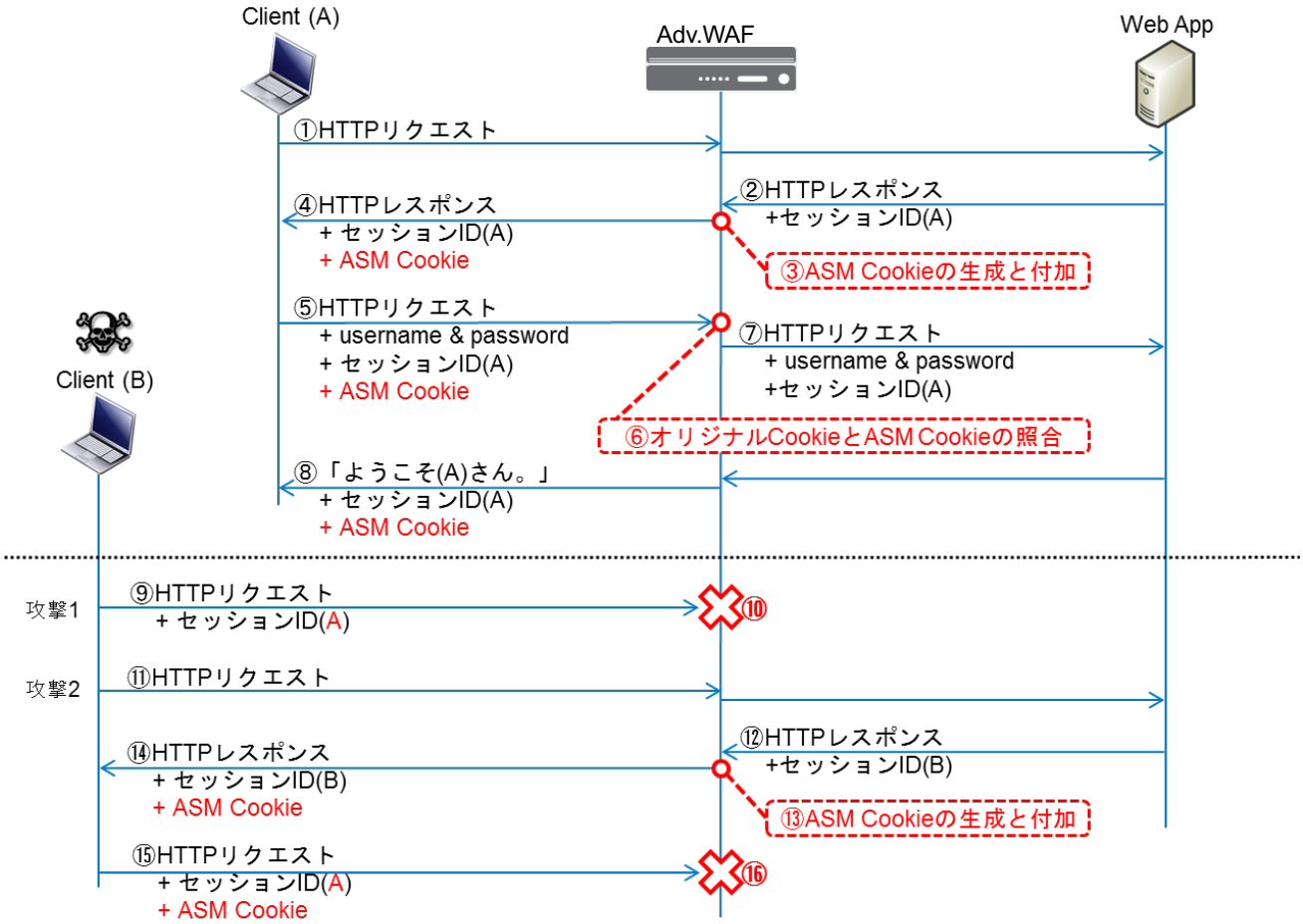
① Client(A)が Web App に HTTP リクエストを送る。
② Web App はセッション ID(A)をつけて、認証フォーム画面の HTTP レスポンスを送る。
③ Client(A)は、ユーザ名とパスワードを入力＝POST リクエストを送る。
④ Client(A)用の画面が表示される。

⑤ Client(B)は、何らかの方法でセッション ID(A)を生成して、HTTP リクエストを送る。
⑥ ログイン情報を入力することなく、Client(A)の画面を見ることができる。

9.8.2. セッションハイジャックの防御方法

Adv.WAF では、Web アプリケーションから払い出されたセッション ID(Cookie)のハッシュ値を計算し、それを Cookie (ASM Cookie) として付与します。

クライアントから戻ってきたそのセッション ID(Cookie)のハッシュ値を計算し、ASM Cookie 値と同じであれば、改ざんされていない、と判断します。



① Client(A)が Web App に HTTP リクエストを送る。
② Web App はセッション ID(A)をつけて、認証フォーム画面の HTTP レスポンスを送る。
③ Adv.WAF は、セッション ID(A)のハッシュ値を計算。それを ASM Cookie として HTTP レスポンスに付与する。
④ Client(A)は、ASM Cookie が付与された HTTP レスポンスを受け取る。
⑤ Client(A)は、ユーザ名とパスワードを入力＝POST リクエストを送る。
⑥ Adv.WAF は、セッション ID(A)のハッシュ値を計算。それを ASM Cookie 値と比較する。
⑦ 比較結果が同じであれば、改ざんされていないと判断し、Web App へ HTTP リクエストを転送する。
⑧ Client(A)用の画面が表示される。

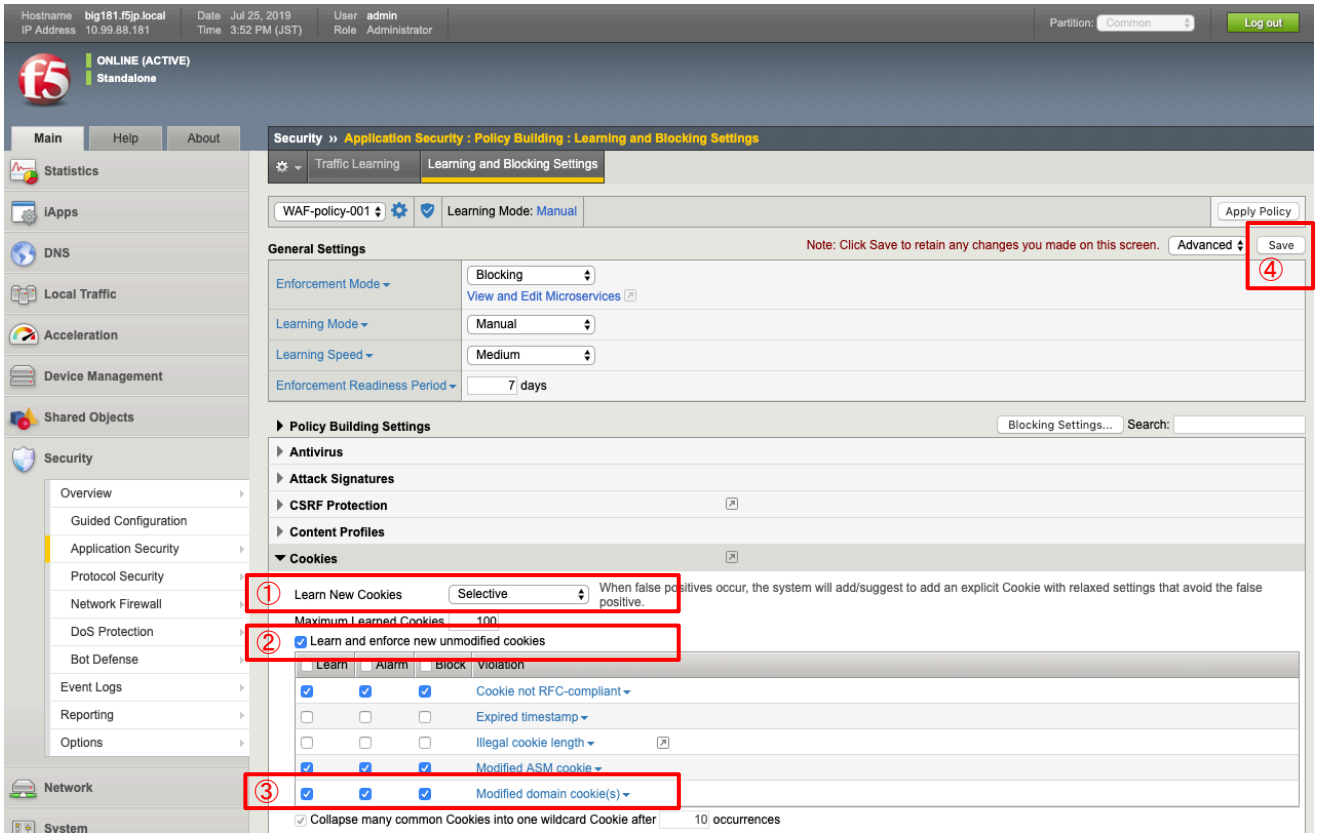
⑨ [攻撃 1]Client(B)は、何らかの方法でセッション ID(A)を生成して、HTTP リクエストを送る。
⑩ ASM Cookie 値を持たないため、Adv.WAF で拒否される。

⑪ [攻撃 2]Client(B)が HTTP リクエストを送る。
⑫ Web App はセッション ID(B)をつけて、認証フォーム画面の HTTP レスポンスを送る。
⑬ Adv.WAF は、セッション ID(B)のハッシュ値を計算。それを ASM Cookie として HTTP レスポンスに付与する。
⑭ Client(B)は、ASM Cookie が付与された HTTP レスポンスを受け取る。
⑮ Client(B)は、セッション ID(B)をセッション ID(A)に書き換えて、HTTP リクエストを送る。
⑯ セッション ID(A)のハッシュ値と ASM Cookie 値が異なるため、ブロックされる。

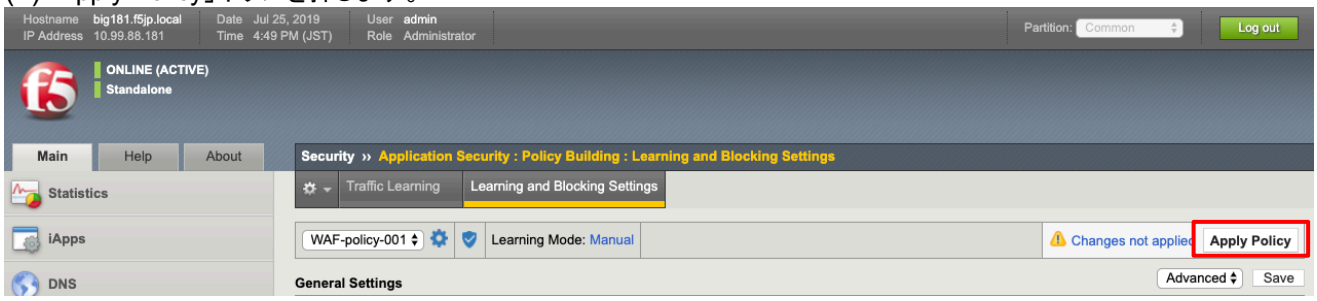
9.8.3. セッションハイジャック対策の設定

このようなセッションハイジャック攻撃を防御する設定を示します。

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」で表示された画面で、「Learn New Cookies」にて”Selective”を選択し、「Learn and enforce new unmodified cookies」と「Modified domain cookie」の Learn / Alarm/Block を有効にし、「Save」ボタンを押します。

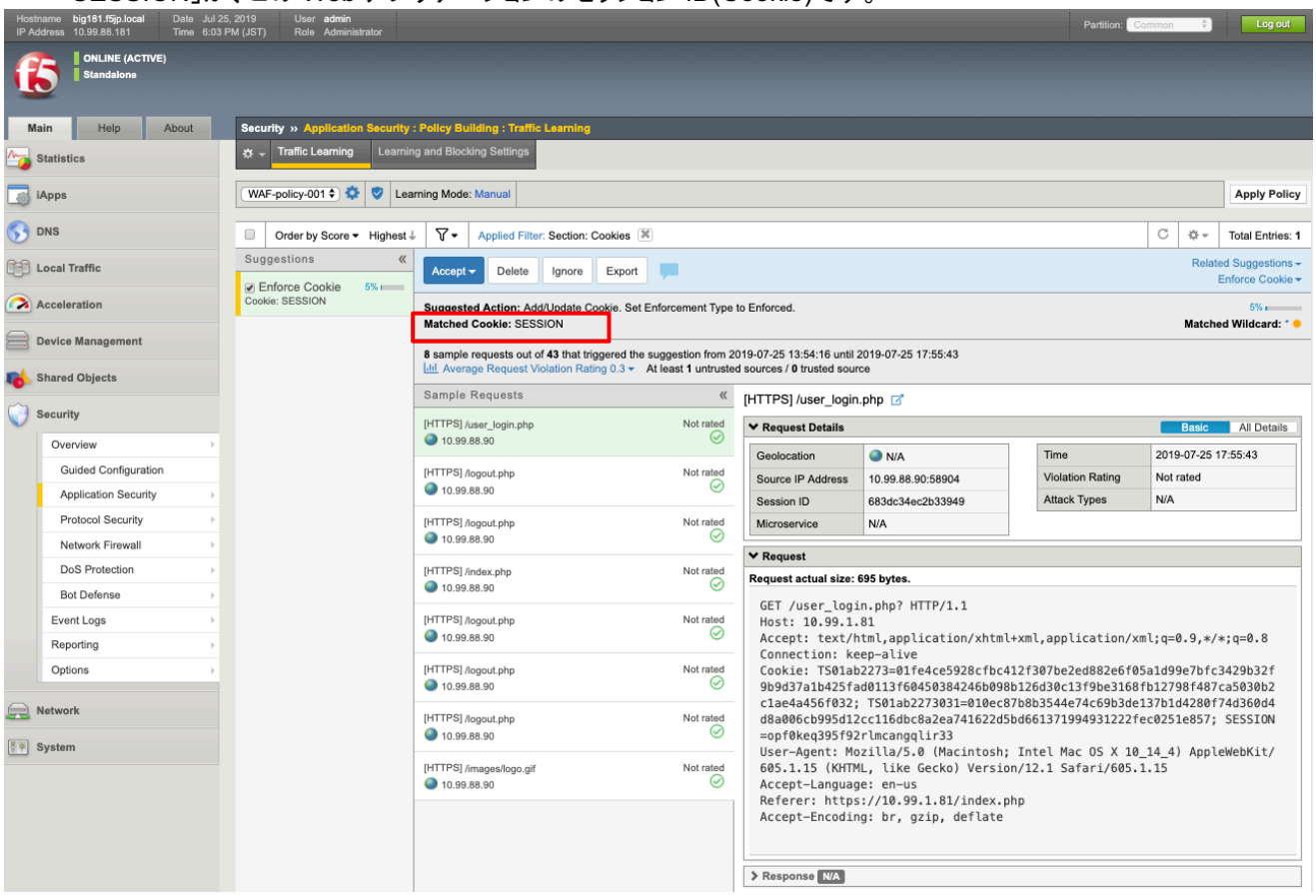


(2) Apply Policy」ボタンを押します。

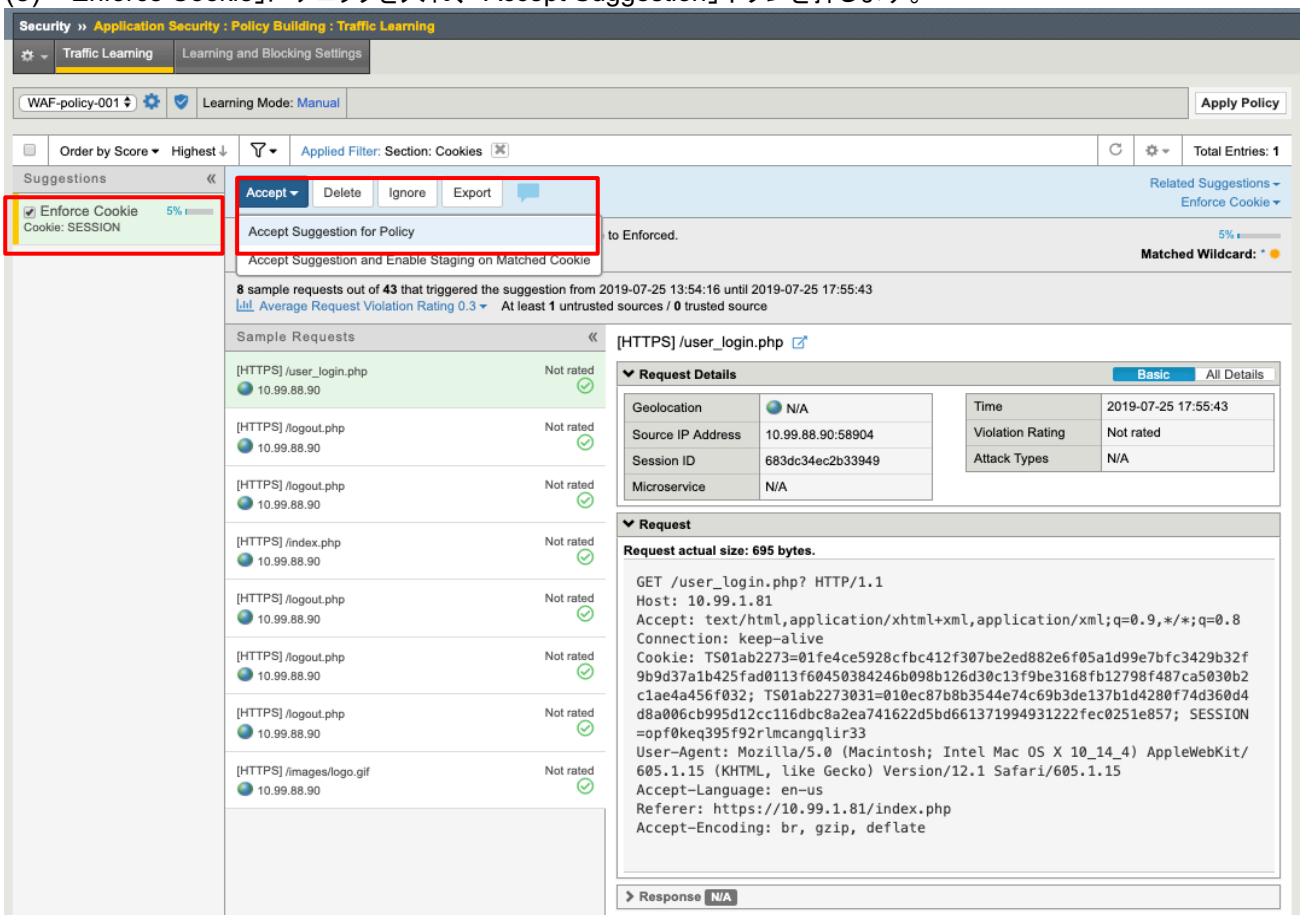


(3) クライアントのブラウザで、ユーザ（f5userX）ログインします。

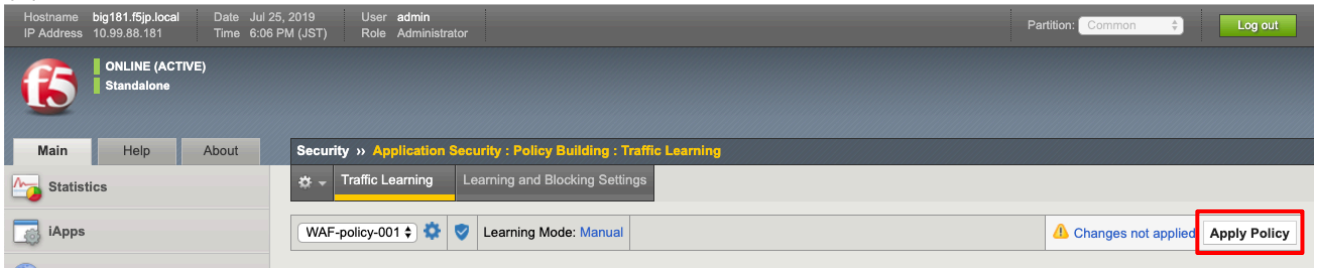
(4) 「Security」→「Application Security」→「Policy Building」→「Traffic Learning」で表示された「Enforce Cookie」をクリックすると、以下の画面が現れます。
「SESSION」が、この Web アプリケーションのセッション ID(Cookie)です。



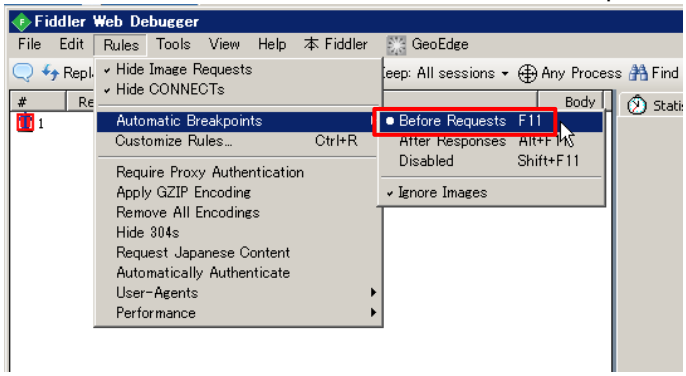
(5) 「Enforce Cookie」にチェックを入れ、「Accept Suggestion」ボタンを押します。



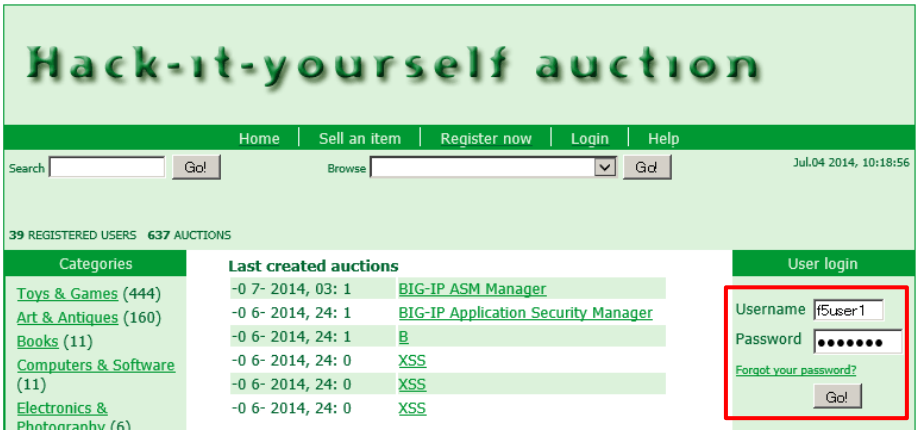
(6) 「Apply Policy」ボタンを押します。



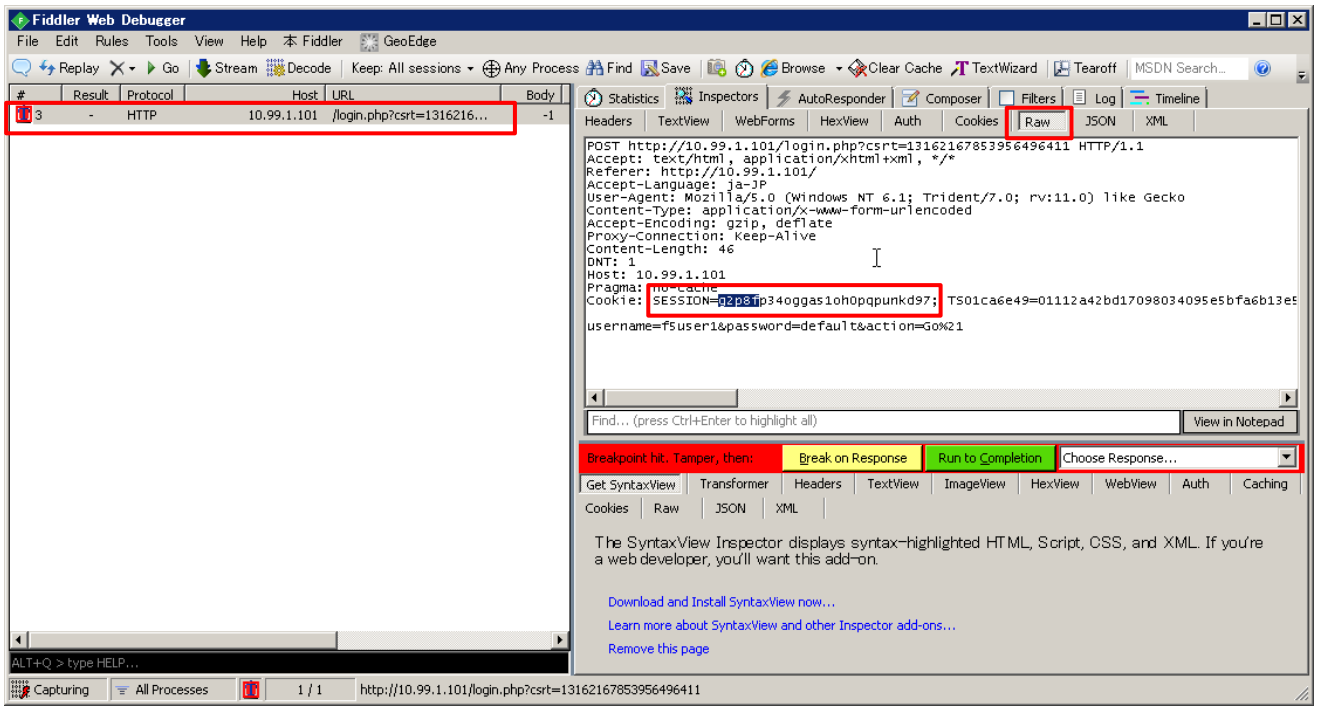
(7) クライアントのブラウザから攻撃を模擬するために、Fiddler2 を利用します。
Fiddler2 を起動し、「Rules」→「Automatic Breakpoints」→「Before Requests」を選択（有効に）します。



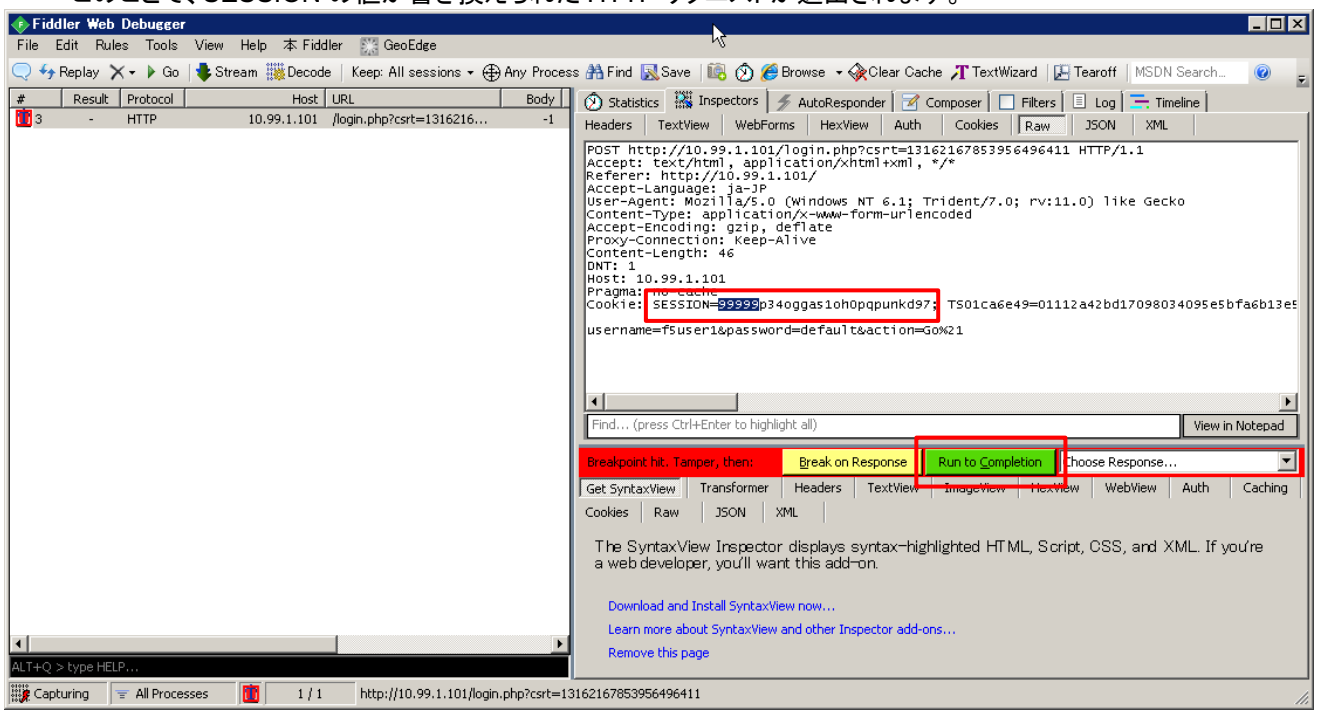
(8) ブラウザで http の Virtual Server にアクセスし、正当なユーザ(f5userX)とパスワードを入力し、「Go!」ボタンを押します。



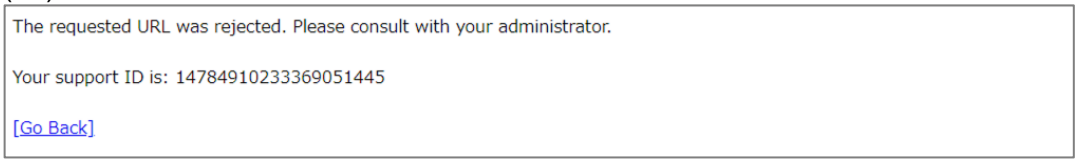
(9) Fiddler2 がこの HTTP リクエストをインターセプトするので、先に進みません。
Fiddler2 の左画面でその HTTP リクエストを選択し、右画面で「Inspectors」→「Raw」タブを選択します。
「Cookie: SESSION=」を見つけます。



(10) ここでは、SESSION 値の先頭五文字を、「99999」に書き換えてみました。
「Run to Completion」ボタンを押します。
このことで、SESSION の値が書き換えられた HTTP リクエストが送出されます。

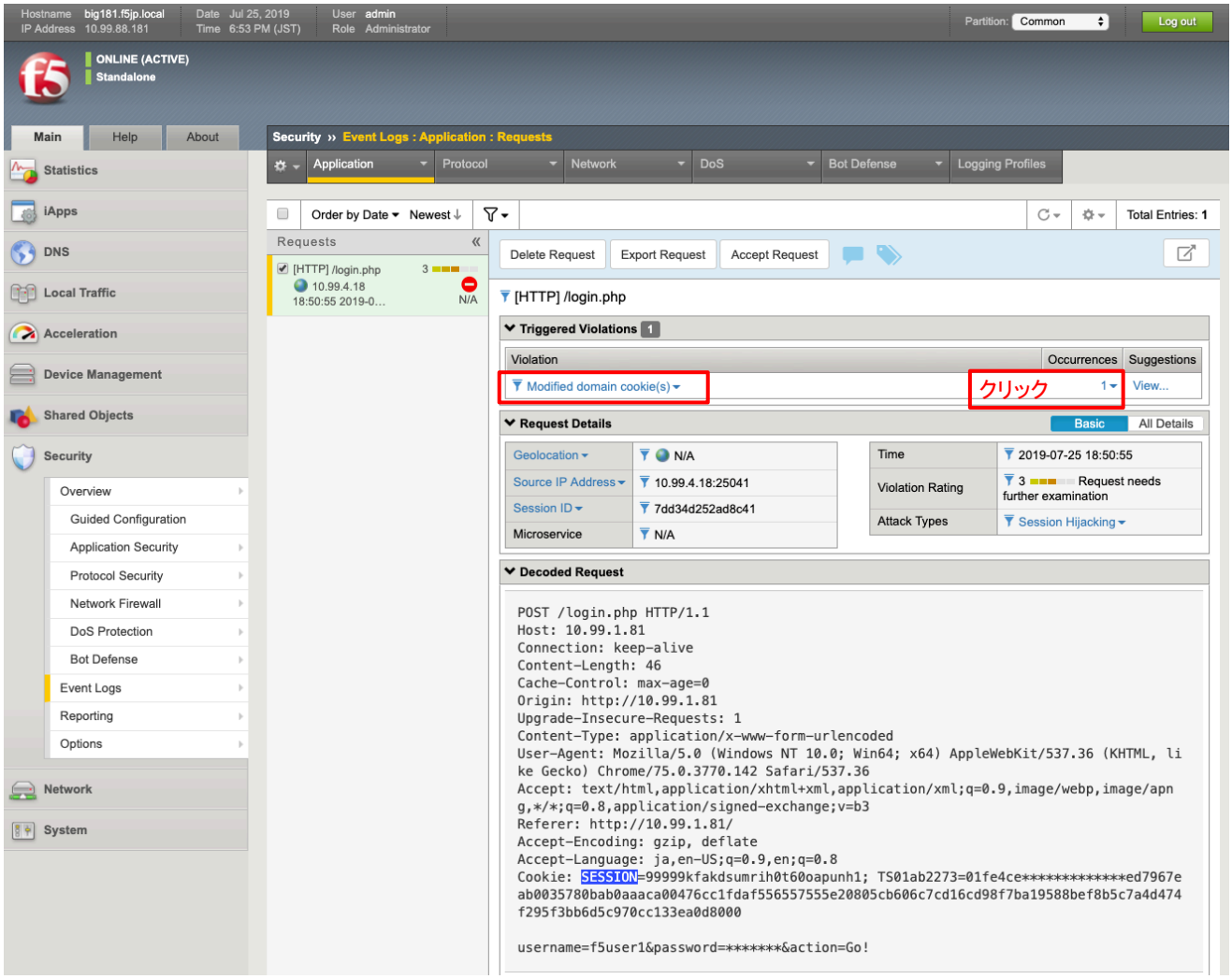


(11) 攻撃として検知されます。

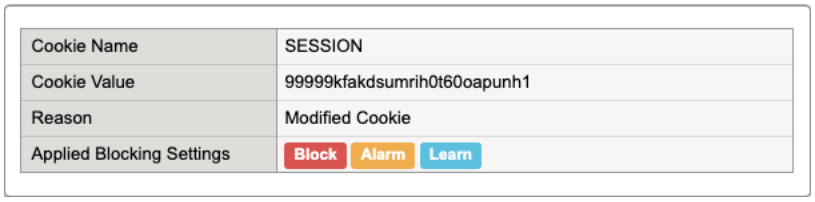


Fiddler2 を終了するか、「Rules」→「Automatic Breakpoints」→「Disabled」を選択（無効に）します。

(12)「Security」→「Event Logs」→「Application」→「Requests」でログを確認します。Modified domain cookie が検知されていることが分かります。



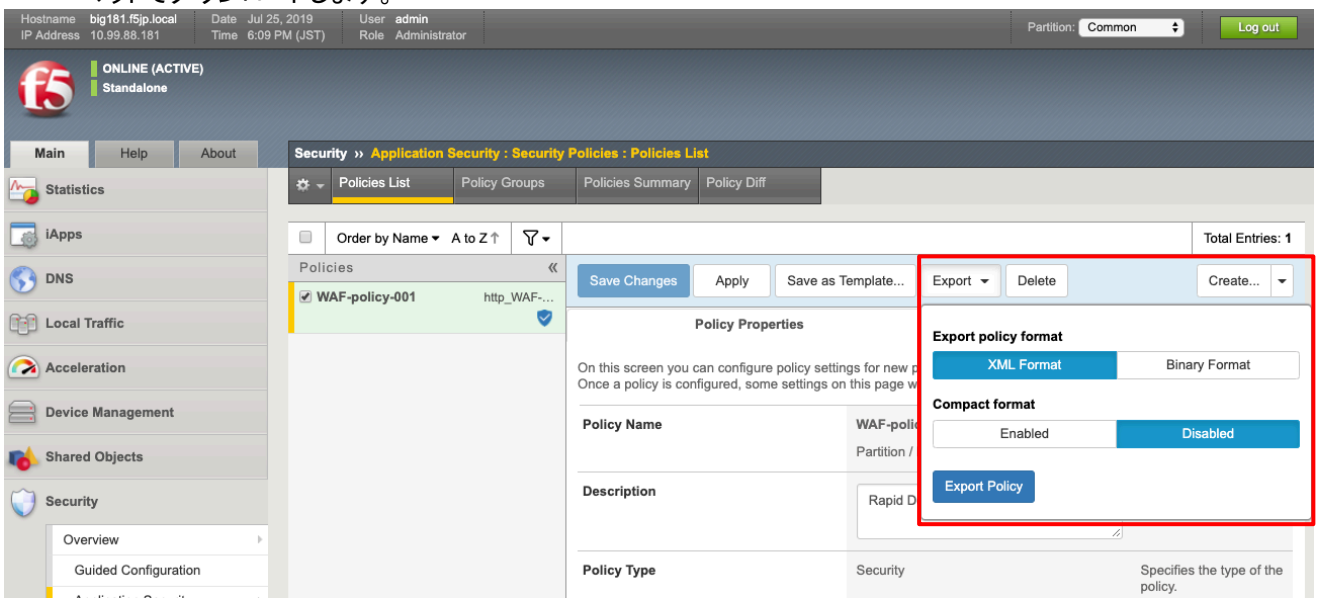
(13) Occurrences をクリックすると、以下のように SESSION の値がどのような値に書き換えられたか表示されます。



10. Adv.WAF ポリシーの保存

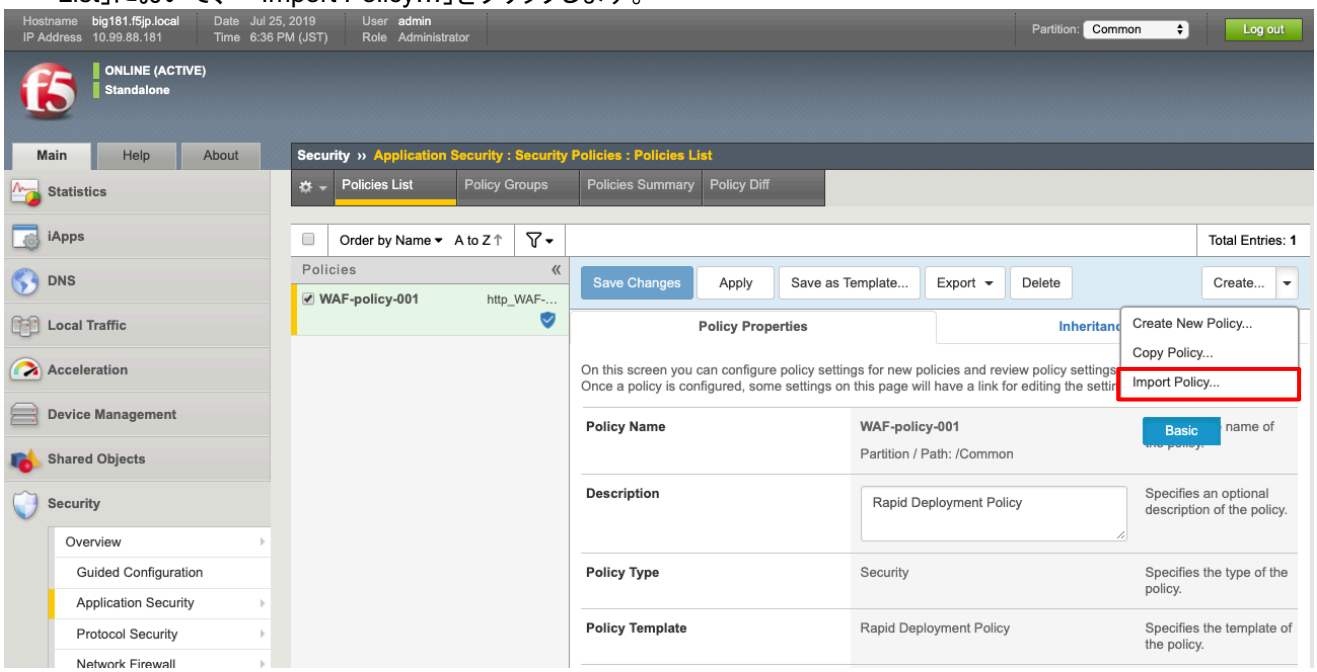
作成したポリシーを保存することが可能です。また、保存したポリシーを import することも可能です。

(1) はじめに、作成したポリシーを保存します。「Security」→「Application Security」→「Security Policies」→「Policies List」をクリックすると以下の画面が表示されます。「Export」→「Export Policy」をクリックし、XML フォーマットでダウンロードします。

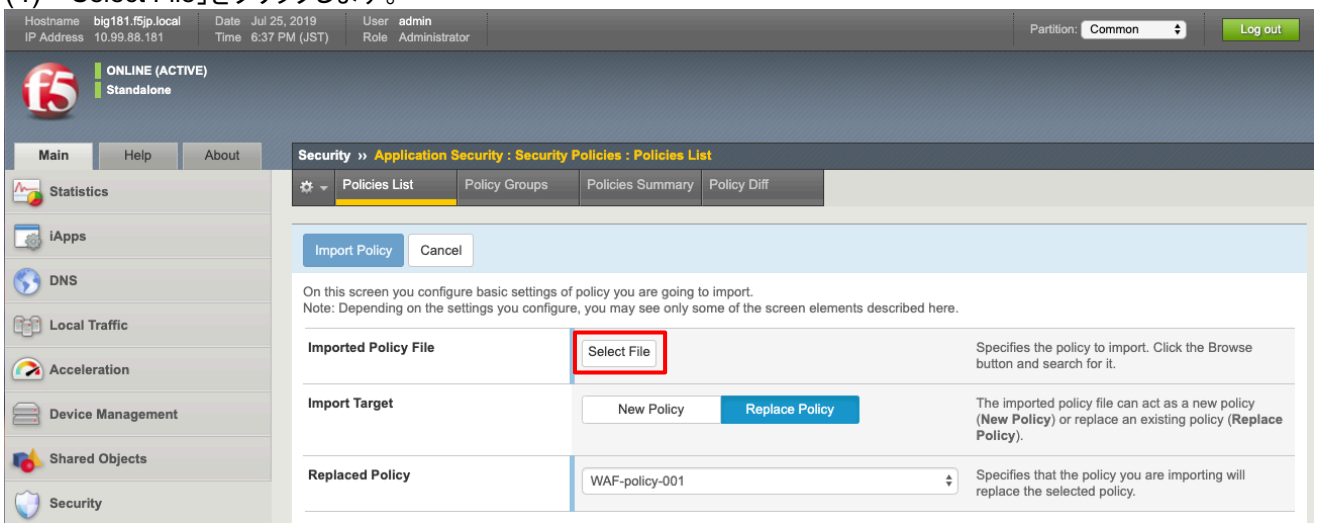


(2) Downloads フォルダに以下のようにファイルが保存されています。

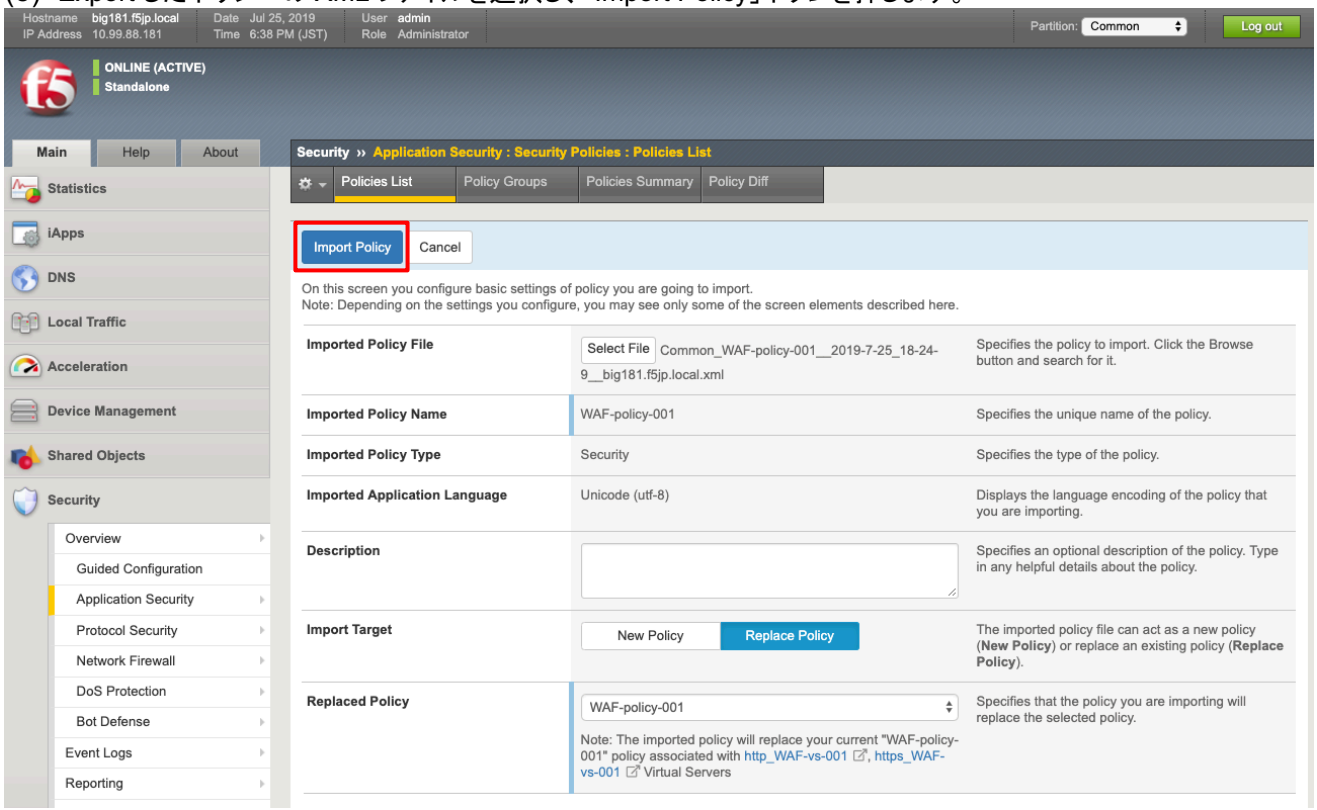
(3) 次に、保存したポリシーを import します。「Security」→「Application Security」→「Security Policies」→「Policies List」において、「Import Policy…」をクリックします。



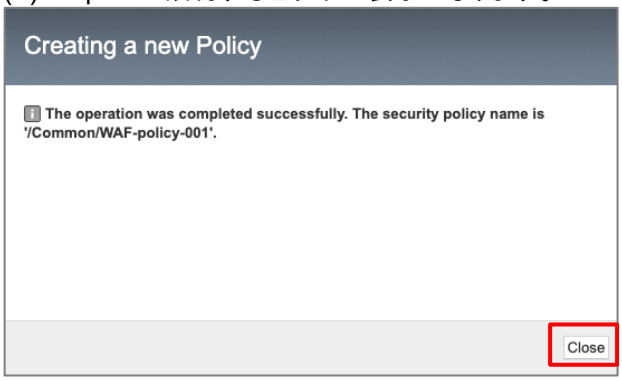
(4) 「Select File」をクリックします。



(5) Export したポリシーの XML ファイルを選択し、「Import Policy」ボタンを押します。



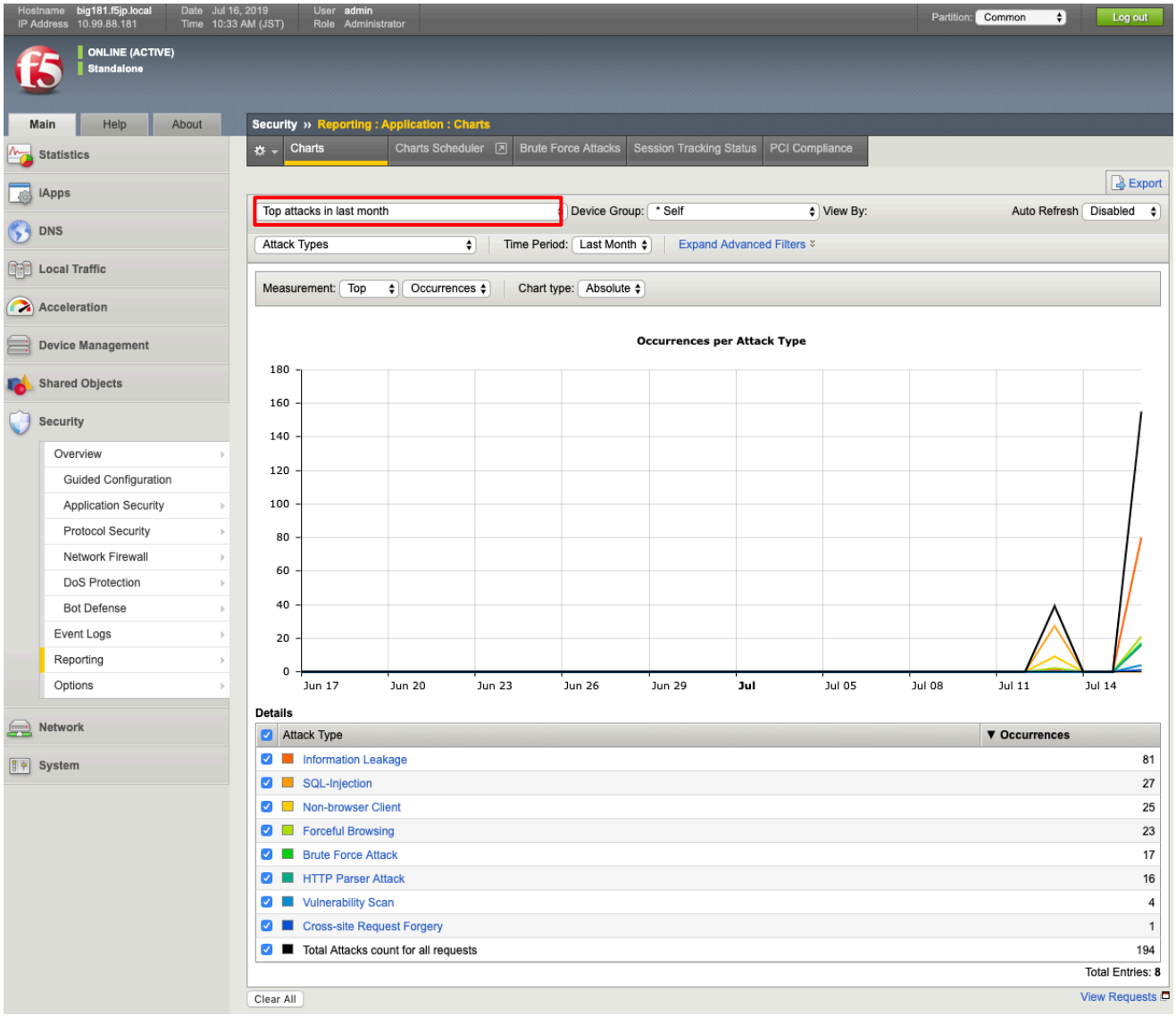
(6) Import が成功すると以下の表示になります。Close ボタンを押します。



11. レポーティング

Reporting 機能にて、グラフィカルなレポートを表示することが可能です。

(1) 「Security」→「Reporting」→「Application」→「Charts」で表示された画面にて確認します。例えば、「Top attackers in last month」を選択すると、1ヶ月分の攻撃が表示されます。



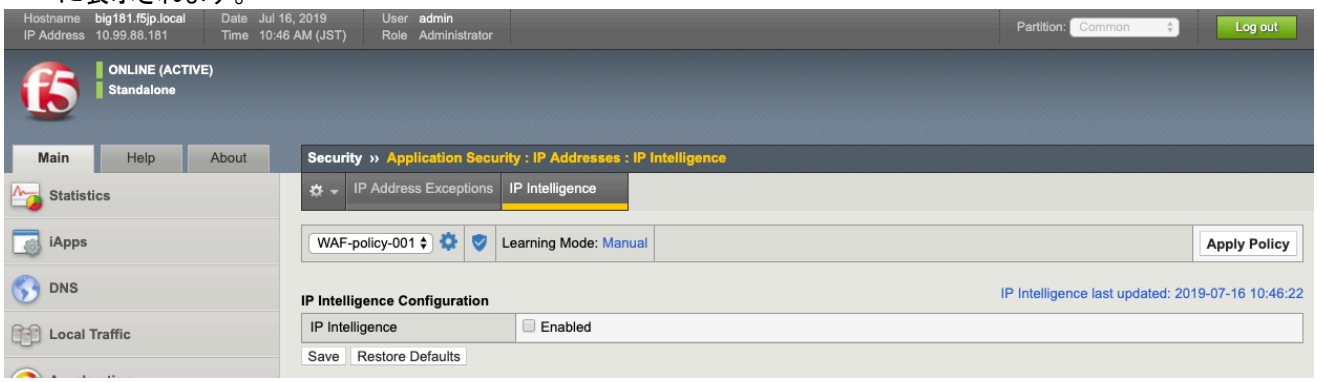
他にも、「Top violations in last month」や「Top blocked URLs」など様々なパターンのレポートを表示可能となっています。

12. IP Intelligence の設定

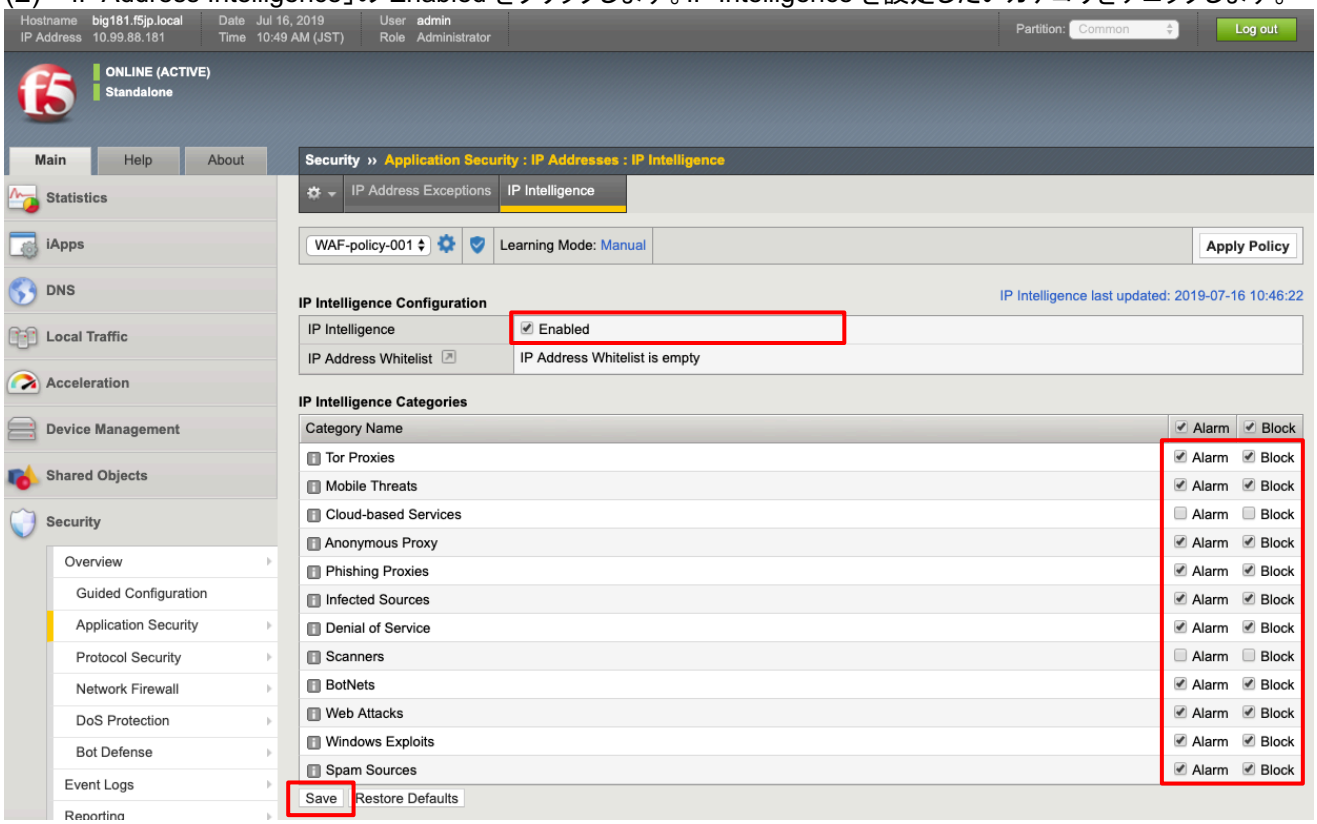
※ F5 ハンズオンでは実施しません。

サブスクリプションライセンスの IP Intelligence を設定することで、悪意ある IP アドレスからの攻撃を Block することが可能です。Appliction Security 処理の前段で IP アドレスの評価が行われるため、CPU 負荷高騰を和らげる効果があります。WAF と L7DDoS において IP Intelligence を利用することが可能です。
インターネットから悪意ある IP アドレスデータベースを取得するため、事前に DNS 設定が必要になります。

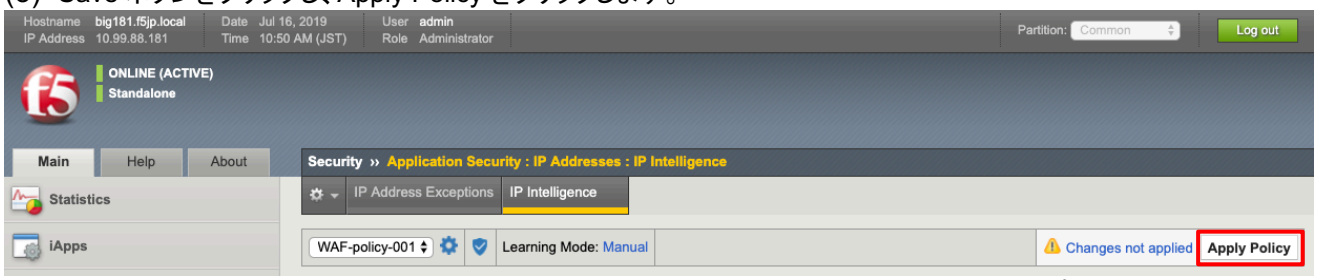
(1) 「Security」→「Application Security」→「IP Addresses」→「IP Address Intelligence」をクリックすると以下のように表示されます。



(2) 「IP Address Intelligence」の Enabled をクリックします。IP Intelligence を設定したいカテゴリをチェックします。



(3) Save ボタンをクリックし、Apply Policy をクリックします。

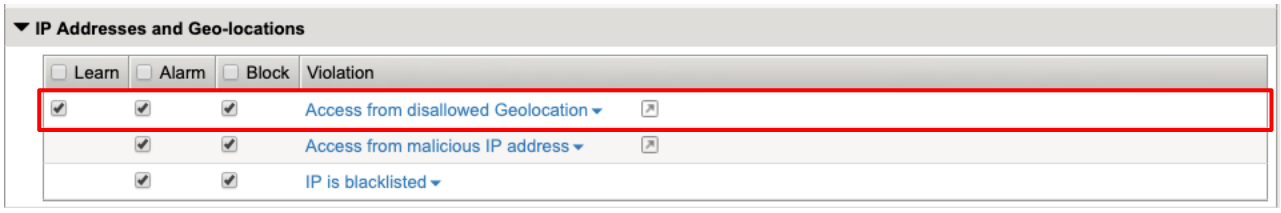


上記の他、L7DoS Shun と IP Intelligence を組合せることによって、IP Intelligence の IP レピュテーション DB のリストを L7DoS 対策の shun list（Auto-blacklisting）として利用が可能です。

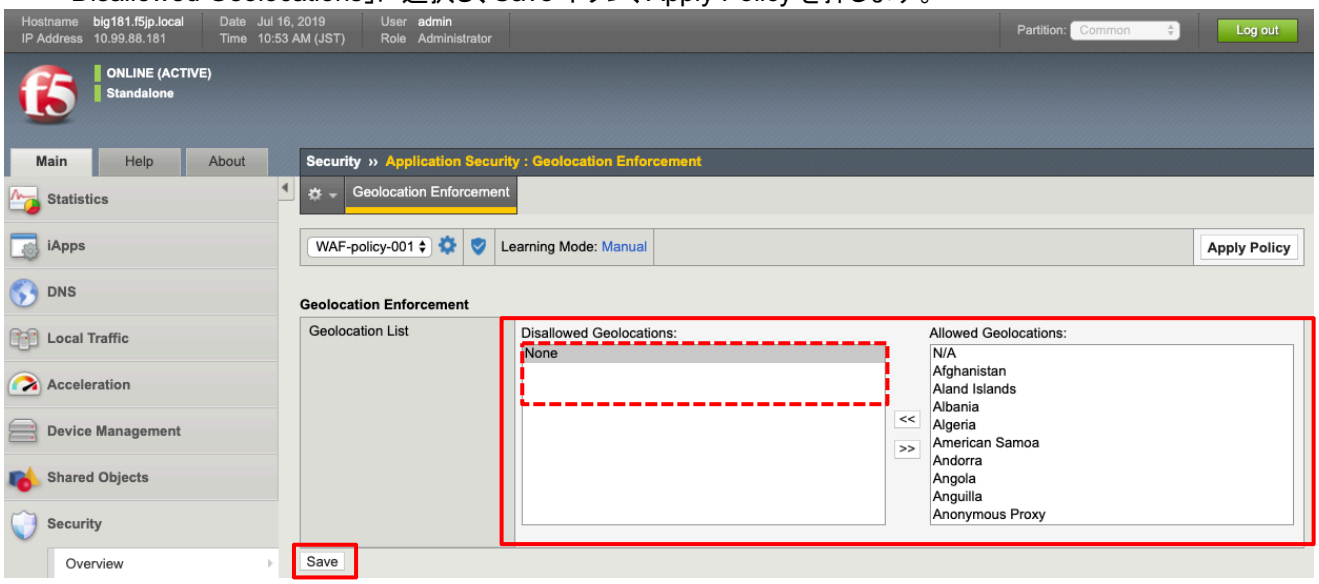
13. Geolocation の設定

Geolocation Enforcement の設定を行うことで、接続される予定のない国からの接続をブロックすることができます。

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」の「IP Addresses/Geolocations」において、「Access from disallowed Geolocation」の Learn/Alarm/Block がチェックされていることを確認します。



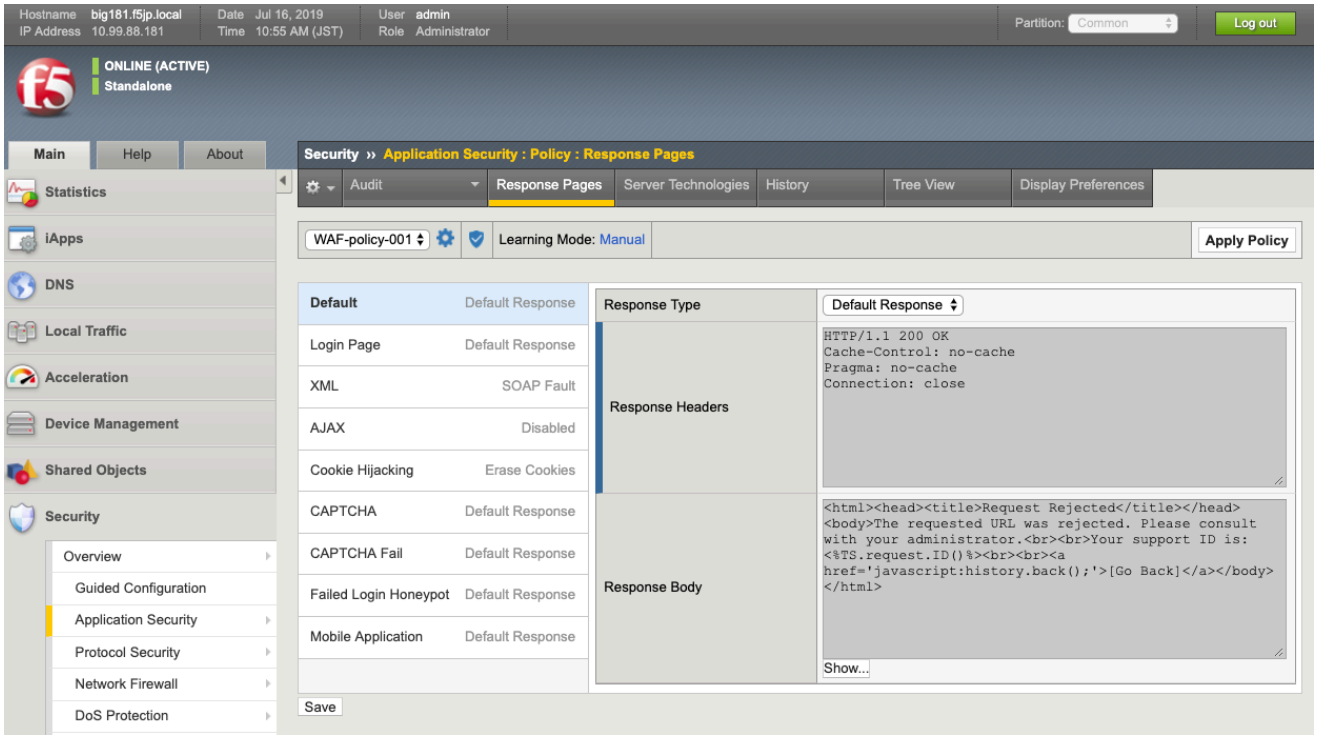
(2) 「Security」→「Application Security」→「Geolocation Enforcement」 にて、接続する予定のない国を「Disallowed Geolocations」に選択し、Save ボタン、Apply Policy を押します。



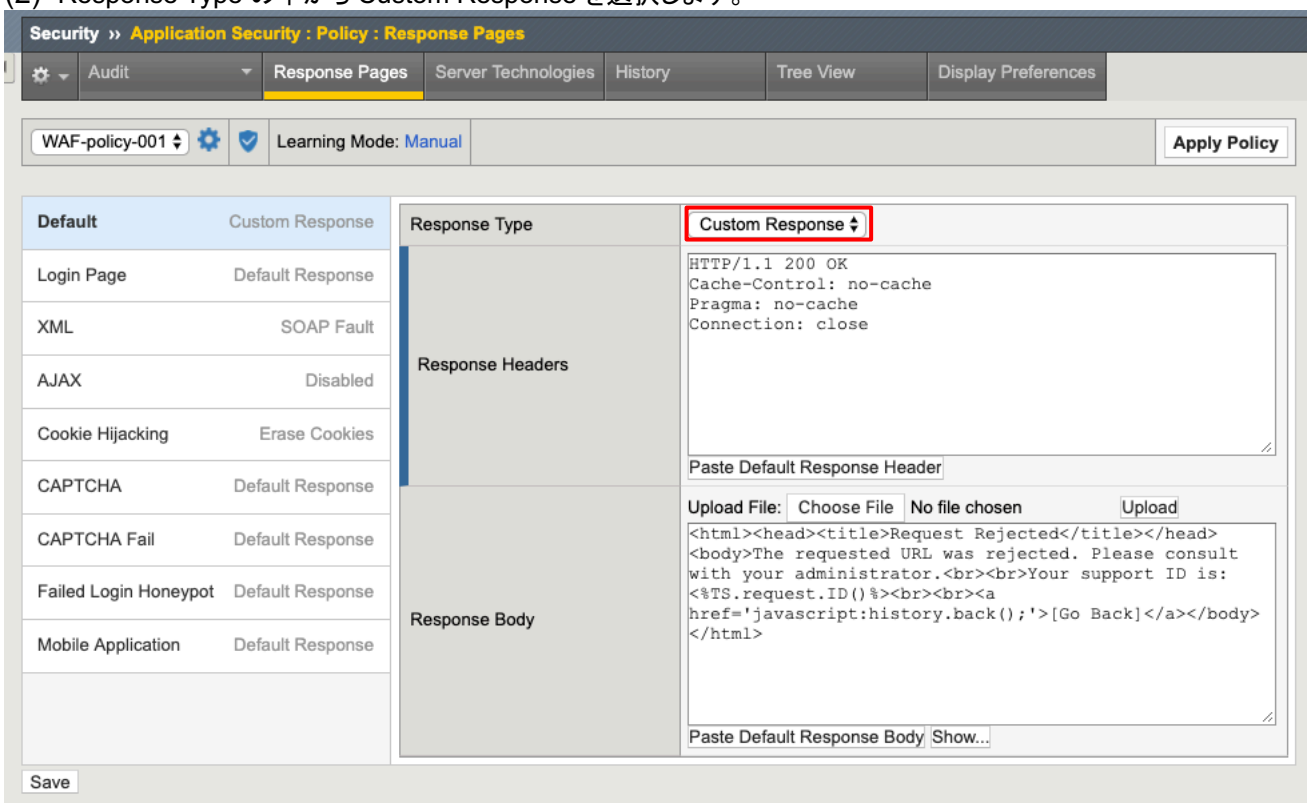
14. ブロック時のレスポンスページの変更

攻撃をブロックした際にユーザに返されるレスポンスページの内容を変更することが可能です。

(1) 「Security」→「Application Security」→「Policy」→「Response Pages」をクリックすると以下の画面が表示されます。デフォルトでは編集ができません。



(2) Response Type の中から Custom Response を選択します。



上記から Header や Body の内容を変更することが可能です。

15. シグネチャの運用

シグネチャの追加、変更、更新などについて記載します。

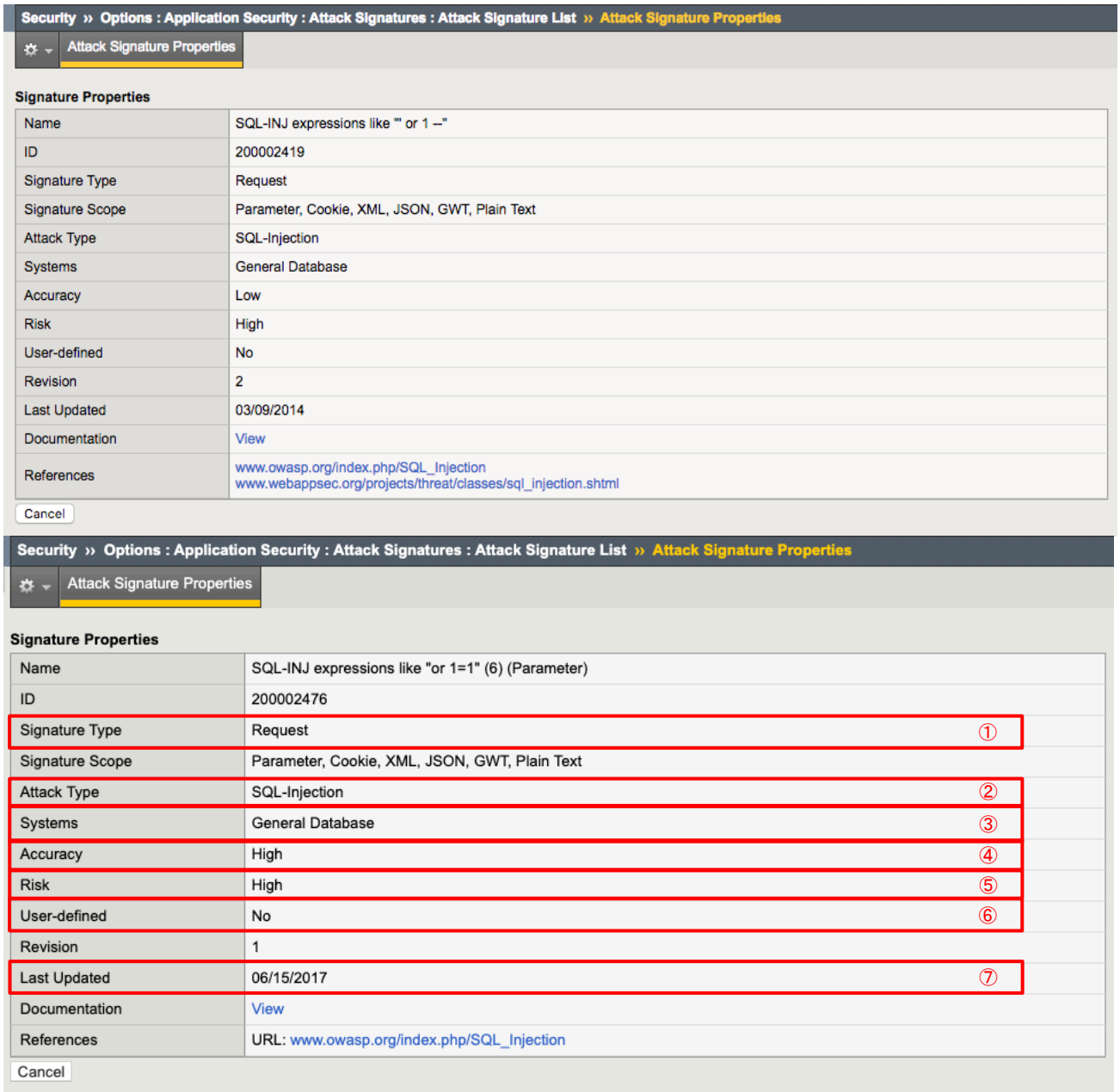
15.1. シグネチャセットとは

Adv.WAF でのシグネチャの割当ては、「シグネチャセット」という単位での割付を行います。
そのシグネチャセットとはどういうものかを説明します。

15.1.1. シグネチャの属性

シグネチャセットを理解するには、まずシグネチャの属性について知る必要があります。

(1) 「Security」→「Options」→「Application Security」→「Attack Signatures」→「Attack Signature List」で表示された一つのシグネチャをクリックして現れた画面で、Signature Name をクリックします。
各シグネチャは全て、以下のような属性を持っています。

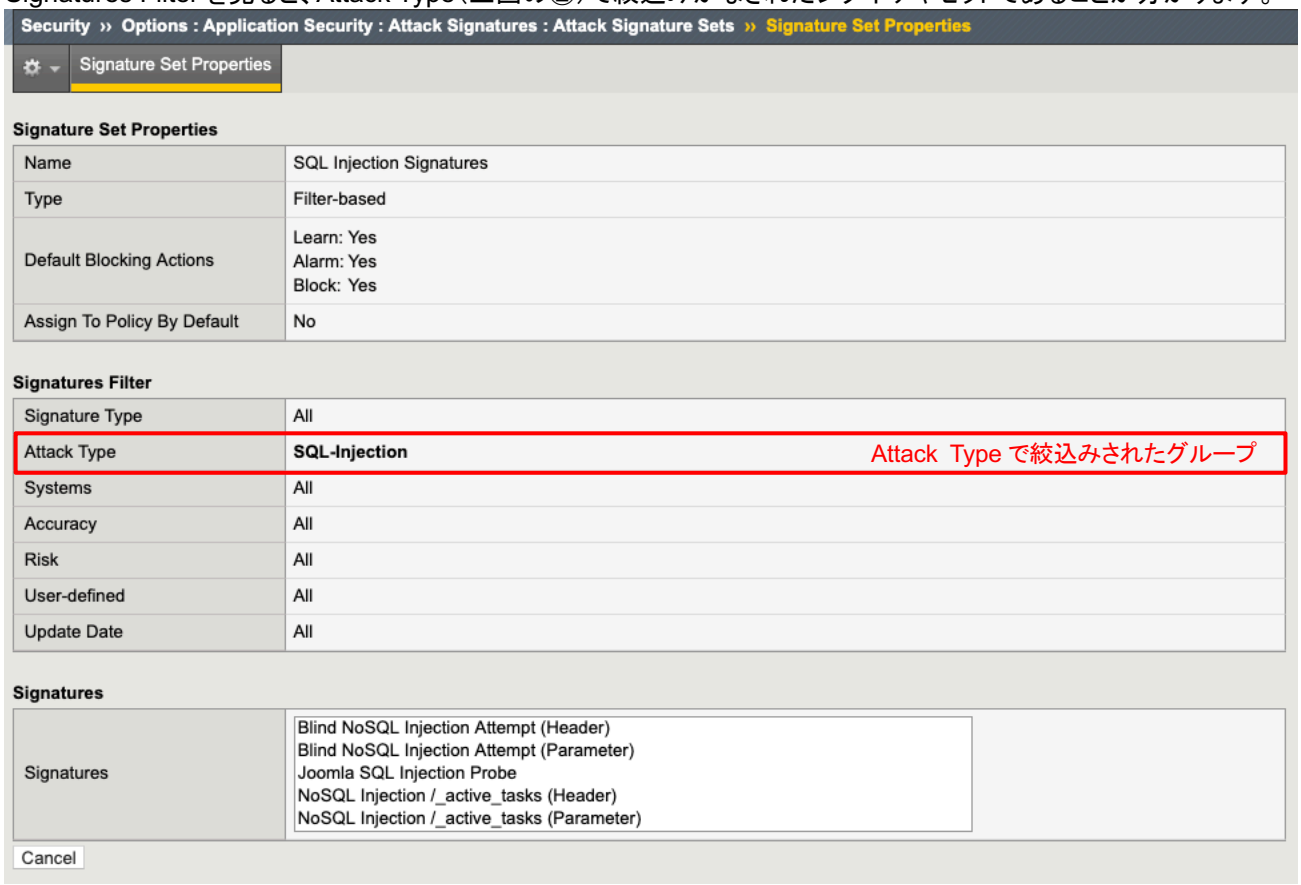


この属性のうち、上記①～⑦の値のいずれか（またはその組合せ）で絞込みした、複数のシグネチャのグループをシグネチャセットと呼びます。

(2) シグネチャセットの例：Generic Detection Signatures

ここでは、デフォルトで割当てられるシグネチャセット：Generic Detection Signatures を見てみます。「Security」→「Options」→「Application Security」→「Attack Signatures」→「Attack Signature Sets」で表示された複数のシグネチャセットの中から、「Generic Detection Signatures」を選びます。

Signatures Filter を見ると、Attack Type（上図の②）で絞込みがなされたシグネチャセットであることが分かります。



このシグネチャセットは、以下3つのような、一般的な攻撃に対するシグネチャに絞り込まれています。

① General Database
- 一般的なデータベースへの攻撃シグネチャ

② System Independent
- 特定のシステムに限定されない攻撃シグネチャ

③ Various Systems
- 特定のシステムまたはコードバージョンに依存していない、一般的な攻撃シグネチャ

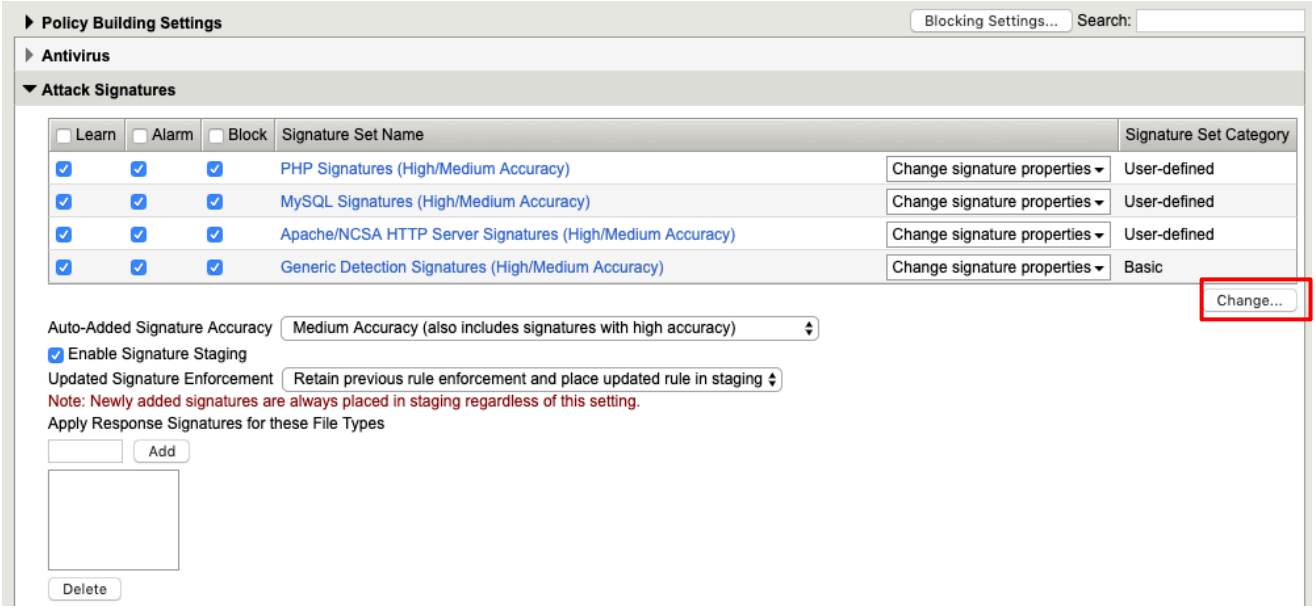
よって、守るべき Web アプリケーションが使っているシステム（例：Apache や PHP など）が明確ならば、それらのシグネチャも追加で割当てることを推奨します。

詳細は「K10726: Overview of the attack signature Systems and Attack Type fields」を参照ください。
http://support.f5.com/csp/article/K10726

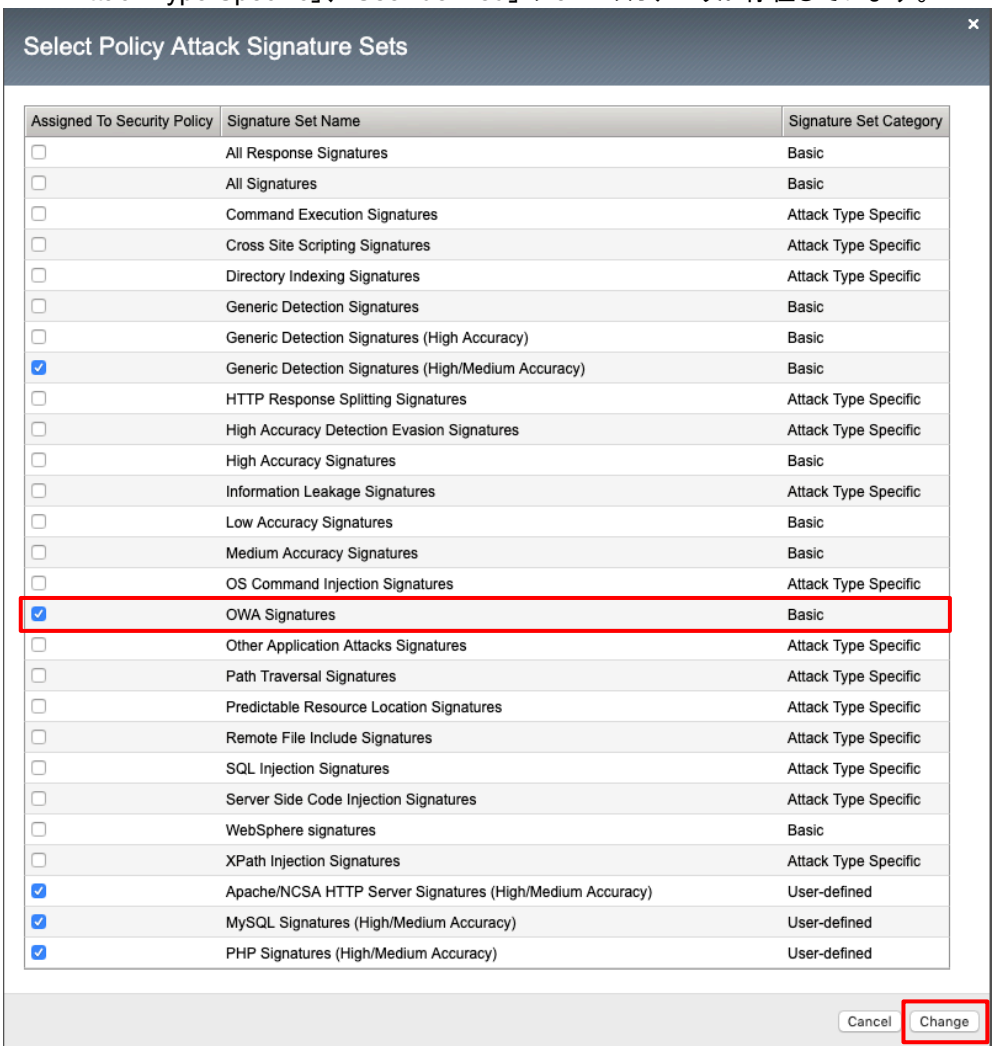
15.1.2. シグネチャセットの割り当て方法

ポリシーへのシグネチャ割当ては、このシグネチャセット単位でしかできません。

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」の「Attack Signatures」において「Change」ボタンを押します。



(2) 一覧から必要なセットを選んで、Change を押します。この「Available Signature Sets」の中には、「Basic」、「Attack Type Specific」、「User-defined」の 3 つのカテゴリが存在しています。



15.1.3. 「Basic」シグネチャセット

「Basic」カテゴリには、以下のようなシグネチャセット（一例）が存在します。
それぞれ、各シグネチャが持ついずれかの属性でグループ化されています。

Signature Set	Generic Detection Signatures	All Signatures	All Response Signatures	OWA Signatures	WebSphere signatures
Signature Type	All	All	Response	All	All
Attack Type	All	All	All	All	All
Systems	General Database System Independent Various systems	All	All	ASP IIS Microsoft Windows Outlook Web Access WebDAV	Apache Apache Tomcat CGI General Database IBM DB2 IIS Java Servlets/JSP Lotus Domino Macromedia ColdFusion Microsoft SQL Server Microsoft Windows MySQL Oracle Other Web Server PHP PostgreSQL Proxy Servers SSI (Server Side Includes) Sybase/ASE System Independent Unix/Linux Various systems XML
Accuracy	All	All	All	All	All
Risk	All	All	All	All	All
User-defined	All	All	All	All	All
Update Date	All	All	All	All	All

15.1.4. 「Attack Type Specific」 シグネチャセット

攻撃手法が限定されたシグネチャセットです。これらのセットは「Attack Type」で絞込みされています。

Signature Set	Command Execution Signatures	Cross Site Scripting Signatures	Directory Indexing Signatures	・・・	XXXX Signatures
Signature Type	All	All	All	・・・	All
Attack Type	Command Execution	Cross Site Scripting (XSS)	Directory Indexing	・・・	XXXX
Systems	All	All	All	・・・	All
Accuracy	All	All	All	・・・	All
Risk	All	All	All	・・・	All
User-defined	All	All	All	・・・	All
Update Date	All	All	All	・・・	All

15.1.5. 「User-defined」シグネチャセット

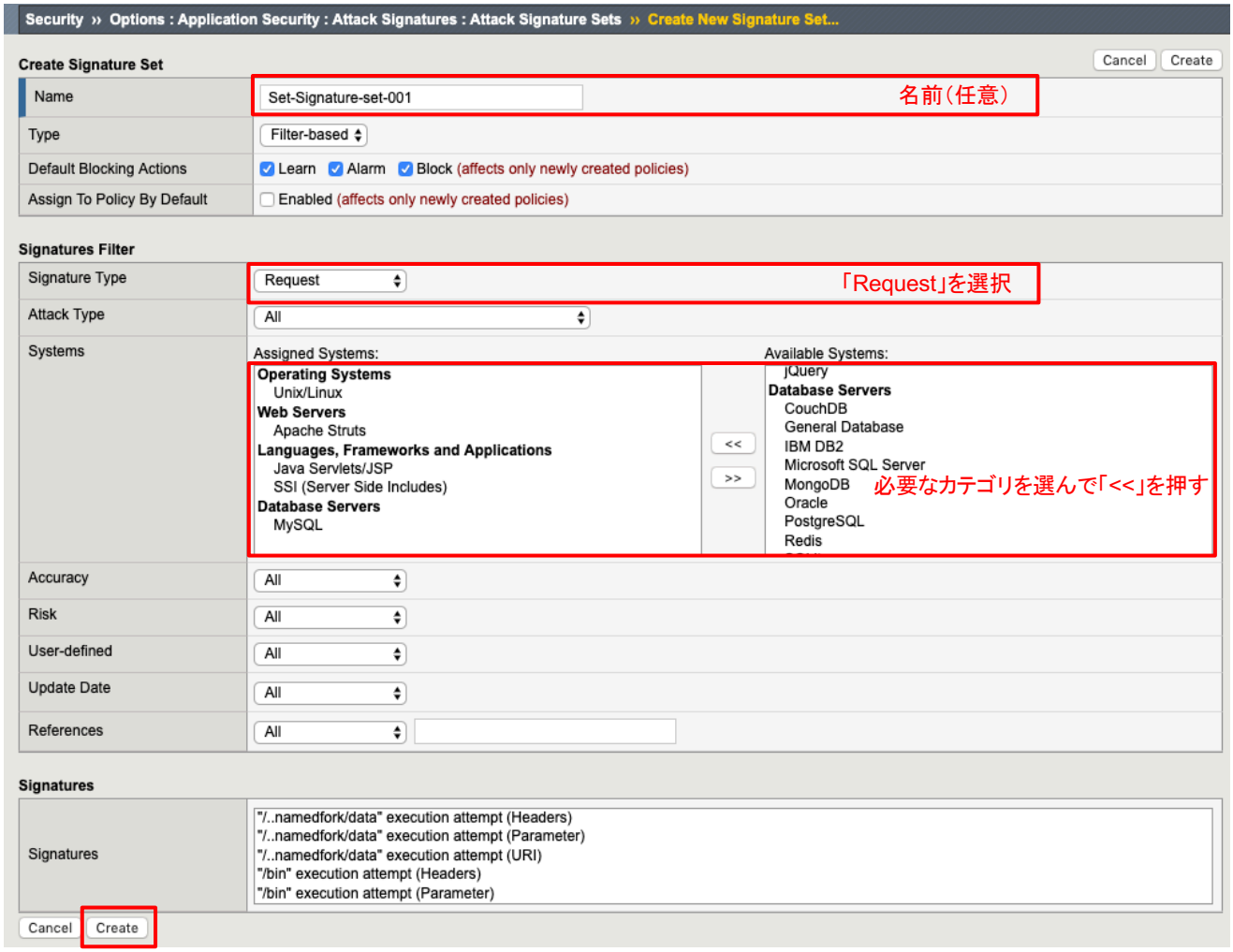
ユーザが設定（組合せ）変更可能なシグネチャセットです。
いくつかのテンプレートが事前に用意されています。

15.2. シグネチャセットの作成

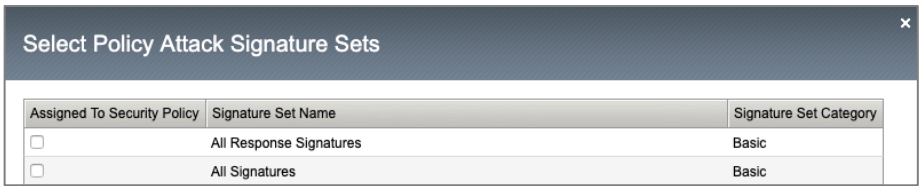
シグネチャセットは自分で作ることもできます。

例えば、「All Signatures」は Request だけでなく Response に対するシグネチャも入っているので、Request だけのシグネチャを作りたい、と仮定します。その例を以下に示します。

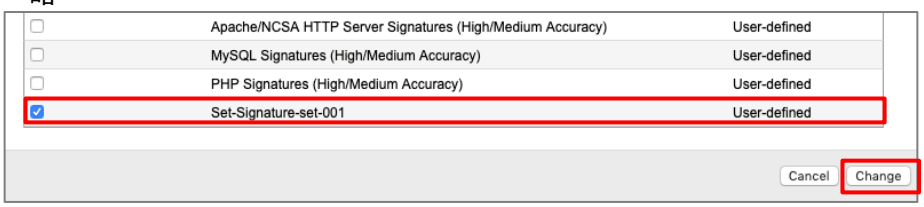
(1) 「Security」→「Options」→「Application Security」→「Attack Signatures」→「Attack Signature Sets」で表示された画面の右上の「Create」ボタンを押すと、以下の画面が現れます。以下のように設定します。



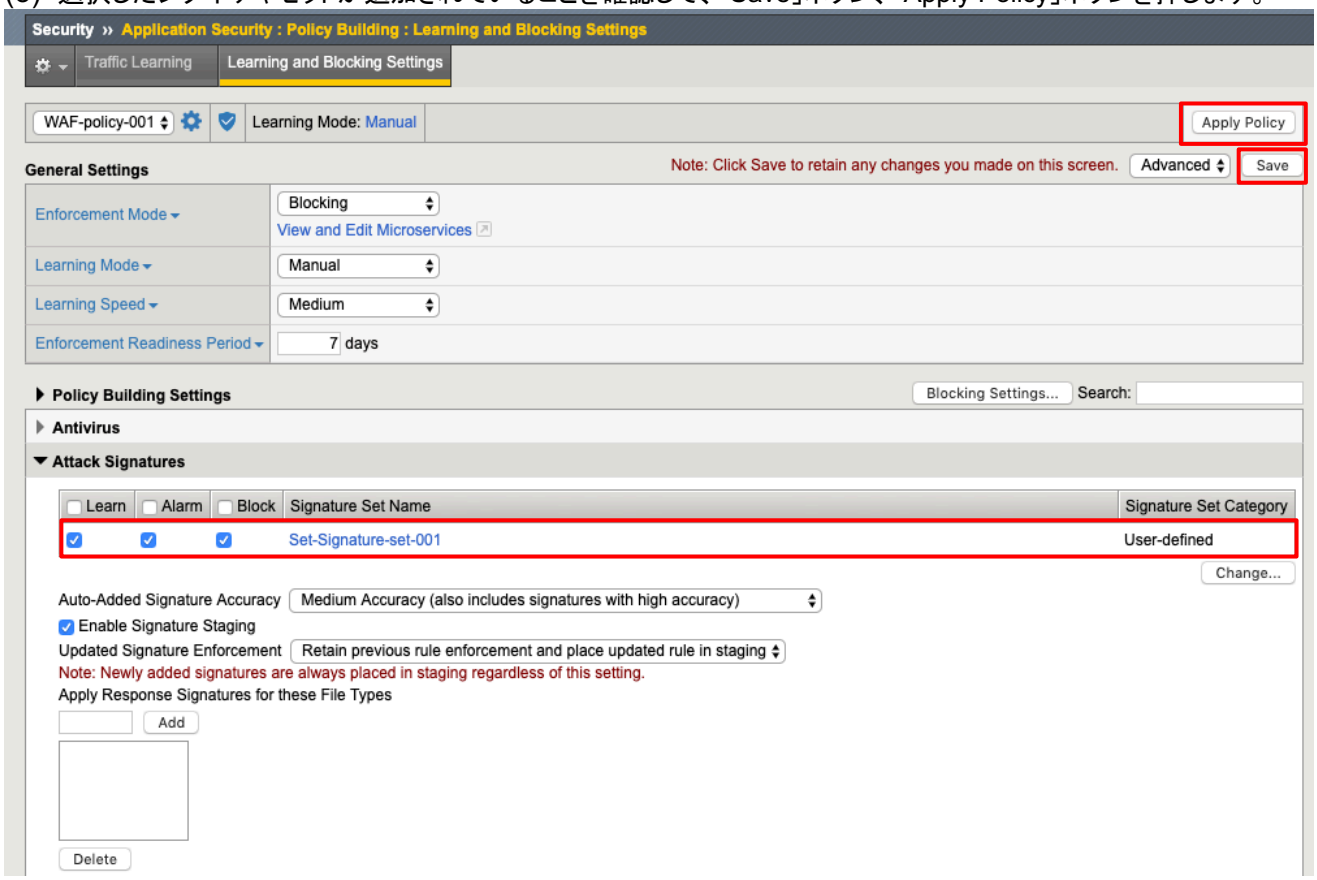
(2) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」の「Attack Signatures」において、「Change」ボタンを押し、前項で作成したシグネチャセットを探します。以下のように該当のシグネチャセットを選択して、「Change」ボタンを押します。



～略



(3) 選択したシグネチャセットが追加されていることを確認して、「Save」ボタン、「Apply Policy」ボタンを押します。



これで、作成したシグネチャセットの割当ては完了です。

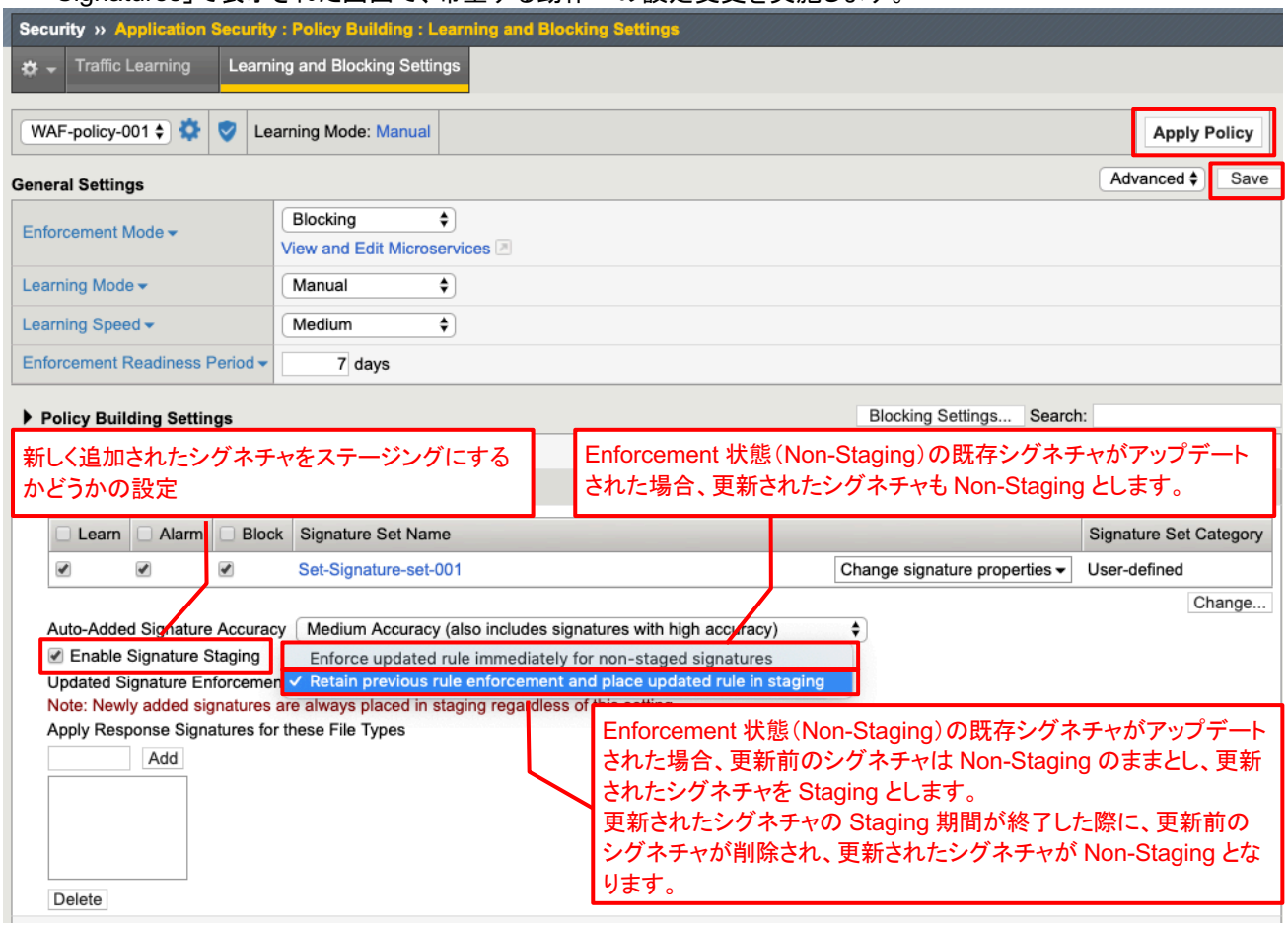
15.3. 新しい Signature の更新

F5 から新しい Signature が出た場合、新しい Signature を Update することで新たな攻撃に対応することができます。Signature は定期的に更新されております。以下の記事を参考にして下さい。
https://support.f5.com/csp/article/K8217

15.3.1. シグネチャが更新された直後の振る舞い

シグネチャが更新された場合に、ステージングモードで運用するか、即座に Learn/Alarm/Block の設定が適用されるかの指定が可能です。
また、既存シグネチャの更新後の振る舞いについての指定も可能です。

(1) 「Security」→「Application Security」→「Policy Building」→「Learning and Blocking Settings」の「Attack Signatures」で表示された画面で、希望する動作への設定変更を実施します。

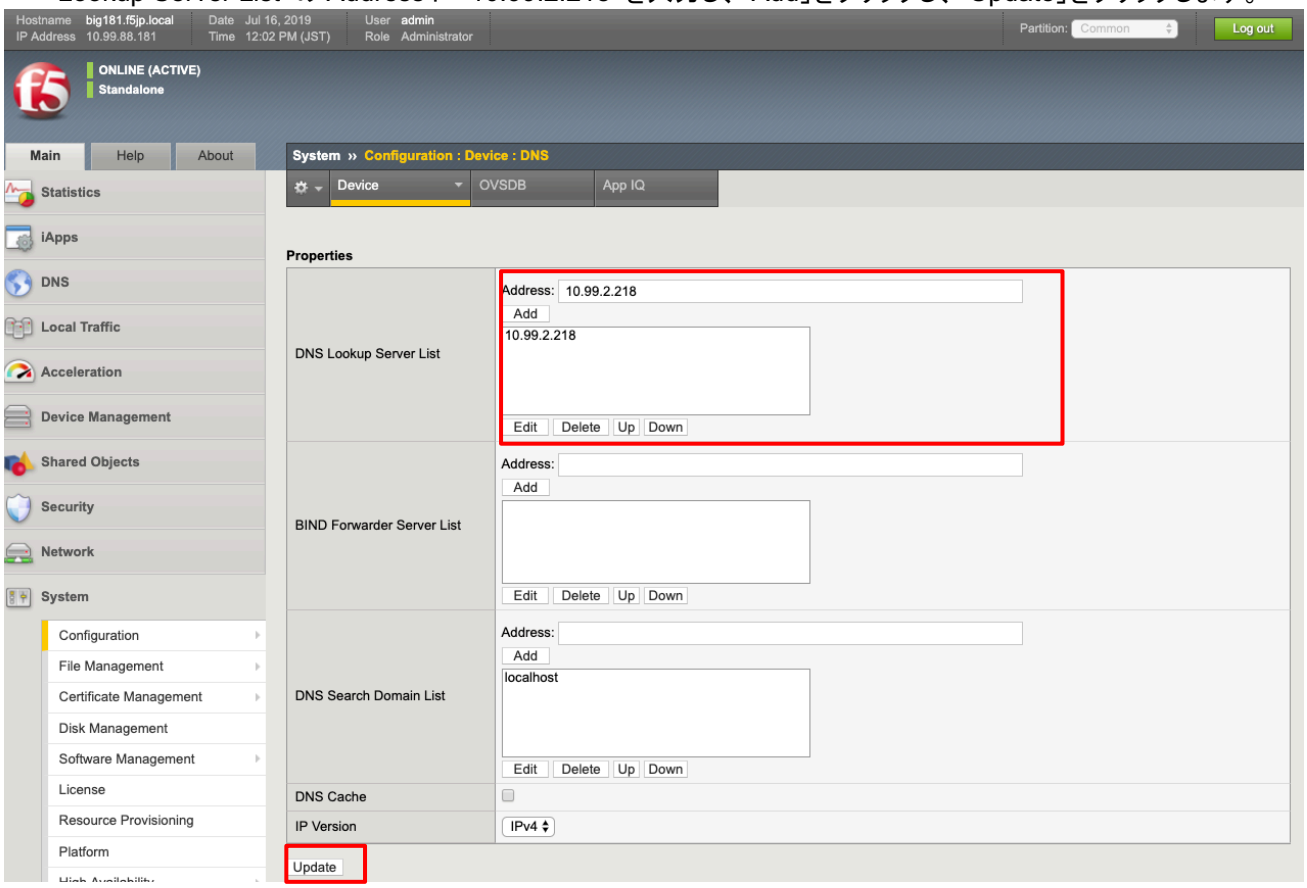


(2) 変更後、「Save」、「Apply Policy」を押します。

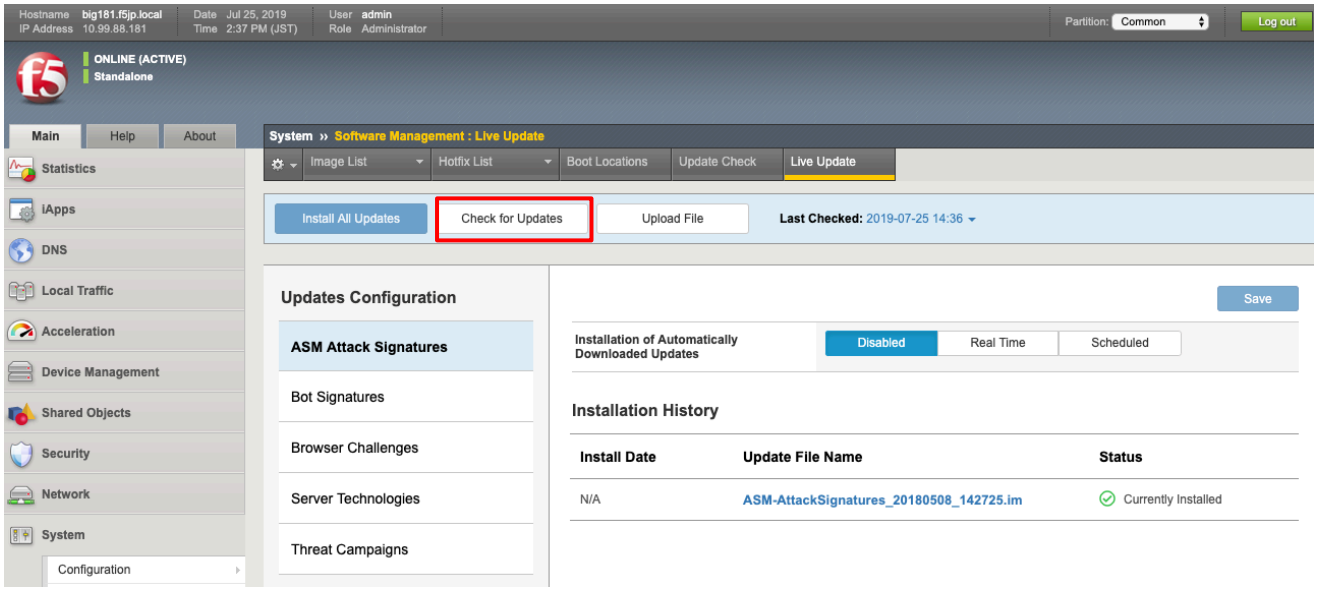
15.3.2. シグネチャ更新の実行

※F5 ハンズオンでは実施しません

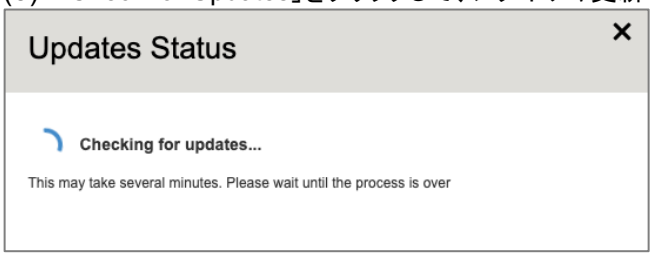
(1) まず、DNS の設定を行います。「System」→「Configration」→「Device」→「DNS」で表示された画面で、“DNS Lookup Server List”の Address に“10.99.2.218“を入力し、「Add」をクリックし、「Update」をクリックします。



(2) 「System」→「Software Management」→「Live Update」で表示された画面で、現状の設定を確認します。以下のように表示されます。

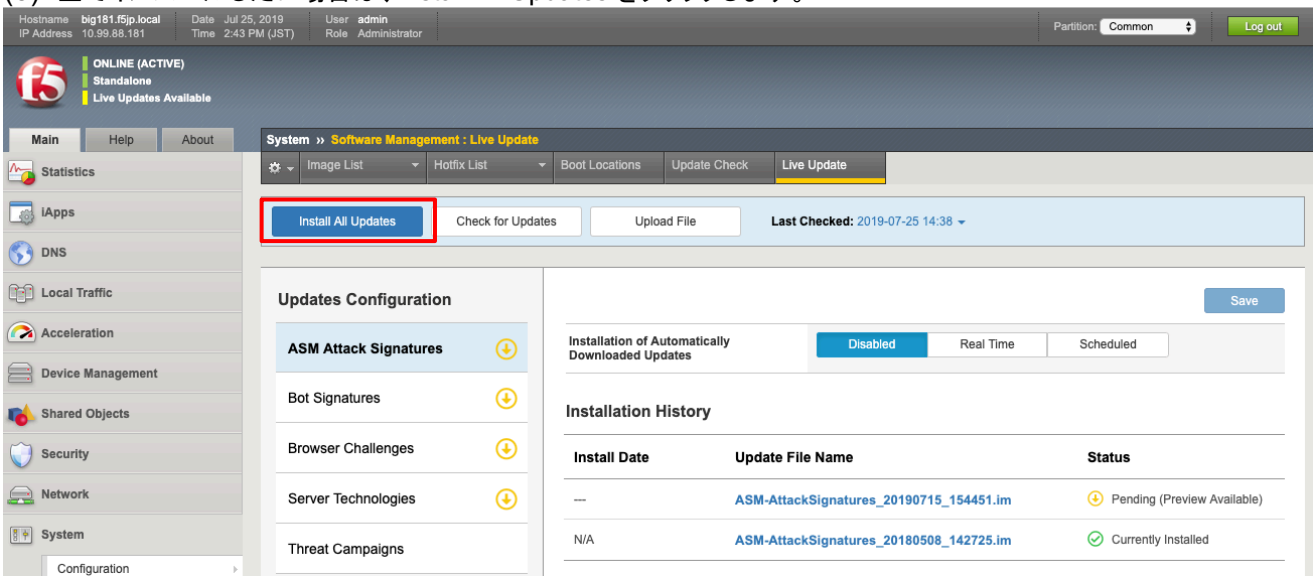


(3) 「Check for Updates」をクリックして、シグネチャ更新の有無を確認します。（チェックには数分かかります。）

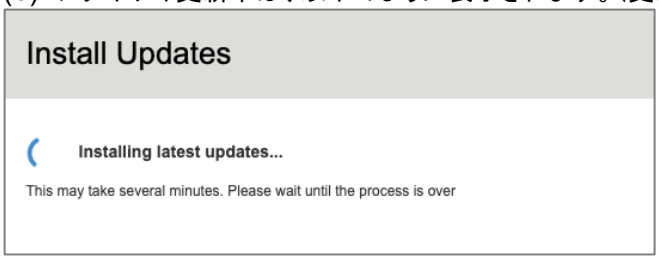


(4) 更新可能なシグネチャがある場合、以下のように表示されます。×を推し、画面を閉じます。

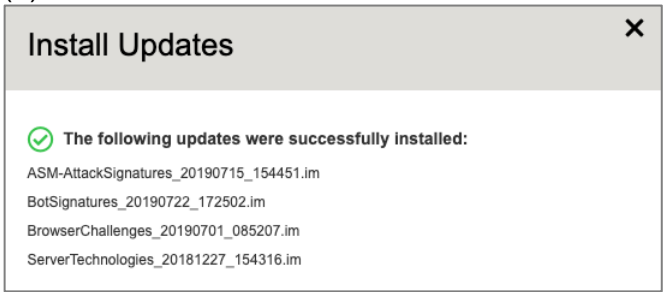
(5) 全てインスールしたい場合は、Install All Updates をクリックします。



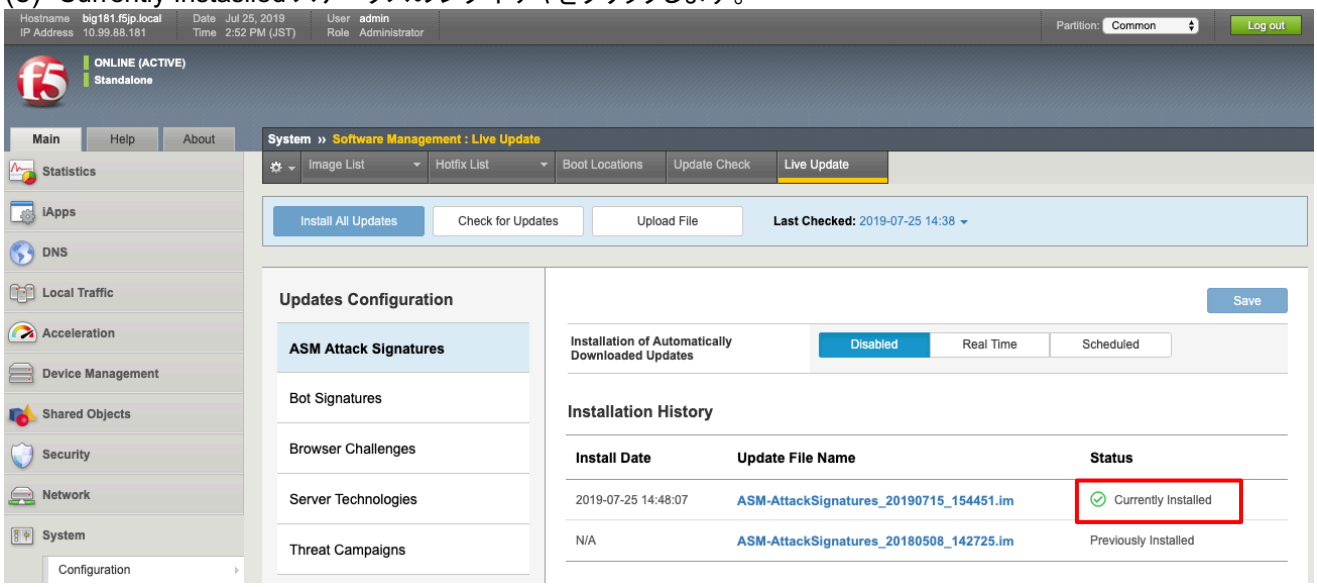
(6) シグネチャ更新中は、以下のように表示されます。（更新には数分かかります。）



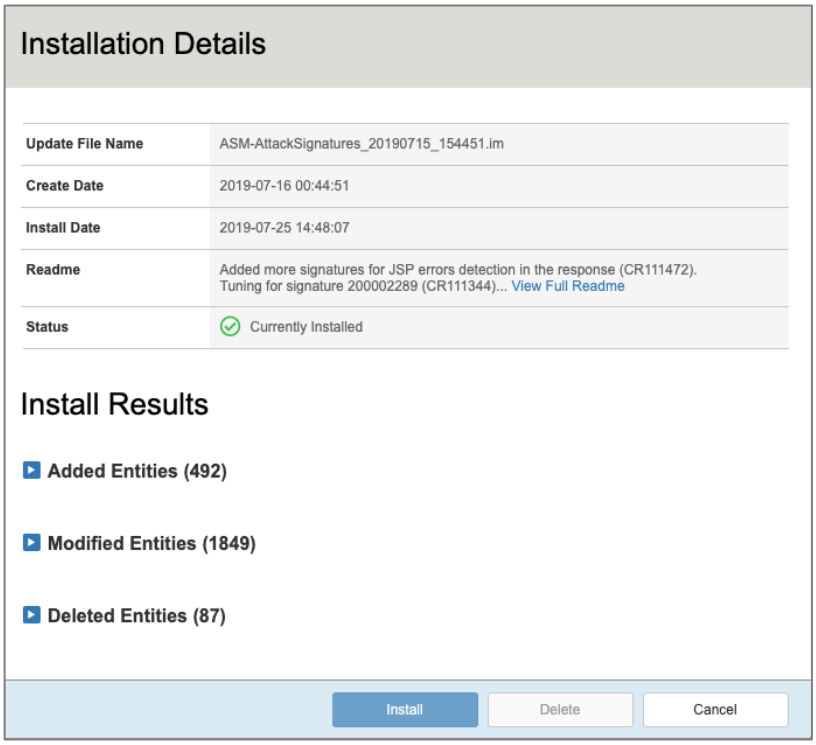
(7) Install が完了すると以下のようになります。×を推し、画面を閉じます。



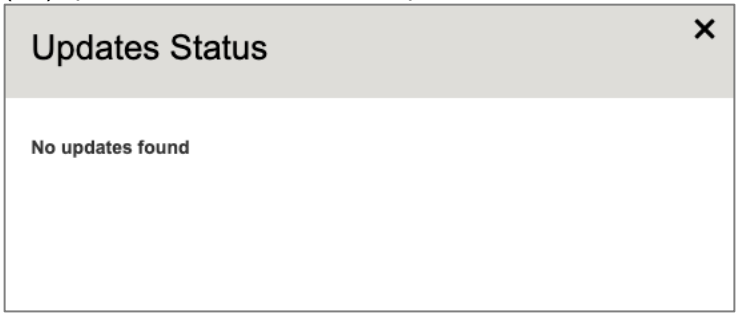
(8) Currently Instaslled ステータスのシグネチャをクリックします。



(9) Update された Signature の情報が表示されます。各 Entity をクリックすると、該当するシグネチャー覧が確認できます。



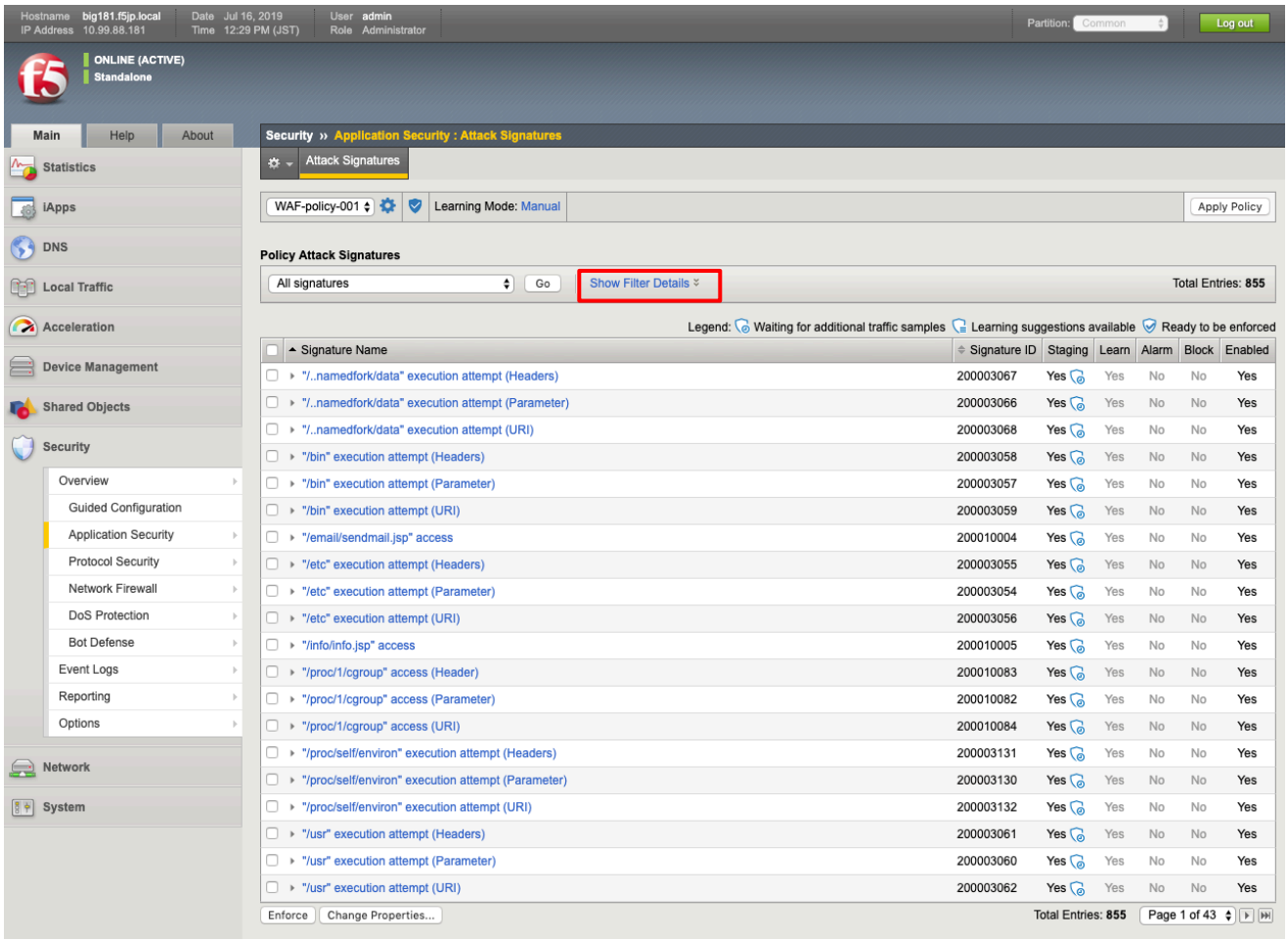
(10)Update がない場合は Install Updates をクリックしても以下のように表示されます。



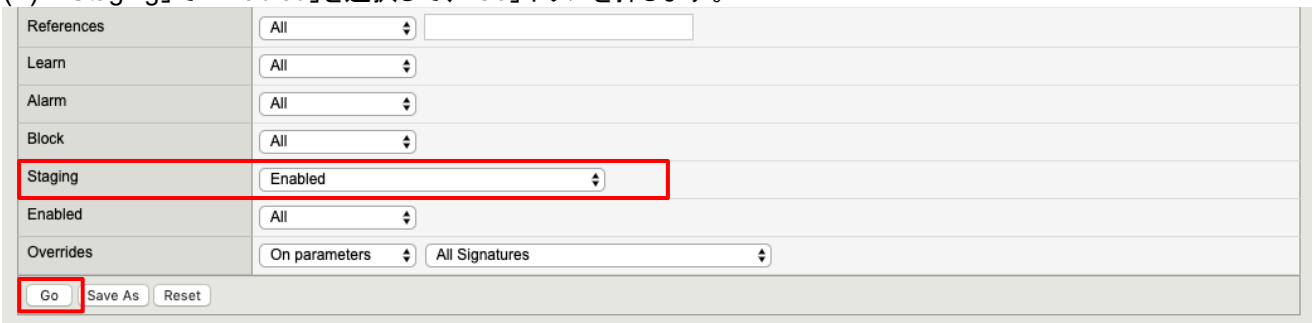
15.4. 更新されたシグネチャのステージング状態確認

更新されたシグネチャがステージングになっているかどうかの確認方法を示します。

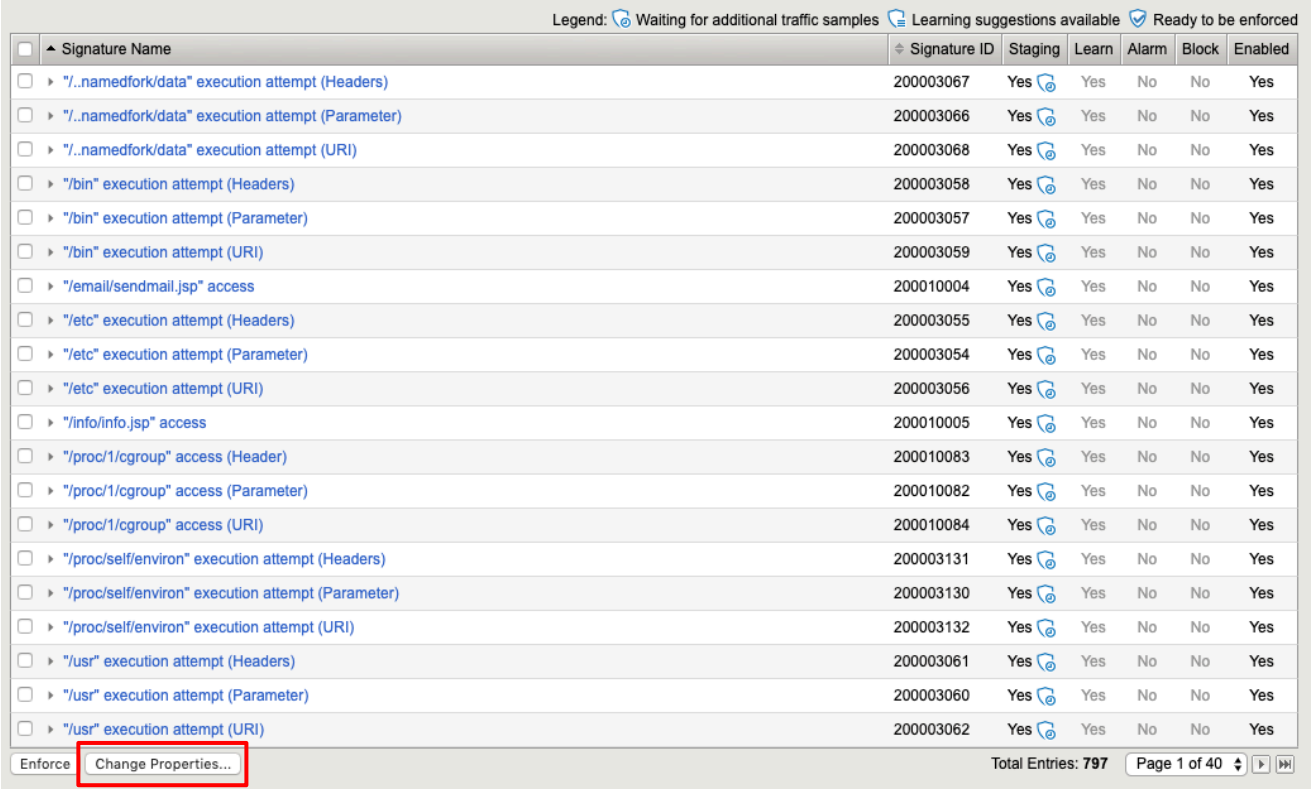
(1) 「Security」→「Application Security」→「Attack Signatures」で表示された画面の「Show Filter Details」をクリックします。



(2) 「Staging」で「Enabled」を選択して、「Go」ボタンを押します。



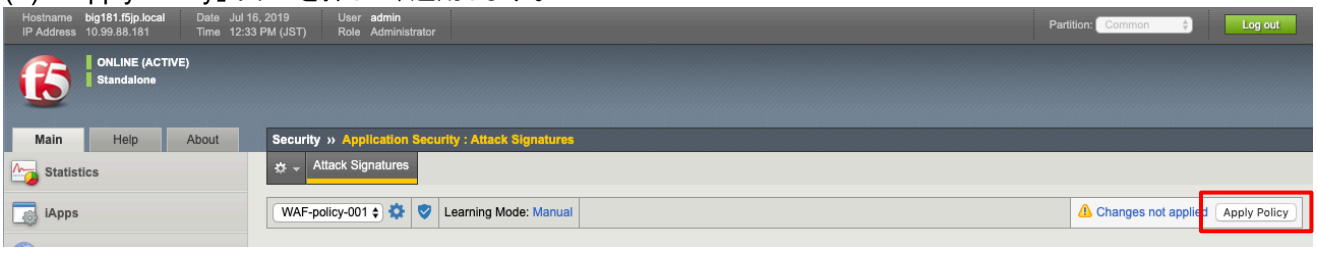
(3) 以下のように、追加分が「Staging」になっています。



通常はこのまま Staging モードで運用し、特に問題がないようでしたら、Learn/Alarm/Block モードに戻します。
その場合には、もう一度この画面に戻り、同じように表示フィルタを実施します（Staging のみ表示します）。

(4) 「Change Properties」ボタンを押し、以下のように設定します。

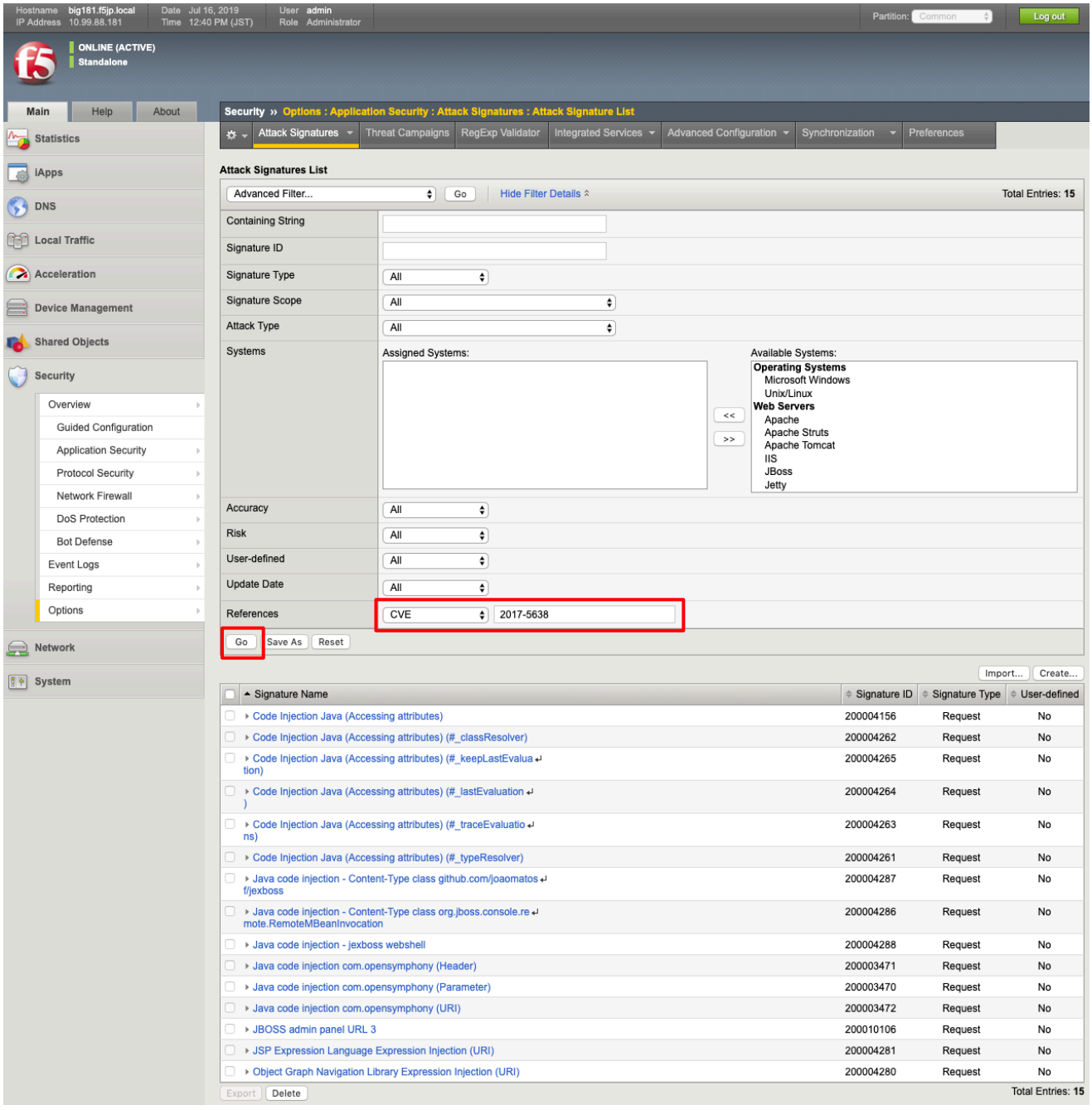
(5) 「Apply Policy」ボタンを押して、適用します。



15.5. CVE 番号によるシグネチャの検索

各シグネチャが CVE に対応しているか確認することが可能です。

(1) 「Security」→「Options」→「Application Security」→「Attack Signatures」→「Attack Signature List」で表示された画面の「Show Filter Details」をクリックすると、以下のような画面が表示されます。References から CVE を選択し、CVE 番号を入力して 「Go」ボタンを押します。



上記の CVE 番号（CVE-2017-5638）は、Apache Struts 2 の脆弱性に対応したシグネチャー覧であることを表示しています。Adv.WAF では、1 つの CVE 番号に関連したシグネチャが複数存在していることがあります。

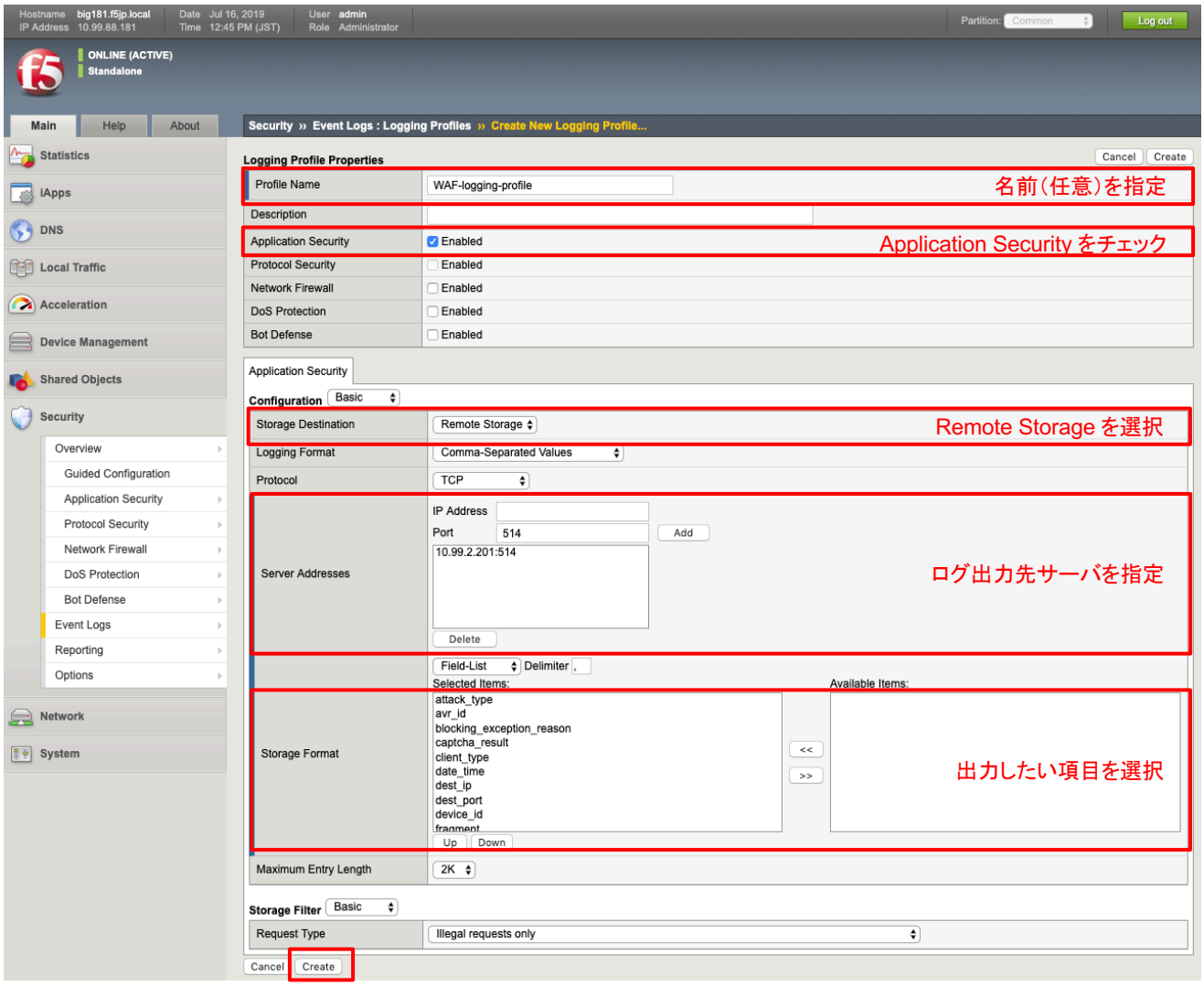
16. [ご参考]ログの外部出力の設定

※F5 ハンズオンでは実施しません

ログの出力を Adv.WAF 内部ではなく、外部サーバへのみ行う設定方法を示します。この設定により、Adv.WAF 内部でのログ出力のディスク I/O が発生しないため、その分パフォーマンスは良くなります。

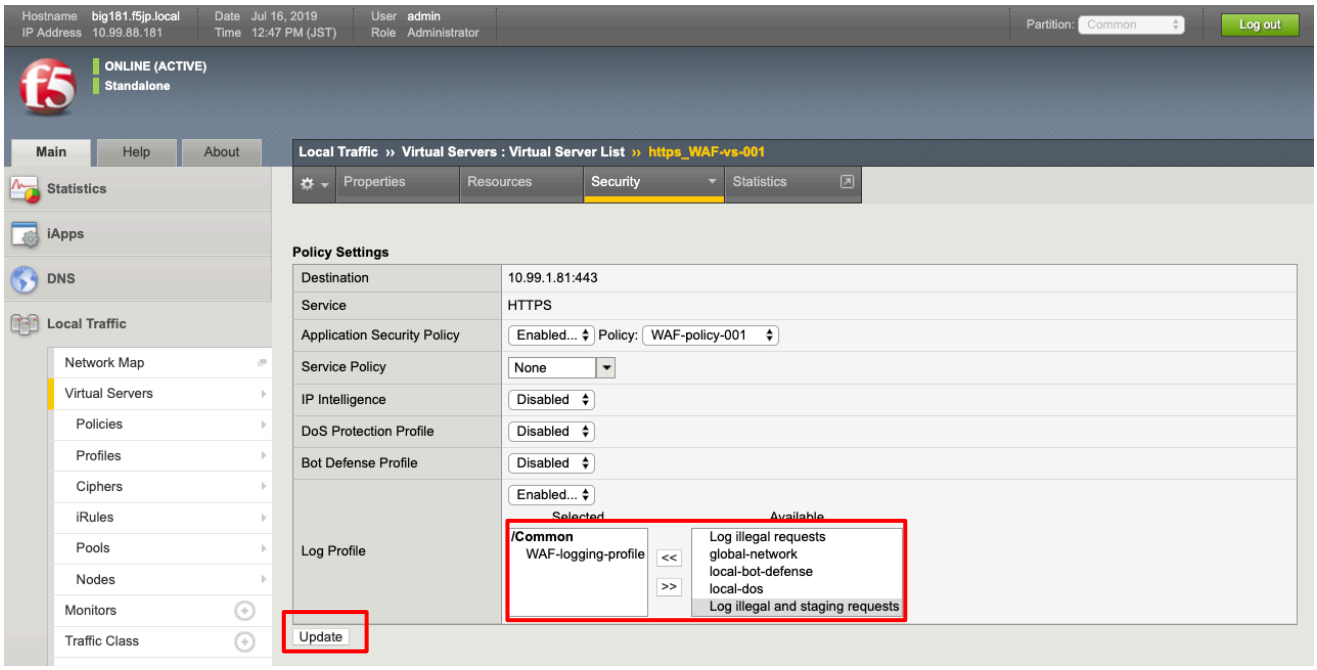
16.1. Logging Profile の作成

(1) 「Security」→「Event Logs」→「Logging Profiles」で表示された画面右上の「Create」ボタンを押し、現れた画面で以下のように設定します。



16.2. Logging Profile の VS への適用

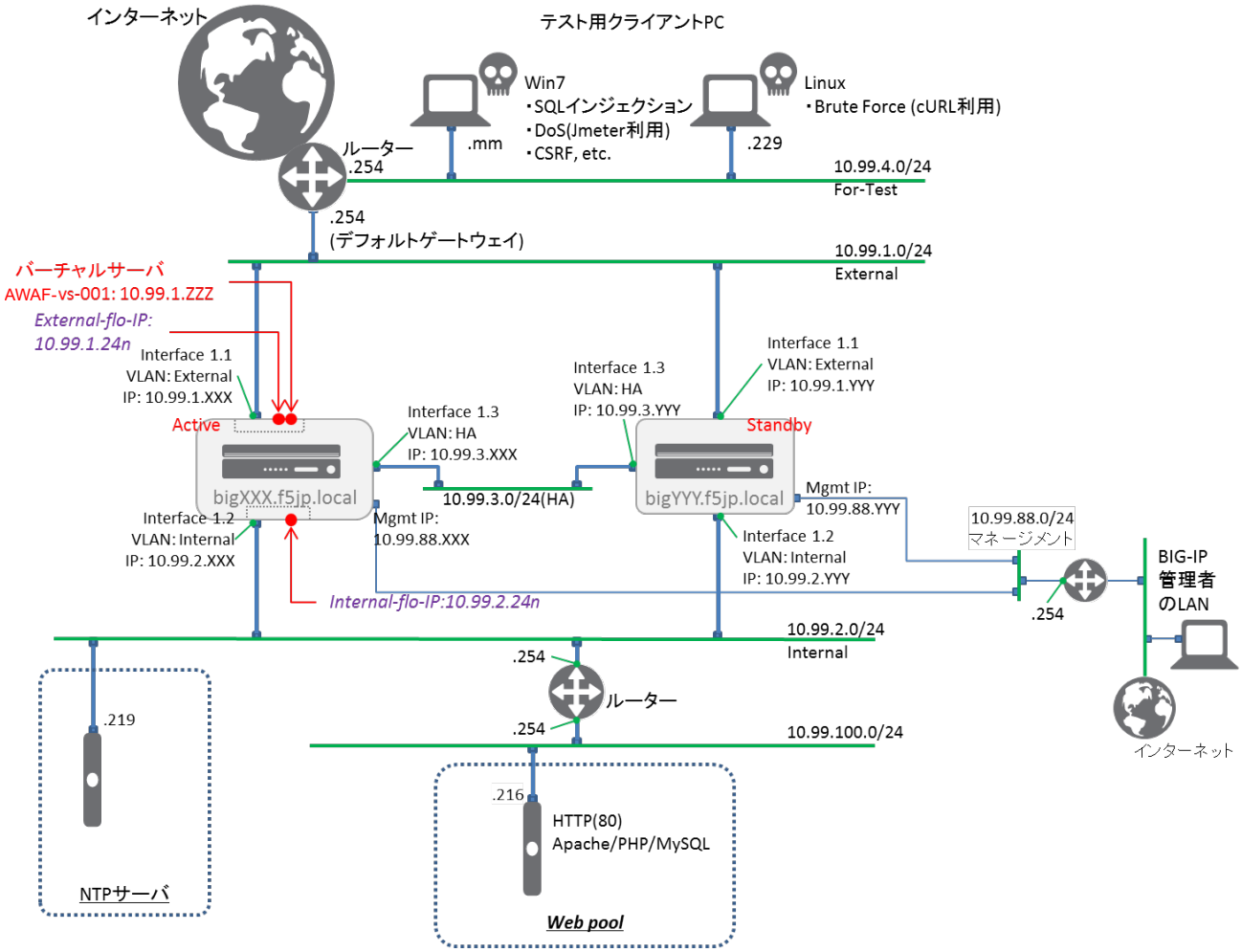
(1) 各バーチャルサーバへ、設定した Logging Profile を適用します。「Local Traffic」→「Virtual Servers」→「Virtual Sever list」で表示された該当 VS をクリックし、「Security」タブの「Policies」をクリックすると、以下の画面が表示されます。作成した profile を選択して、Update ボタンを押します。



17. 冗長化

17.1. 冗長化のネットワークサンプル

もう一台 Adv.WAF を追加して、L3 構成の冗長化設定を行います。



Adv.WAF 間の HA (High Availability) VLAN は、冗長化の制御プロトコルをやり取りする専用の VLAN です。
External や Internal VLAN を利用することも可能ですが、HA 専用の VLAN を追加することを推奨しています。

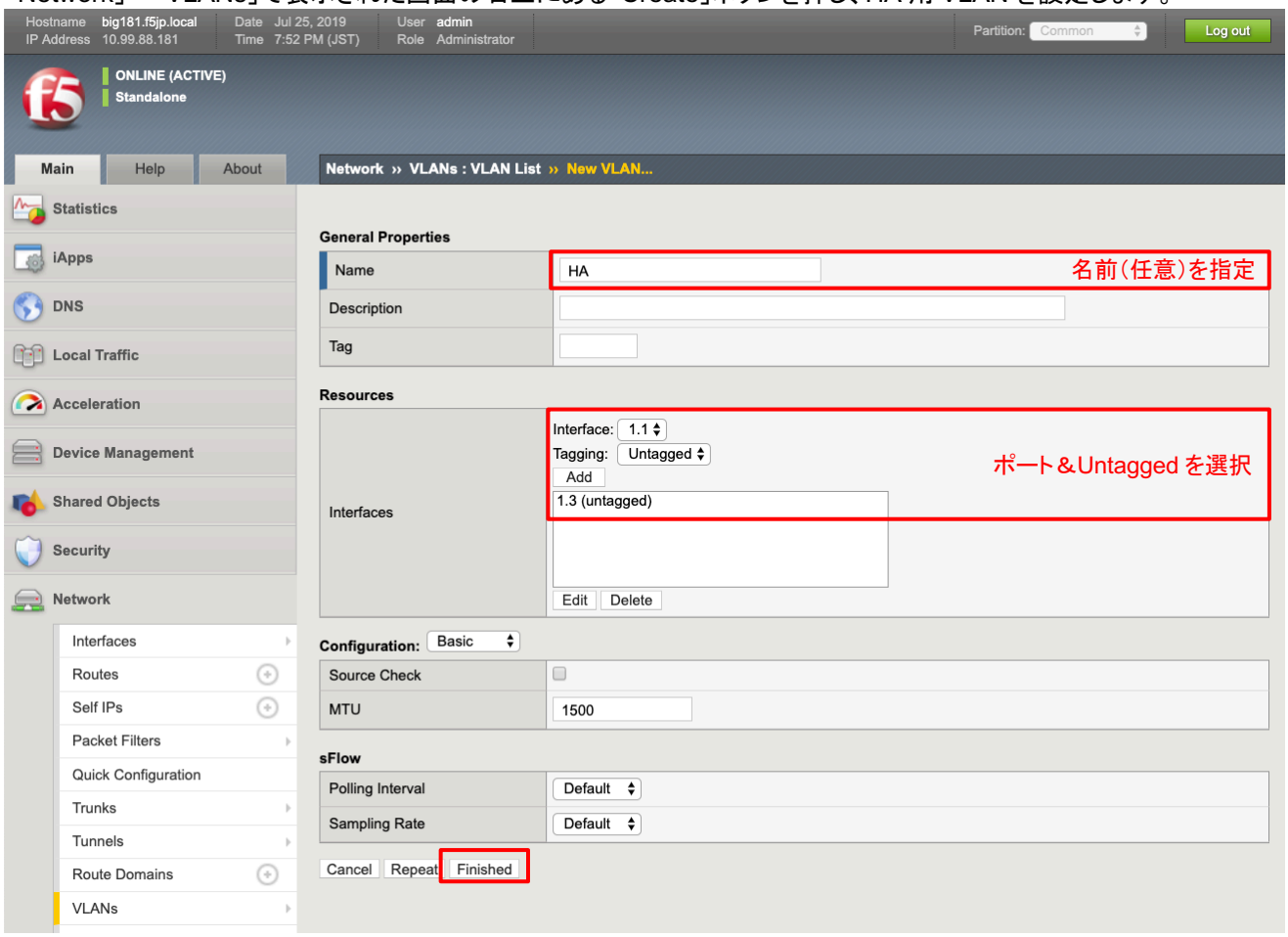
よって、本構成においては、HA VLAN を追加しています。

17.2.Active 機の設定

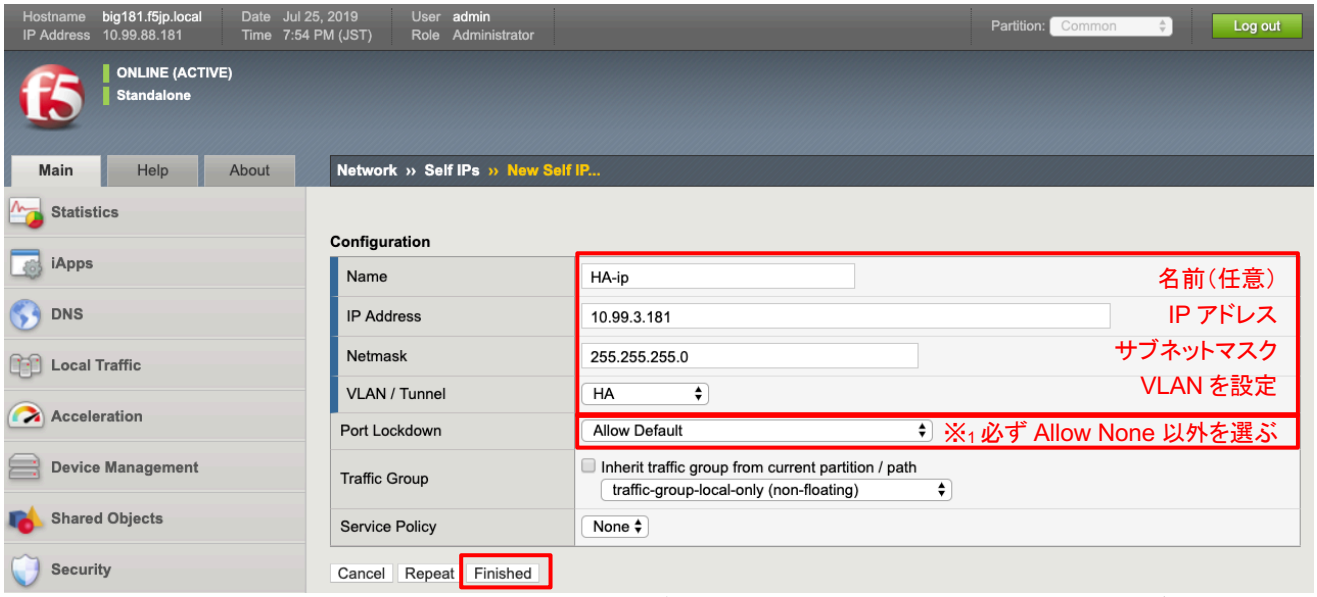
管理（マネージメント）IP アドレスの 4 オクテット目の数字が大きい方が、デフォルトで”Active”となります。
F5Lab 環境では、bigXXX.f5jp.local が Active 機となります。

17.2.1. VLAN、SelfIP の設定

(1) HA VLAN の設定
「Network」→「VLANs」で表示された画面の右上にある「Create」ボタンを押し、HA 用 VLAN を設定します。



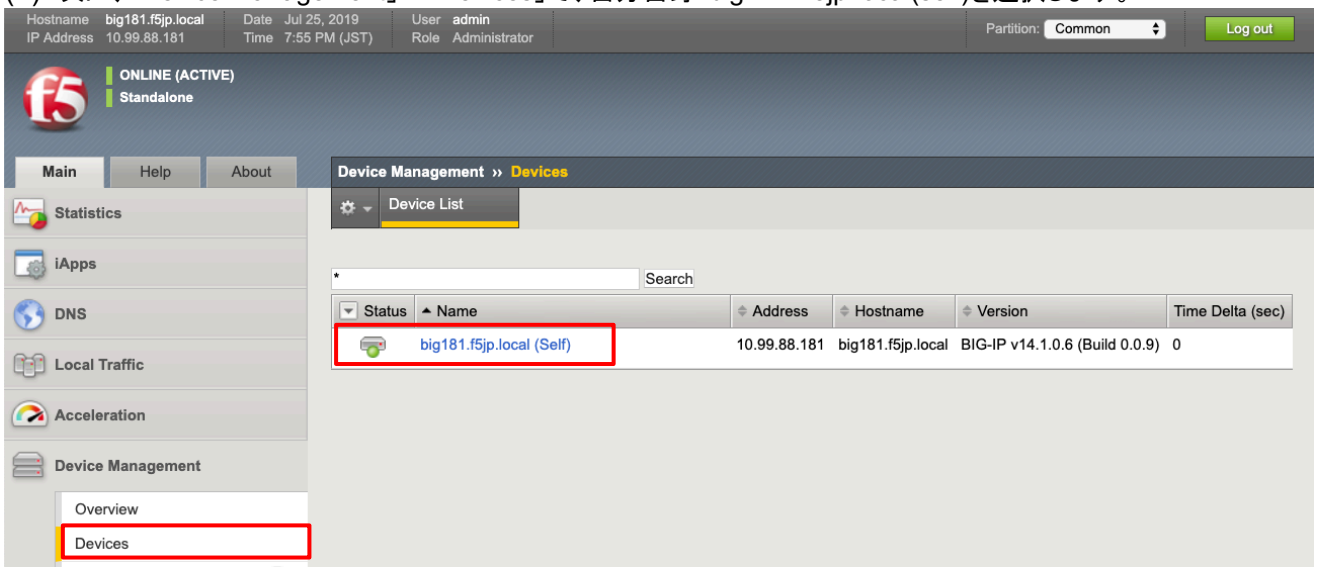
(2) HA VLAN の IP 設定
「Network」→「Self IPs」で表示された画面の右上にある「Create」ボタンを押し、HA 用 VLAN の IP を設定します。



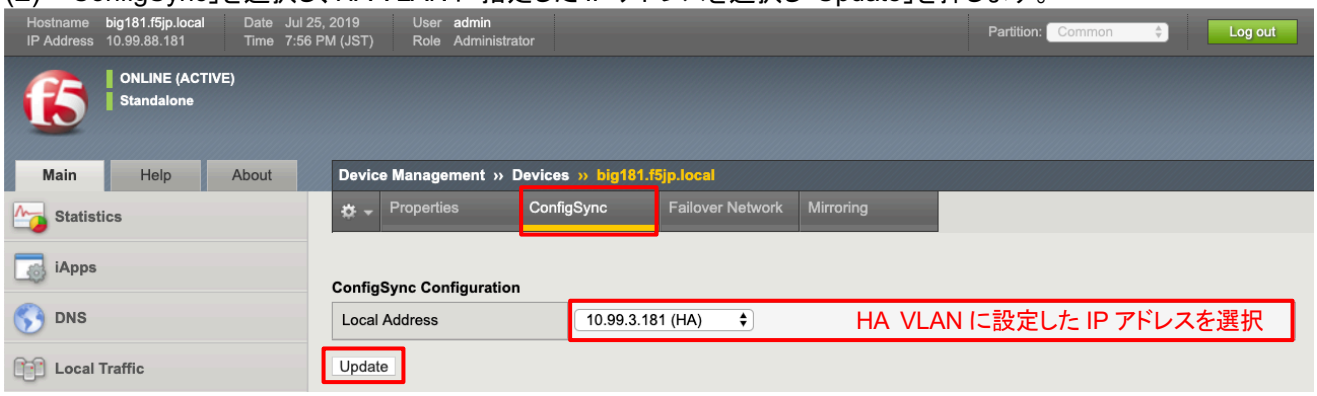
※1 Allow None を選ぶと HA の通信も止めてしまい、HA が組めません。（ここでは Allow Default を選びます）

17.2.2. Device の設定

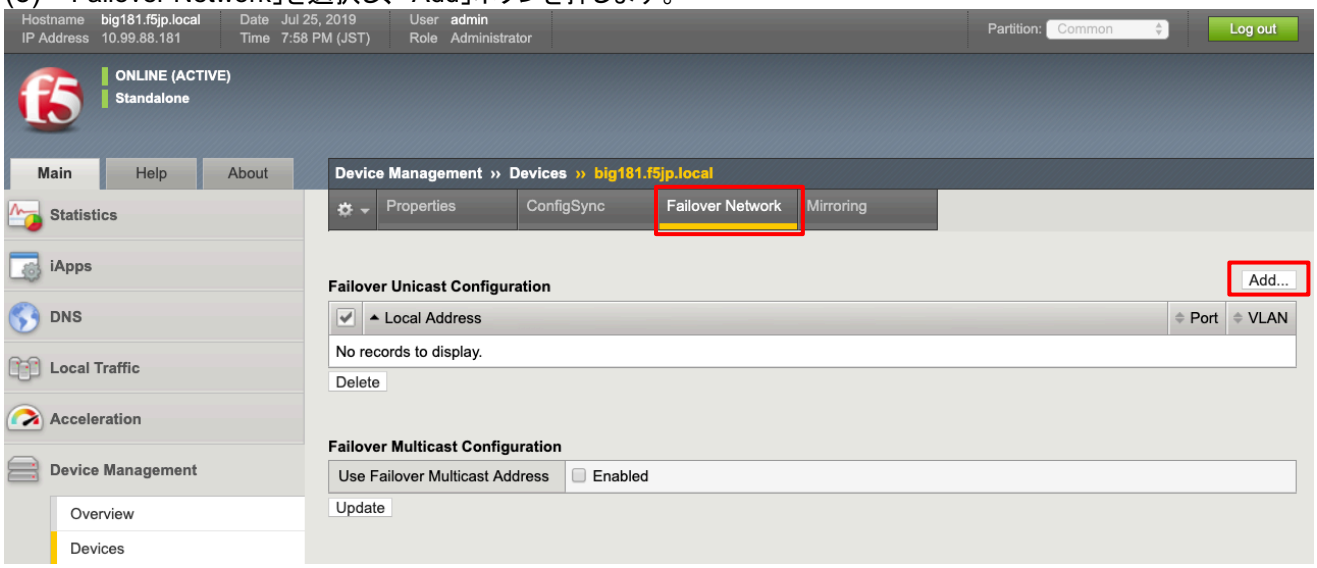
(1) 次に、「Device Management」→「Devices」で、自分自身：bigXXX.f5jp.local(self)を選択します。



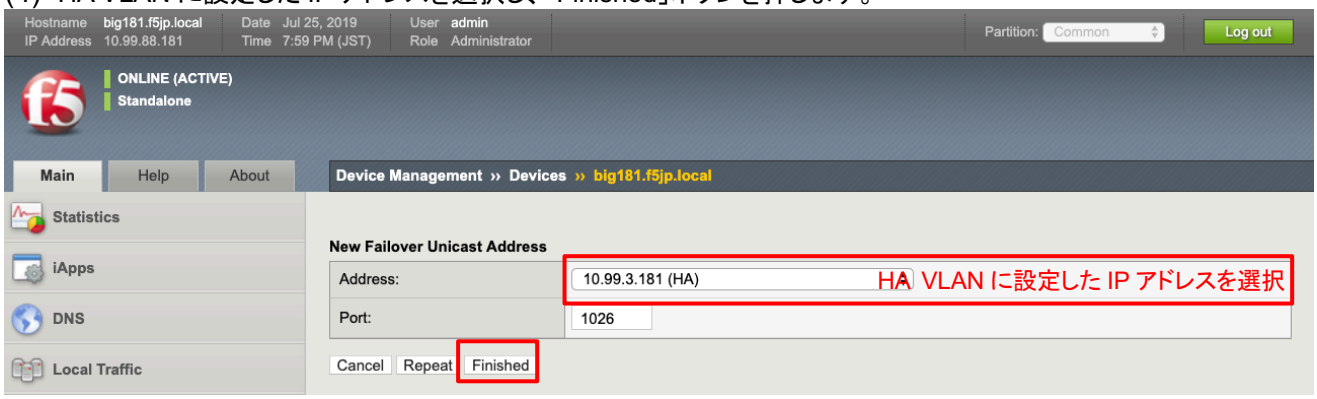
(2) 「ConfigSync」を選択し、HA VLAN に指定した IP アドレスを選択し「Update」を押します。



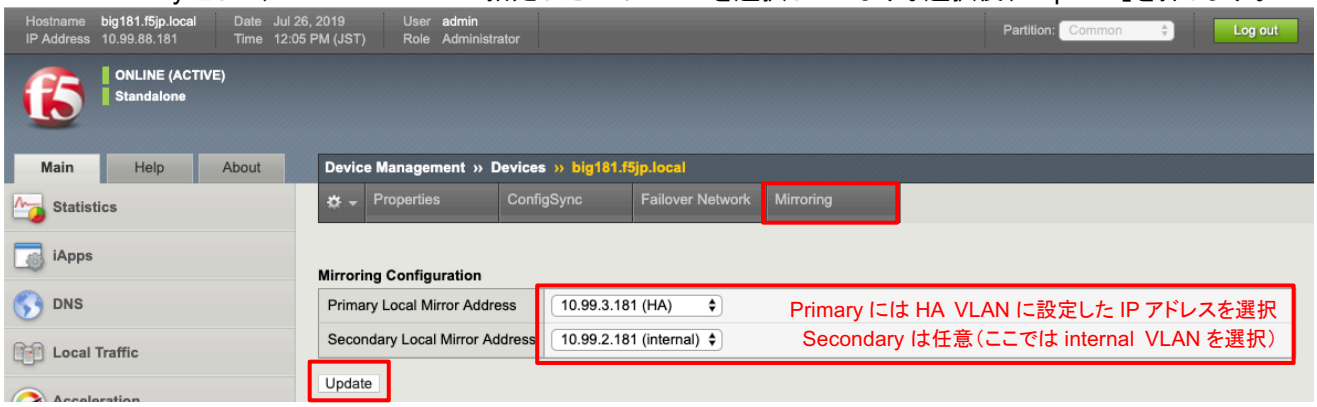
(3) 「Failover Network」を選択し、「Add」ボタンを押します。



(4) HA VLAN に設定した IP アドレスを選択し、「Finished」ボタンを押します。

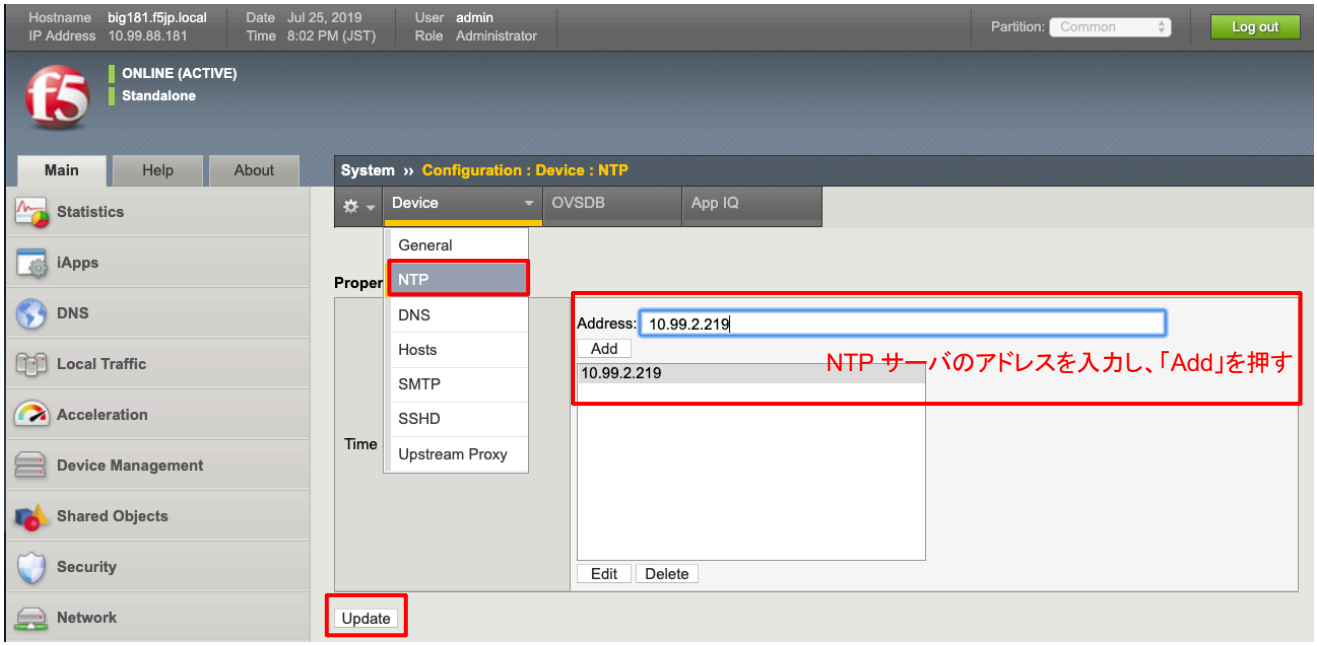


(5) 「Mirroring」を選択し、HA VLAN に指定した IP アドレスをプライマリに指定します。任意ですが、ここでは Secondary として、Observe VLAN に指定した IP アドレスを選択しています。選択後、「Update」を押します。



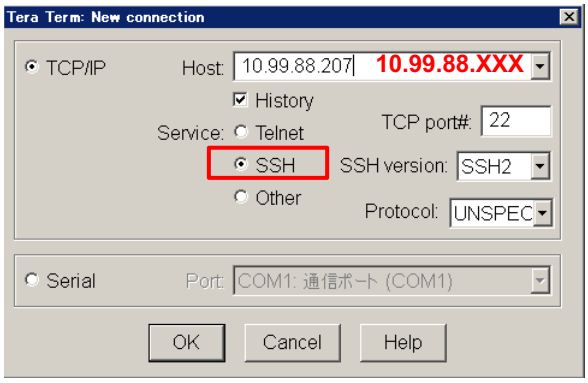
17.2.3. 時刻同期（NTP）設定

(1) 「System」→「Configuration」→「Device」→「NTP」を選択します。Address 欄に、NTP サーバの IP アドレスを入力し、「Add」ボタンを押します。

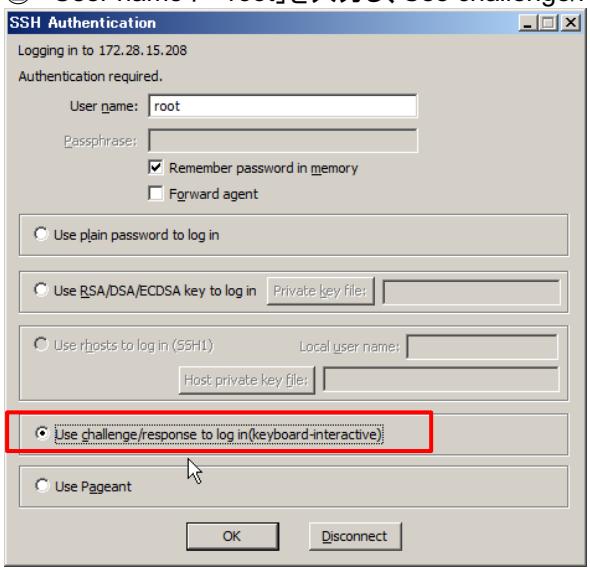


[ご参考] NTP 同期状態の確認
NTP 同期状態の確認は、コマンドラインから実施します。以下に、Tera Term を利用した場合の確認方法を示します。

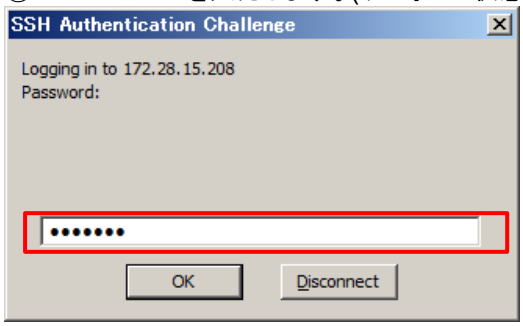
① SSH でログインします。



② User name に「root」を入力し、Use challenge/response to log in をチェックします。



③ パスワードを入力します。(デフォルト状態のパスワードは「default」です)



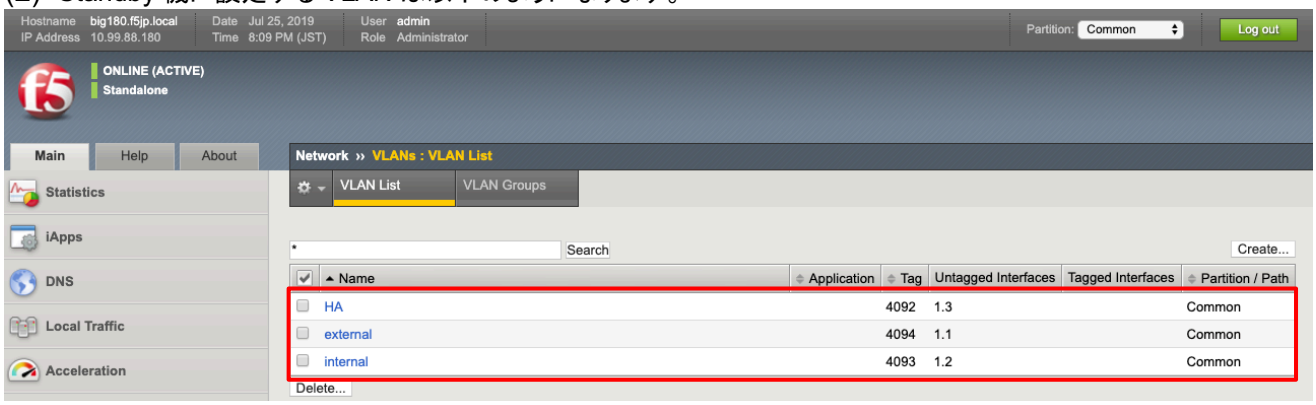
④ SSH アクセスが完了したら、「ntpq -np」を実行します。 先頭に「＊」がついていれば、同期が完了しています。
(同期完了状態になるまで、時間がかかる場合があります。)

```
[root@bigXXX:Active:Standalone] config # ntpq -np
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*10.99.2.219     133.243.238.243  2 u   23   64    1    0.805   -0.003   0.933
```

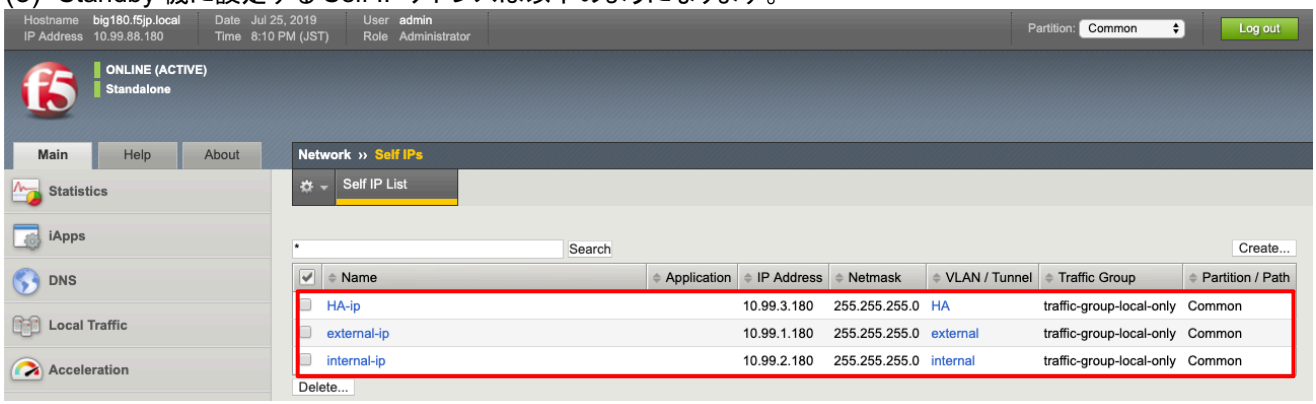
17.3. Standby 機(bigYYY.f5jp.local)の設定

(1) Active 機での VLAN、Self IP、Devices の設定と同様の設定を Standby 機に対しても行います。

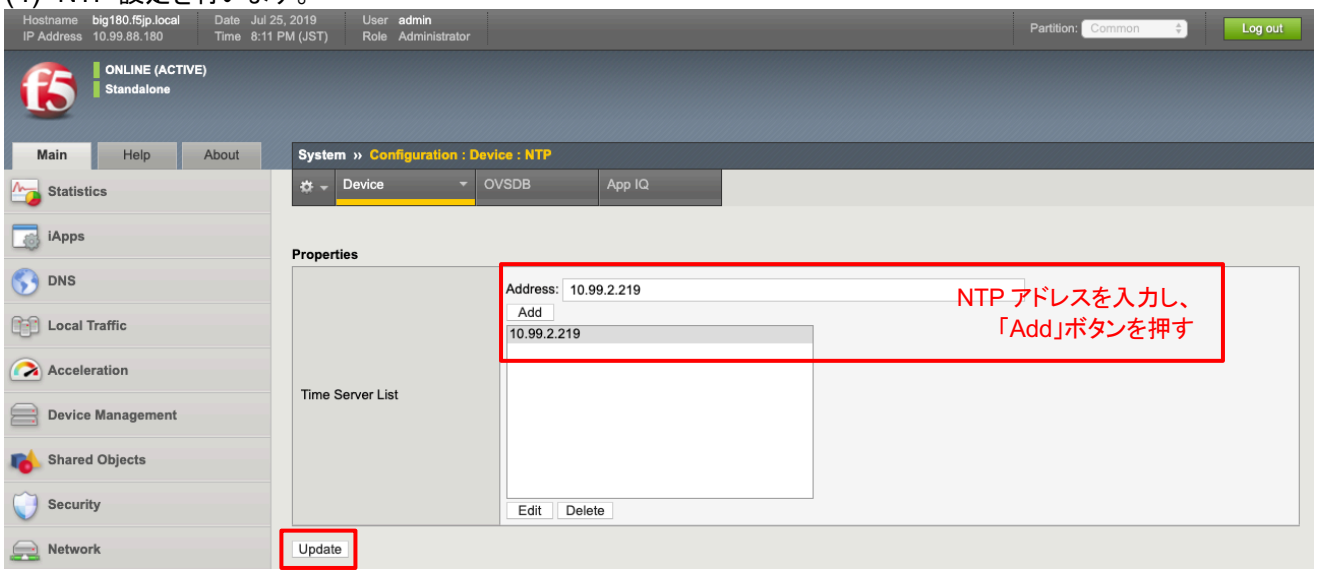
(2) Standby 機に設定する VLAN は以下のようになります。



(3) Standby 機に設定する Self IP アドレスは以下のようになります。



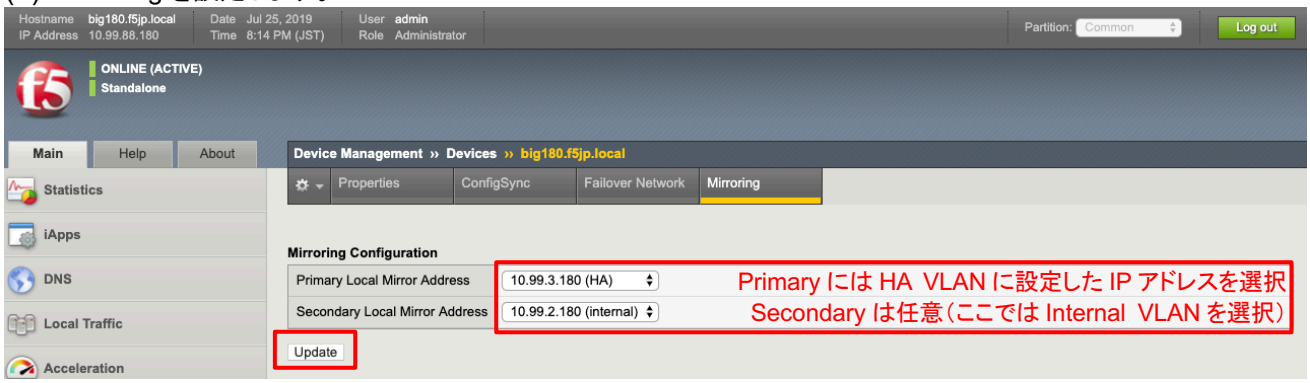
(4) NTP 設定を行います。



(5) 「Device Management」→「Devices」で、自分自身：bigYYY.f5jp.local(self)を選択し、Active 機同様に、Device Connectivity の設定を行います。
まずは ConfigSync を設定します。

(6) Failover Network を設定します。

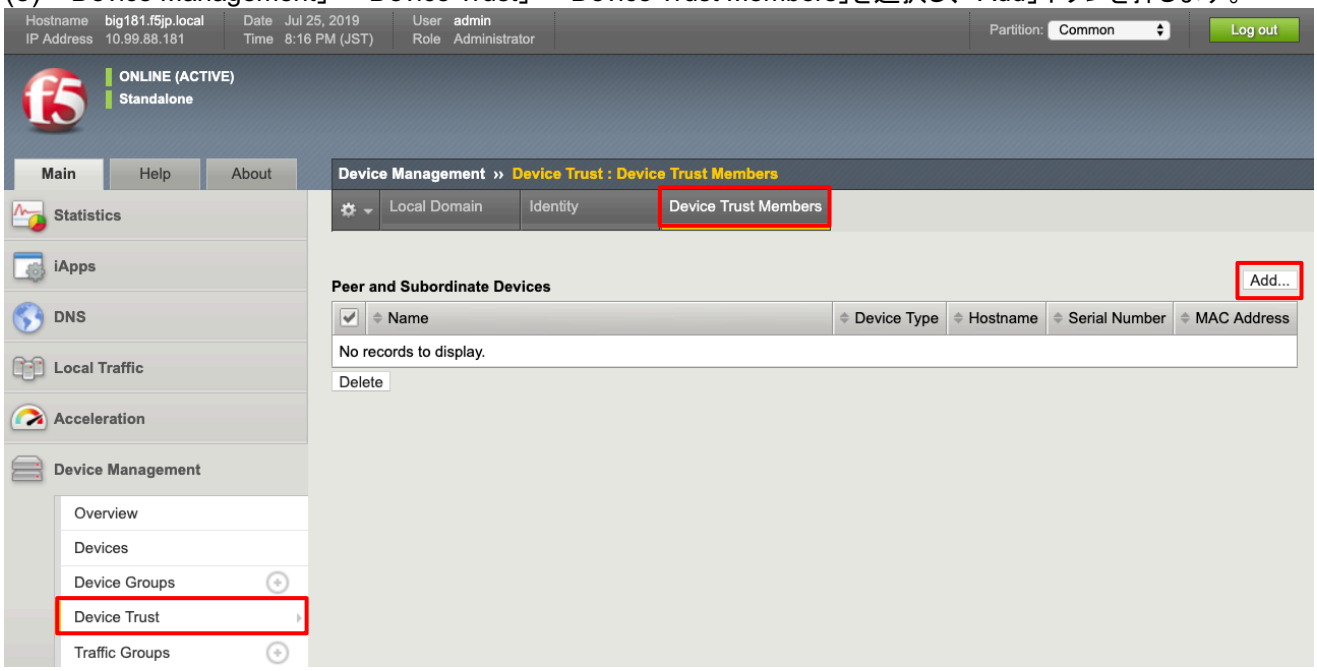
(7) Mirroring を設定します。



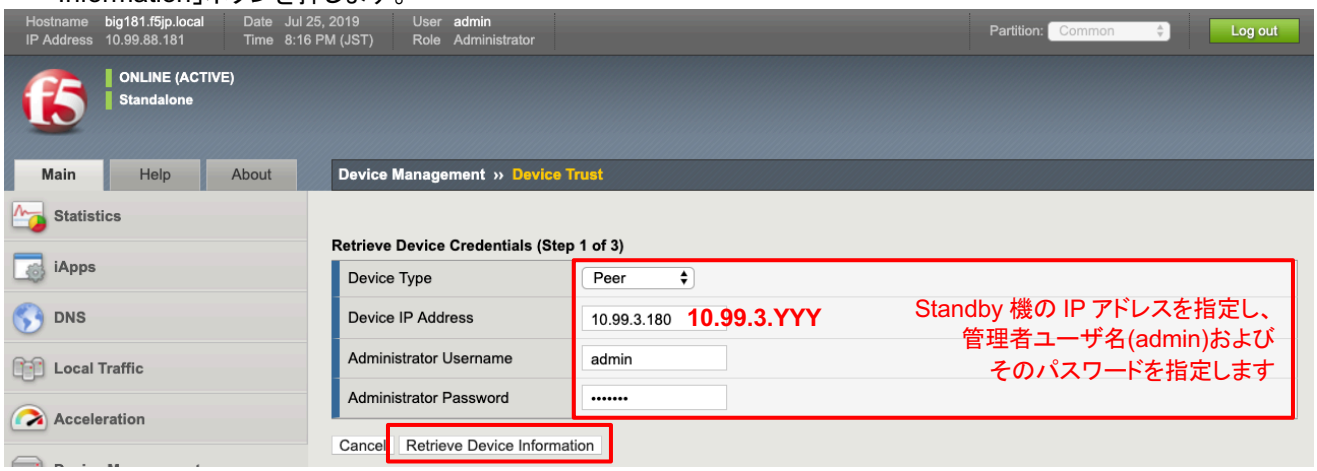
17.4. デバイストラストの設定

(1) デバイストラスト設定にて、冗長化する機器間で信頼関係を結びます。
(2) 本項は、Active 機：bigXXX.f5jp.local からのみ、設定します。

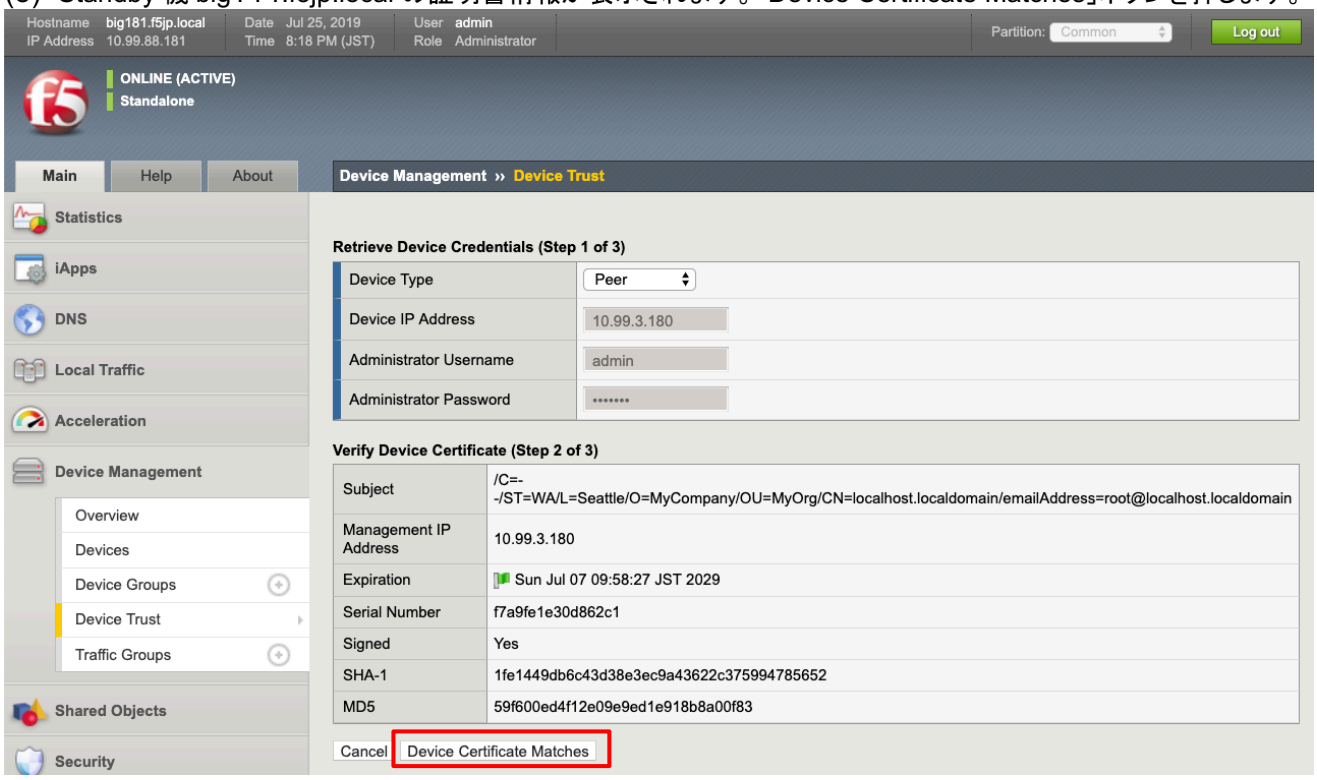
(3) 「Device Management」→「Device Trust」→「Device Trust Members」を選択し、「Add」ボタンを押します。



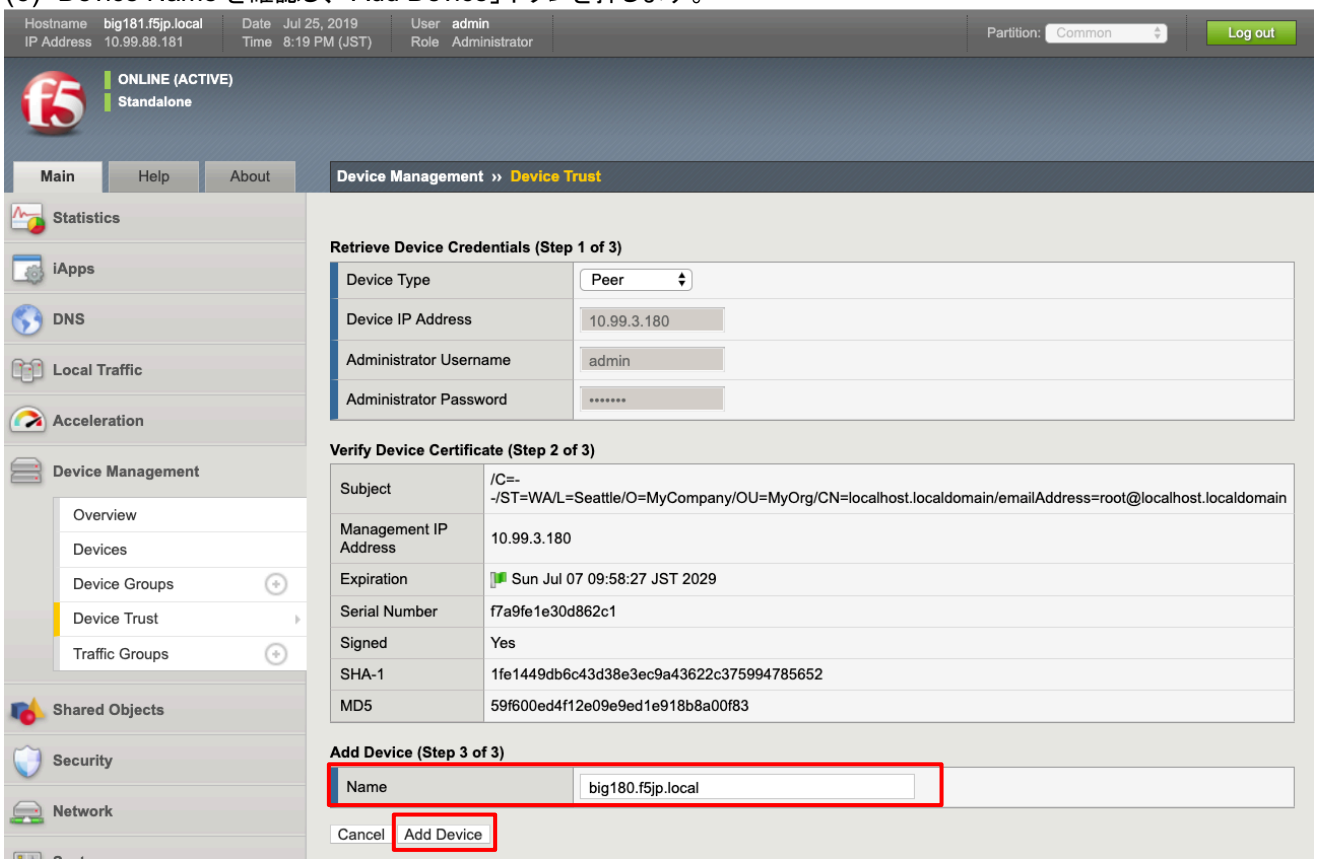
(4) Standby 機：bigYYY.f5jp.local の IP アドレスと管理者 ID(Admin)とパスワードを指定します。「Retrieve Device Information」ボタンを押します。



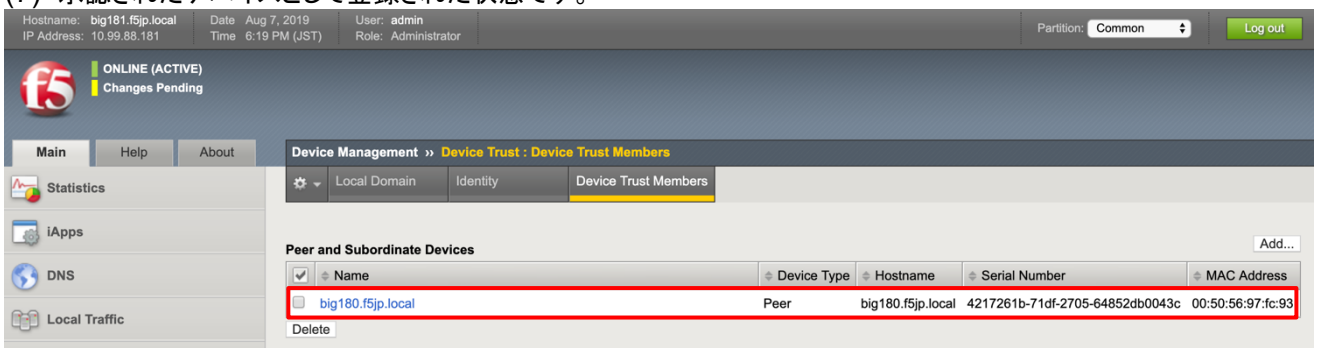
(5) Standby 機 bigYYY.f5jp.local の証明書情報が表示されます。「Device Certificate Matches」ボタンを押します。



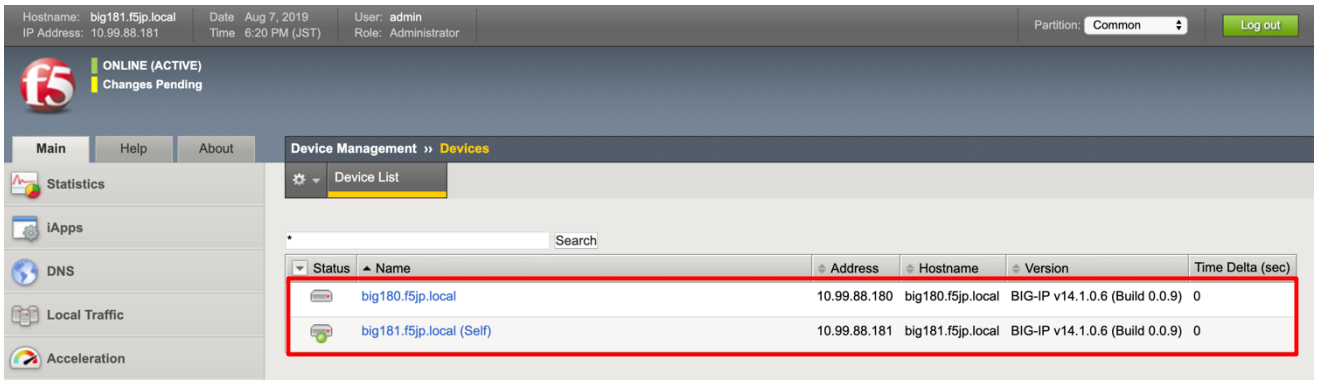
(6) Device Name を確認し、「Add Device」ボタンを押します。



(7) 承認されたデバイスとして登録された状態です。



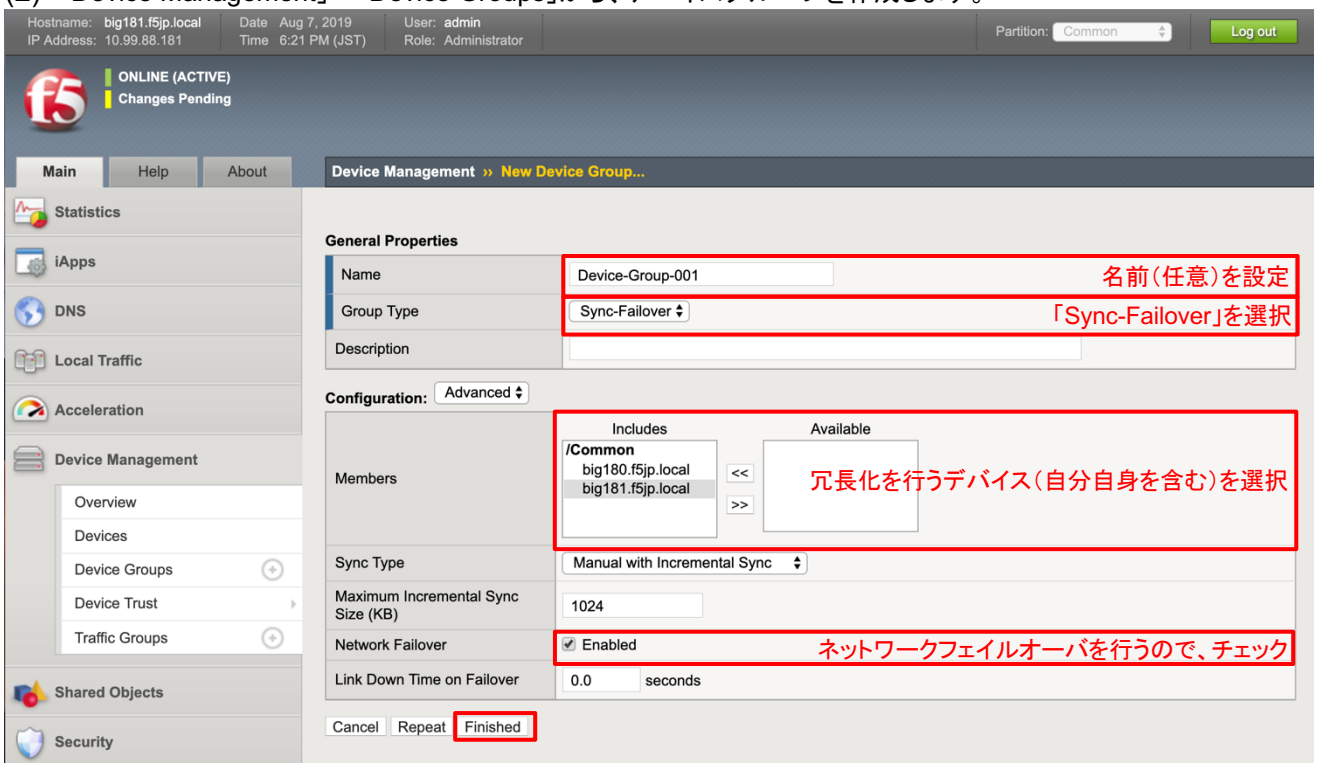
(8) 「Device Management」→「Devices」で見ると、(self)に加え、Standby 機：bigYYY.f5jp.local も表示されます。
（ここは確認のみです。）



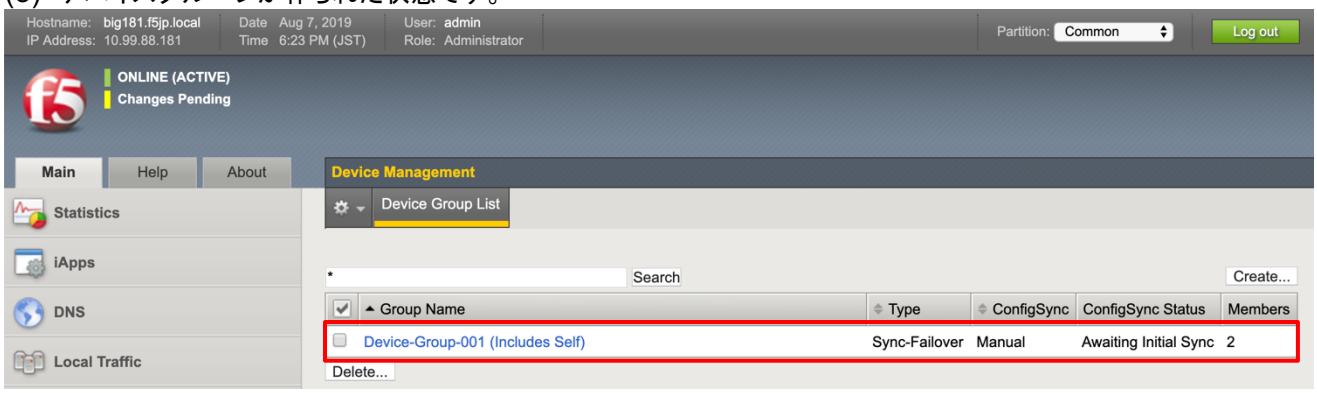
17.5.デバイスグループの設定

(1) デバイスグループは、デバイストラストで信頼関係を結んだ機器の間で、どの機器間で冗長化を行うかの指定です。デバイストラストは Adv.WAF ×3 台以上で構成することも可能で、例えば、(1)と(2)で冗長化を行い、(2)と(3)はコンフィグ同期のみ行う、という組合せが可能となっています。この組み合わせをデバイスグループで指定します。
2 台で冗長化を行う場合はデバイスグループの組み方をあまり意識する必要はありませんが、設定は必要です。

(2) 「Device Management」→「Device Groups」から、デバイスグループを作成します。



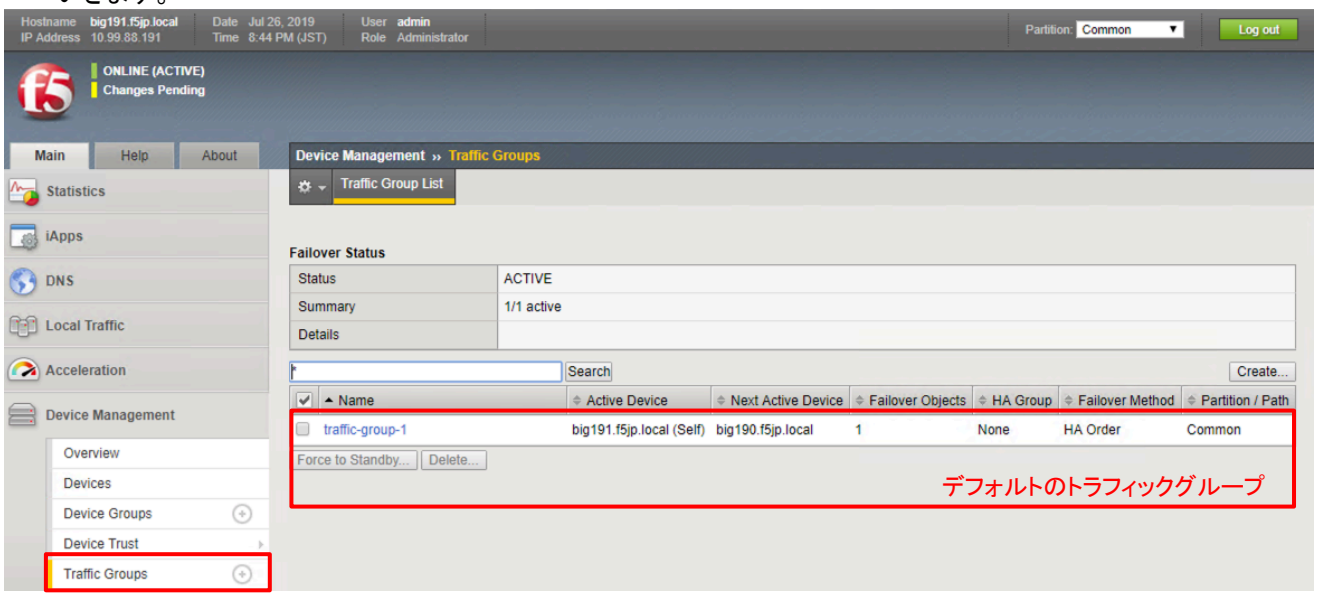
(3) デバイスグループが作られた状態です。



17.6.トラフィックグループの設定

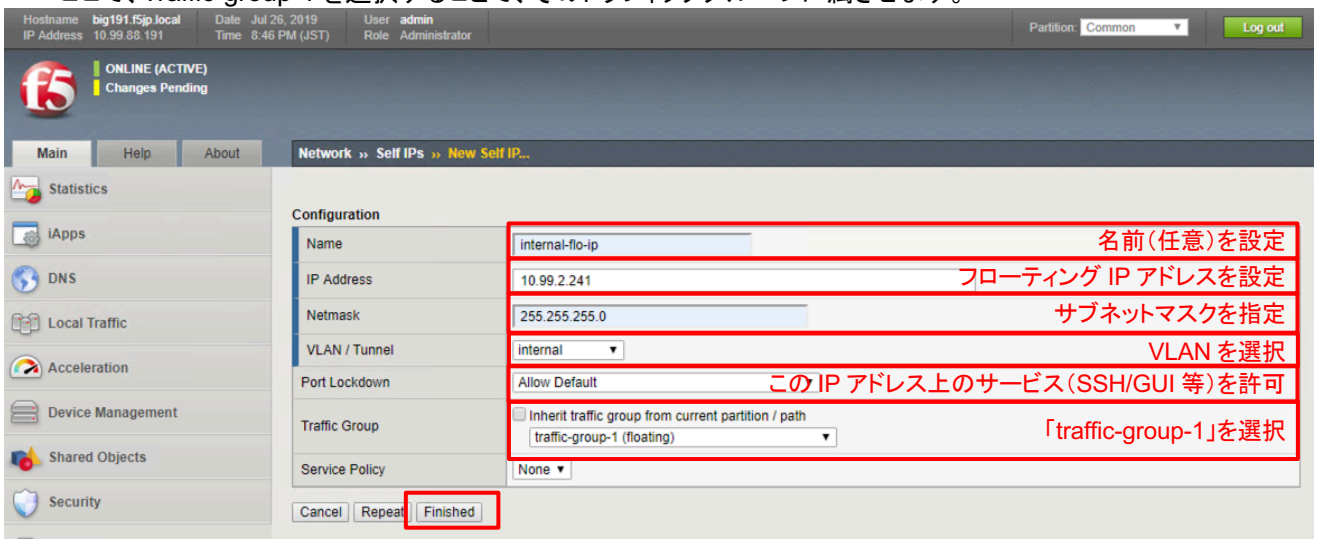
トラフィックグループは、デバイスグループ内で移動するオブジェクトの集合です。
主に、Virtual Server と共用 IP(Floating IP)がトラフィックグループのオブジェクトです。

(1) 「Device Management」→「Traffic Groups」を確認します。デフォルトで、「Traffic-group-1」という名前のトラフィックグループが存在しています。以降、この Traffic-group-1 に対して、Floating IP および Virtual Server を割当てていきます。

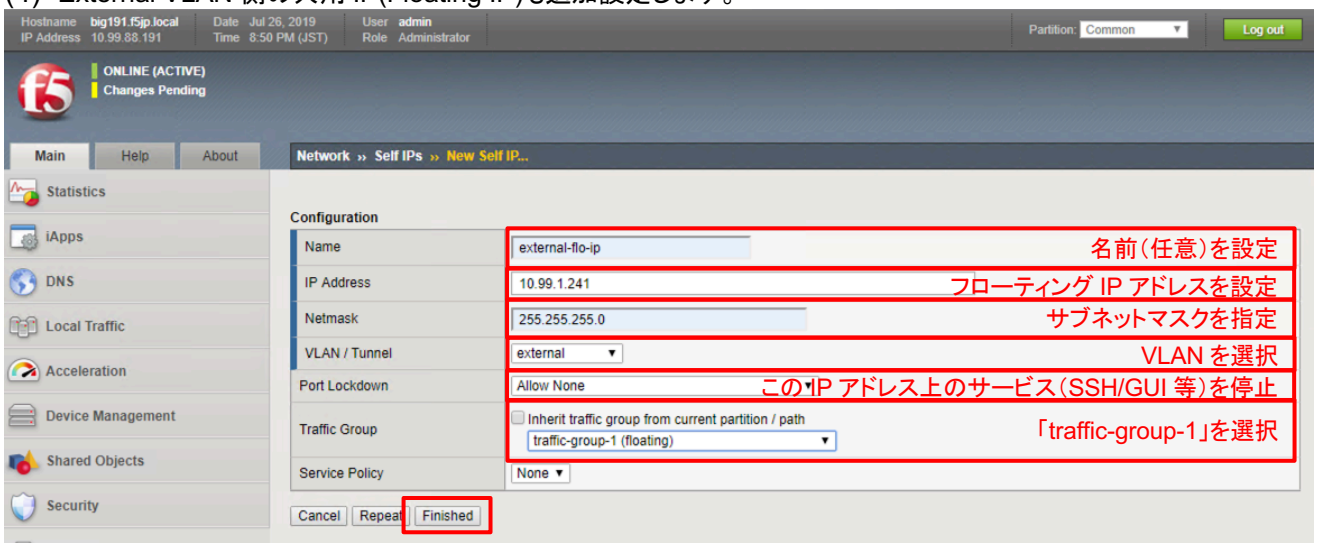


(2) Floating IP（共有 IP）を追加設定します。Floating IP は、Active 機ダウン時に Standby 機が引き継ぐ、自身に設定された IP アドレス(Self IP)を指します。実サーバは、この IP アドレスをデフォルトゲートウェイに指定することで、Active/Standby の切り替わり発生時にも、即座に通信を再開できます。

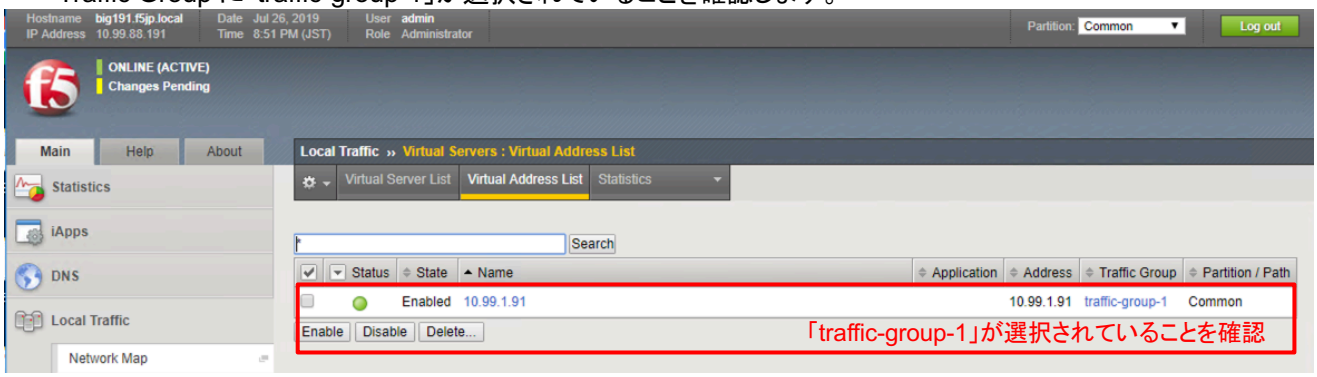
(3) Internal VLAN 側の共用 IP(Floating IP)を追加設定します。「Network」→「Self IPs」から設定します。
ここで、Traffic-group-1 を選択することで、そのトラフィックグループに属させます。



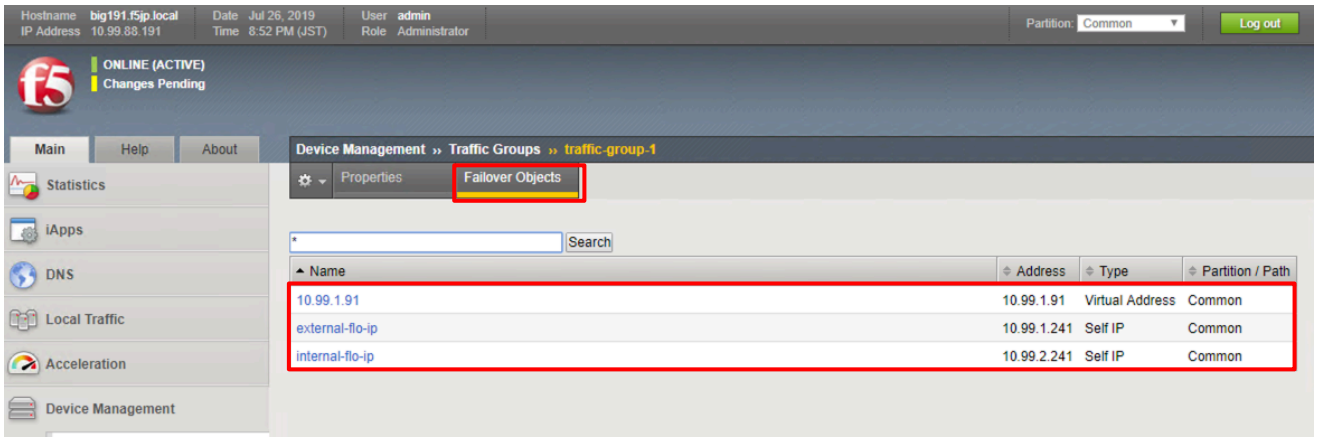
(4) External VLAN 側の共用 IP(Floating IP)も追加設定します。



(5) 「Local Traffic」→「Virtual Servers」→「Virtual Address List」を選択します。
Traffic Group に「traffic-group-1」が選択されていることを確認します。



(6) 「Device Management」→「Traffic Groups」の Traffic-group-1 をクリック→「Failover Objects」タブをクリックして、中身を確認すると、フェイルオーバーオブジェクトは以下のようになっています。



(7) Adv.WAF 固有の設定です。Adv.WAF ポリシーを同期するために必要となります。
「Security」→「Options」→「Application Security」→「Synchronization」→「Application Security Synchronization」で表示された画面で、設定した Device-Group を選択し、「Save」ボタンを押します。

17.7. ConfigSync

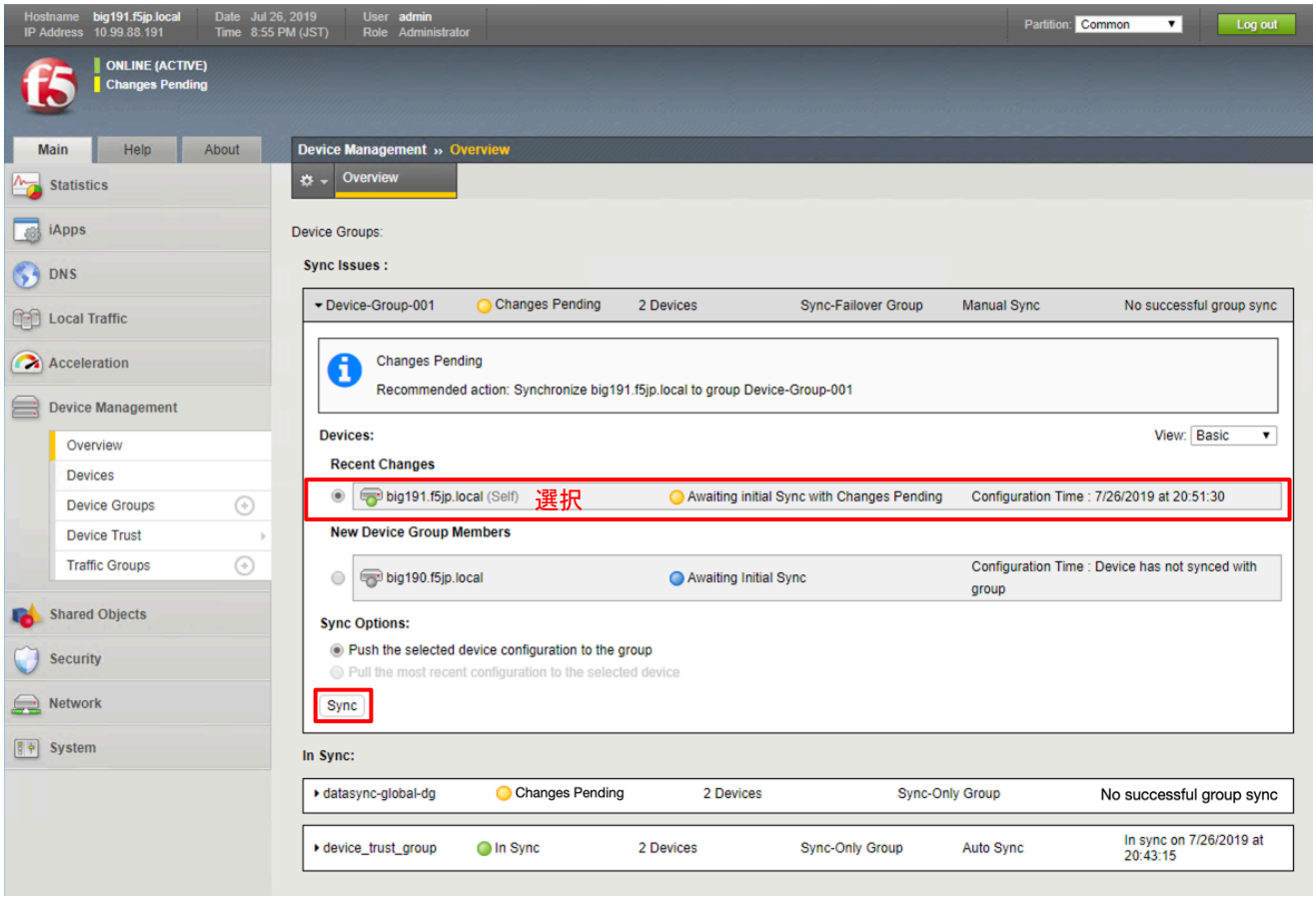
Active 機：bigXXX.f5jp.local にのみに行った設定を、Standby 機：bigYYY.f5jp.local に同期するために、ConfigSync を行います。

「Device Management」→「Overview」を選択すると、3つの Device Group が作成されています。

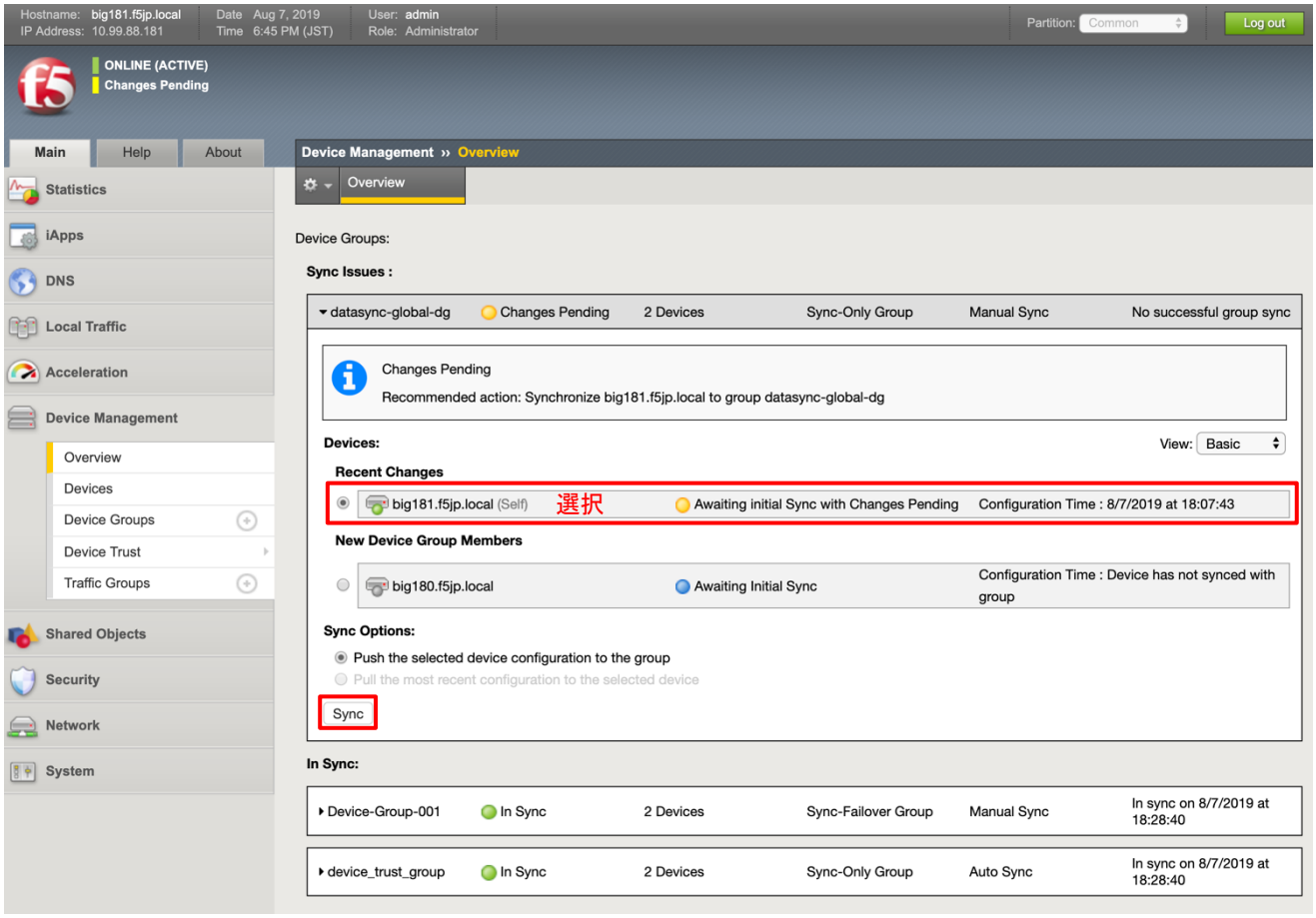
Ⅰ	device_trust_group	trust group に peer を設定すると、システムによって自動的に作成されます。peer の基本情報を Sync します。
Ⅱ	datasync-global-dg	Adv.WAF 特有の Device Group で、システムによって自動的に作成されます。Script や暗号鍵を Sync します。
Ⅲ	Device-Group-001（任意の名前）	前項で作成したユーザ設定領域のデータを Sync します。

Ⅰは自動で Sync されますが、ⅡとⅢはデフォルトでマニュアル Sync の設定となっています。Ⅱは初回設定時、または UCS ファイルからデータをリストアした後に Sync を実施する必要があります。Ⅲはマニュアル設定の場合、ユーザ設定領域の変更を行った場合に Sync を実施します。

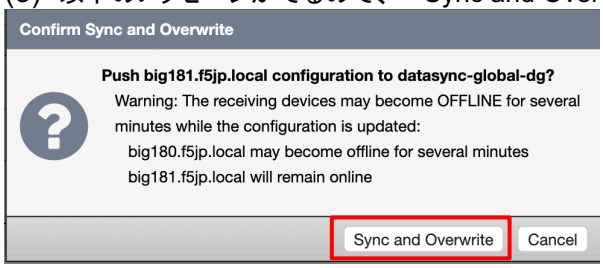
(1) Device-Group-001（任意の名前）を Sync します。アクティブ機(bigXXX.f5jp.local)を選択し、「Sync」ボタンを押すことで、コンフィグ同期が行われます。



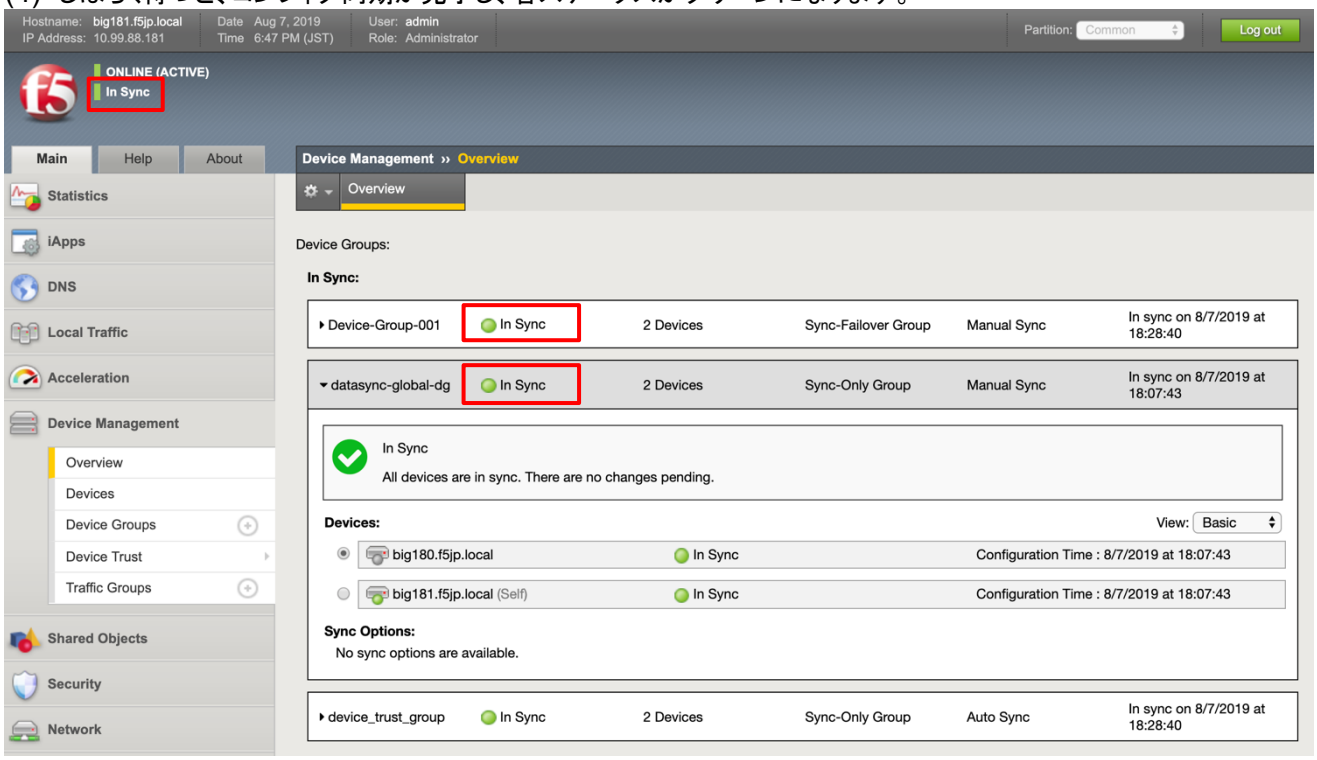
(2) datasync-global-dg を Sync します。アクティブ機(bigXXX.f5jp.local)を選択し、「Sync」ボタンを押すことで、コンフィグ同期が行われます。



(3) 以下のメッセージがでるので、「Sync and Overwrite」を選択します。

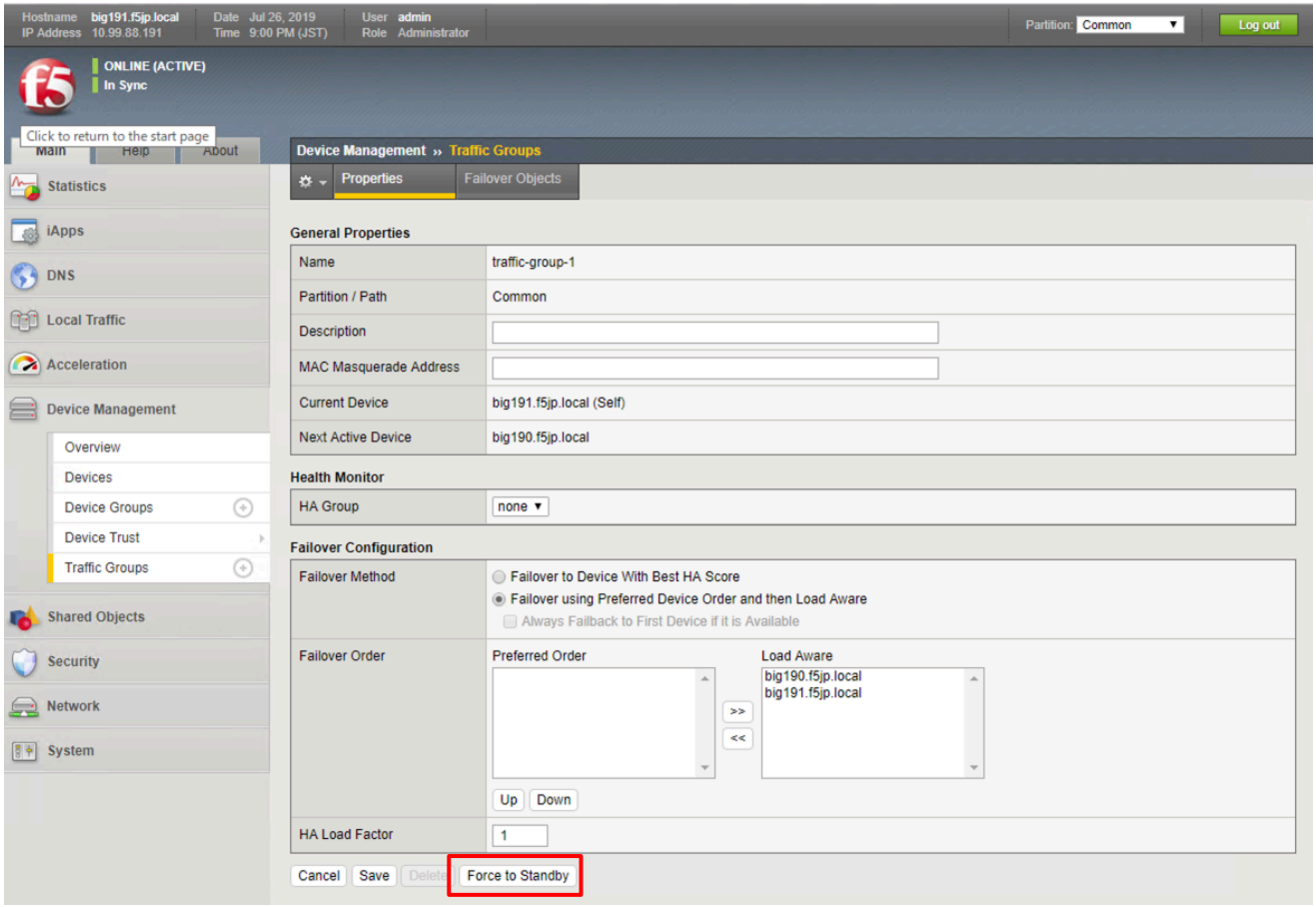


(4) しばらく待つと、コンフィグ同期が完了し、各ステータスがグリーンになります。

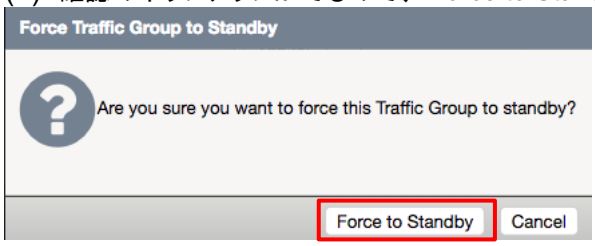


17.8. Traffic-group-1 の Active/Standby の切替え

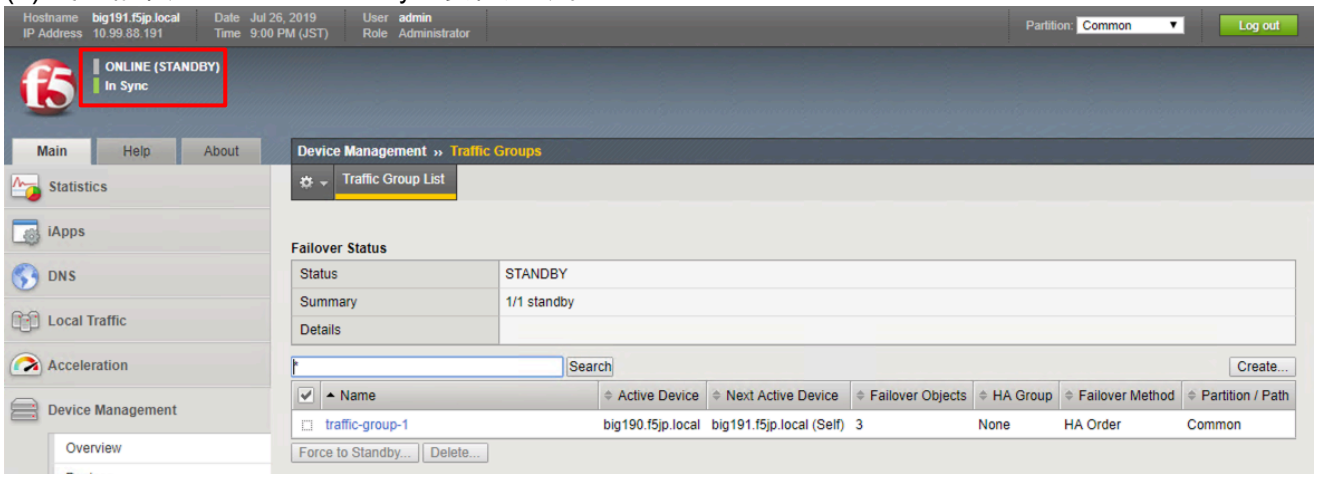
(1) アクティブ機の「Device Management」→「Traffic Groups」から Traffic-group-1 を選択し、「Force to Standby」ボタンを押します。



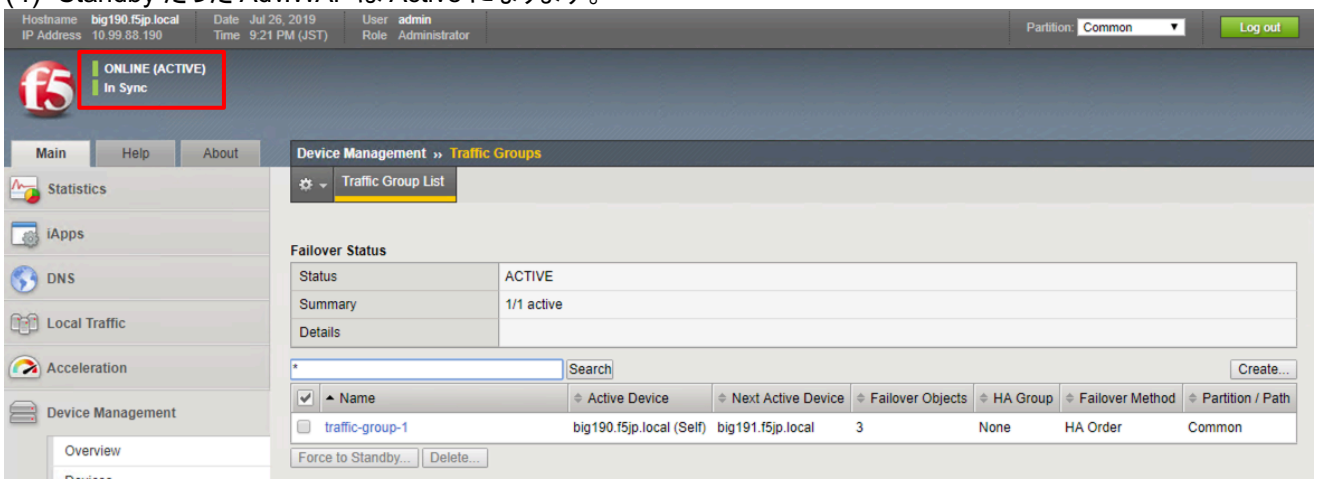
(2) 確認のポップアップがでるので、「Force to Standby」ボタンを押します。



(3) その結果、Active から Standby に変わります。



(4) Standby だった Adv.WAF は Active になります。



(5) テスト用クライアントから、作成した Virtual Server へ Web ブラウザでアクセスし、Web 画面が表示されることを確認します。

(6) Traffic-Group-1 を、再度切替え、クライアントからの通信が復旧するかを確認してください。

18. おわりに

Adv.WAF セットアップに関しては以上で終了となります。

BIG-IP シリーズ製品ラインナップにおいては、ソフトウェアモジュールライセンスを追加することで、サーバ負荷分散はもちろんのこと、広域負荷分散やリモートアクセス機能、ネットワークファイアウォール機能など、アプリケーションアクセスを最適化する為の多彩な機能が使用できるようになります。
詳細は各種 WEB サイトにてご確認いただくか、F5 公式販売代理店にお問い合わせください。

＜F5 ネットワークス WEB サイトの紹介＞

F5 ネットワークスジャパン総合サイト
https://f5.com/jp

F5 のセキュリティ ソリューション
https://f5.com/jp/products/security

AskF5：ナレッジベース総合サイト（英語）
http://support.f5.com/kb/en-us.html

DevCentral：F5 ユーザコミュニティサイト（英語：アカウント登録が必要です）
https://devcentral.f5.com/

F5 公式販売代理店リスト
https://www.f5.com/ja_jp/partners/jp-find-a-partner

以上

更新日：2019-10-15
