F5 SSLO セットアップガイド
i-FILTER 編

2021 年 08 月 16 日

目次:

最終更新日: 2021 年 8 月 16 日

F5 SSL Orchestrator (SSLO) と WEB フィルタリング製品である i-FILTER との連携手順をご紹介致します。(本ガイドは、一度 BIG-IP を設定したことのある方を対象とした内容となっております。)

第 1 章

はじめに

こちらのページでは、以下のオフィシャルなドキュメントの補足となる資料や、複数の機能を組合せてソリューションを実現する方法をご紹介いたします。

F5 のオフィシャルなドキュメントはこちらにございます。

- AskF5: https://support.f5.com/csp/home
- F5 Cloud Docs: https://clouddocs.f5.com/
- F5 DevCentral（コミュニティ）: https://devcentral.f5.com/

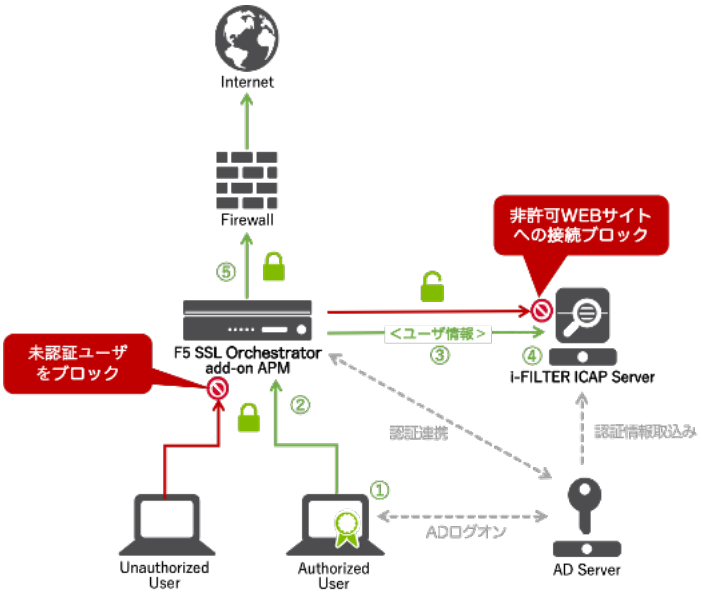
1.1 F5 SSLO (L3 Explicit Proxy) と i-FILTER ICAP 版の連携設定

本章では、SSLO の可視化ゾーンに i-FILTER ICAP 版を配置し、ICAP プロトコルで連携する構成について、ご紹介致します。

1.1.1 F5 SSLO と i-FILTER ICAP 版連携の流れ

F5 SSLO と i-FILTER　ICAP 版の連携の流れは以下の通りです。

1. クライアントがパソコン (AD) にログオンします。
2. SSLO でユーザ認証を実施し (NTLM 認証、Kerberos 認証の場合は自動的に認証が実施されます)、SSL 通信を復号します。
3. WEB 接続先情報と SSLO で認証が成功したユーザ情報を i-FILTER に ICAP で送ります。
4. ユーザ情報を元に URL フィルタリングルールを適用し、接続が許可されていない WEB サイトの場合は、ブロック画面をクライアントに表示します。
5. 許可された通信に関しては、SSL 通信を再暗号化し、WEB サーバへ接続します。



1.1.2 本ガイドの利用バージョンと構成イメージ

本ガイドは、以下の製品バージョンを利用して、作成しております。

1. 利用バージョン

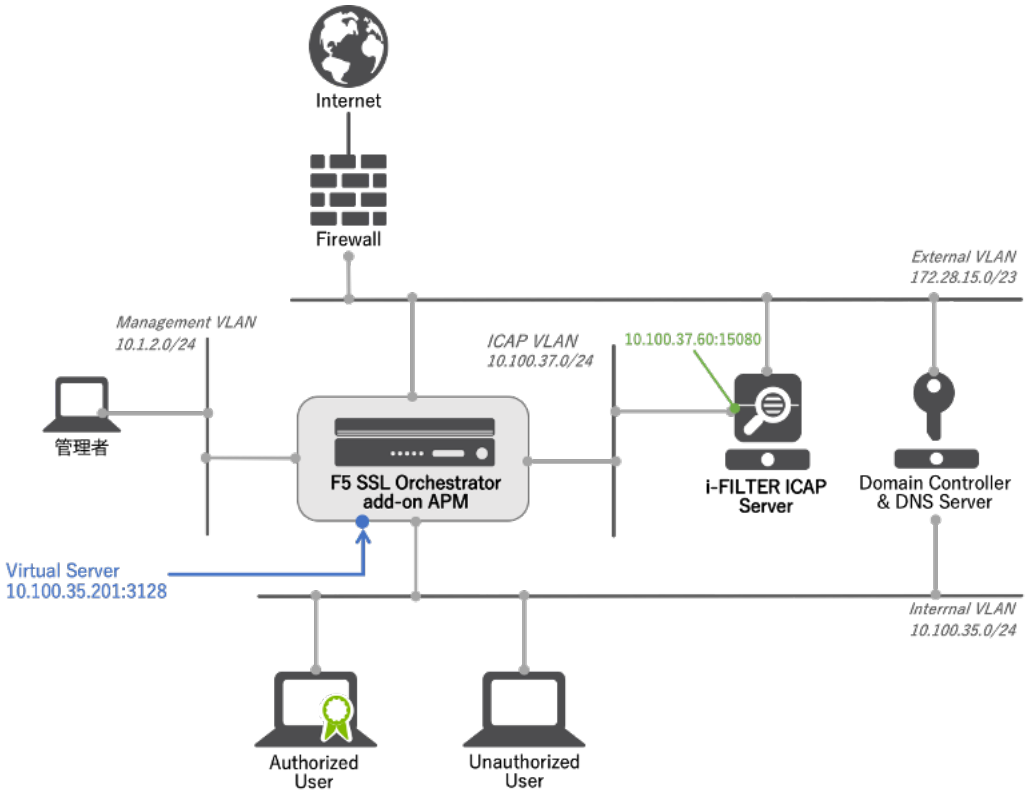
製品名	バージョン
F5 SSL Orchestrator	v15.1.2-0.0.9Final âĂŠ 7.5.2
i-FILTER ICAP Server (Linux)	Ver.10.42R01-20200923 (CentOS v7.7.1908 ※） ※ デジタルアーツ様による製品サポート OS は Red Hat Enterprise Linux となります。

注釈:

- **v15.1.2.1 以上** のバージョンをご利用下さい。(2021 年 3 月追記)
- （各 F5 代理店でサポート可能な範囲において、）極力最新のバージョンを適用頂くことをおすすめ致します。最新のバージョンは AskF5 でご確認下さい。
- Proxy 認証を行いたい場合、もしくは、i-FILTER のログに AD ユーザ名を出力したい場合は、APM のライセンスが必要となります。

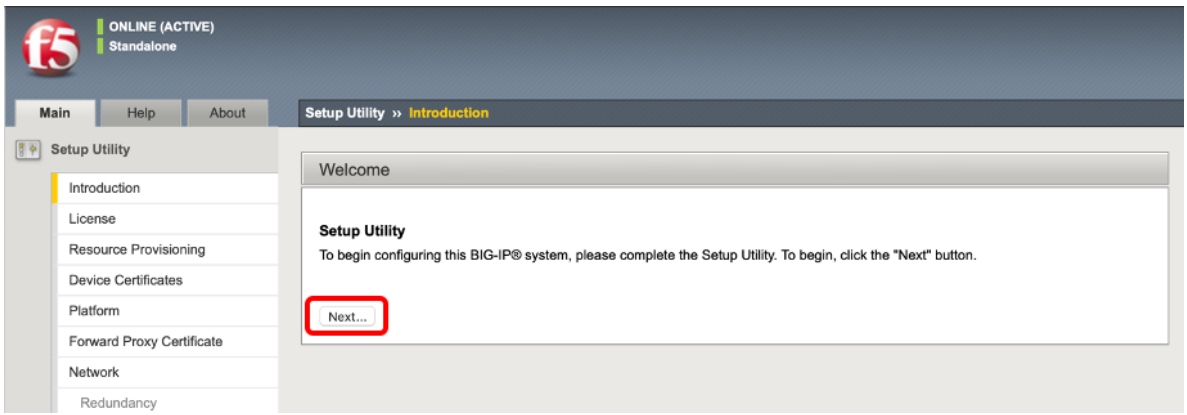
---

2. 本ガイドにおける構成イメージ



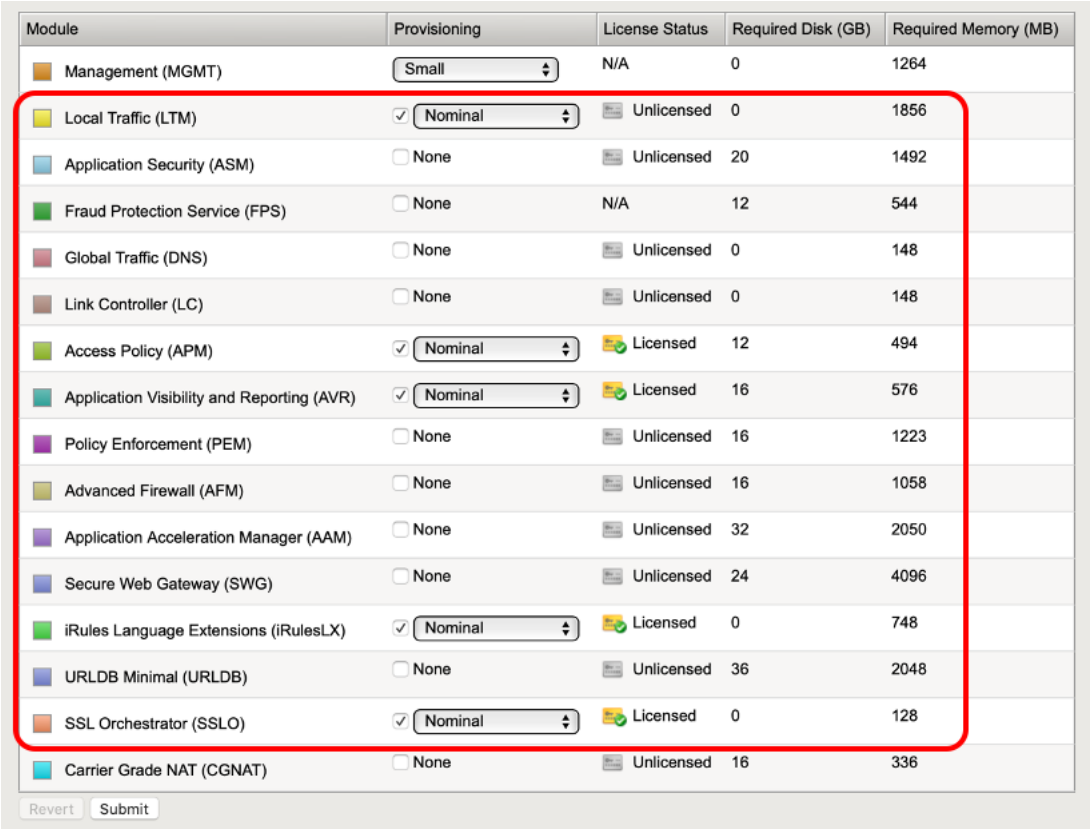
1.1.3 ライセンスアクティベーション、プロビジョニング、CA 証明書/鍵登録

1. *Next* ボタンを押します。

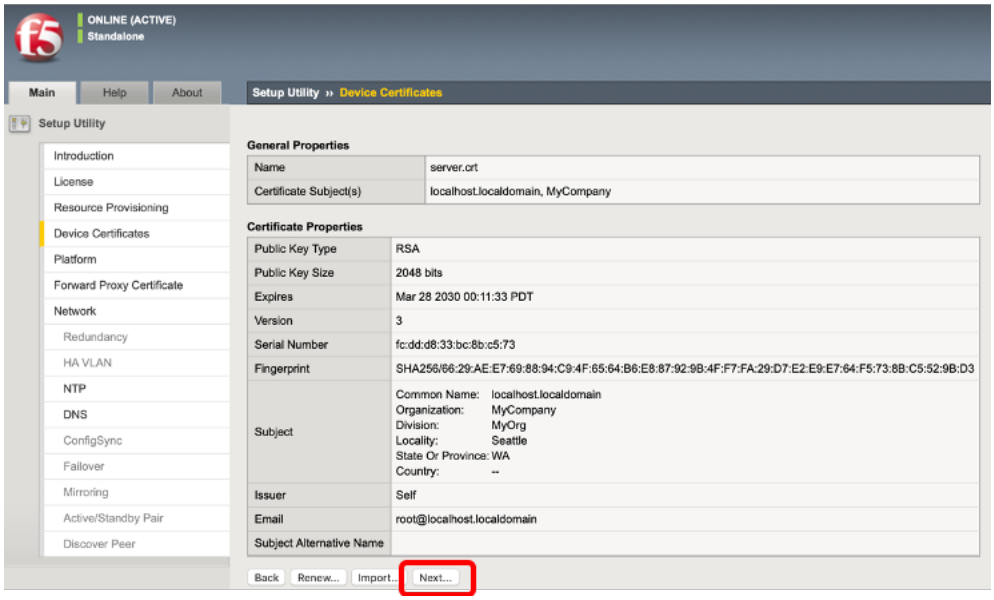


2. ライセンスをアクティベーションします。

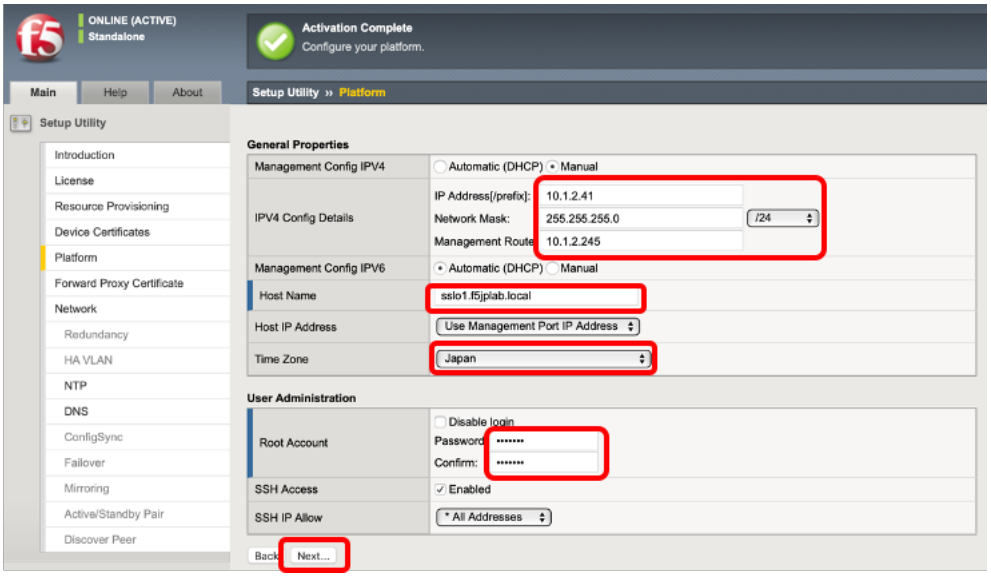
3. 以下のモジュール（(SSL 復号・再暗号化：**LTM,AVR,iRulesLX,SSLO**）＆（認証：**APM**)） をプロビジョニングします。（ライセンスは SSLO と APM の 2 つを利用しています。）



4. *Next* ボタンを押します。

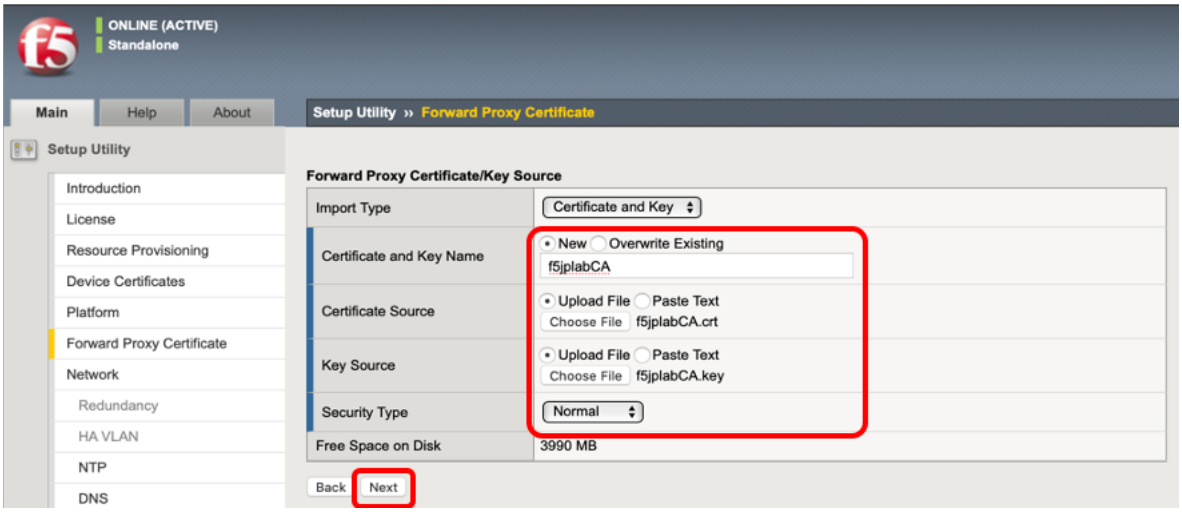


5. **ホスト名**、**タイムゾーン**、**Root パスワード** を設定して、*Next* ボタンを押します。

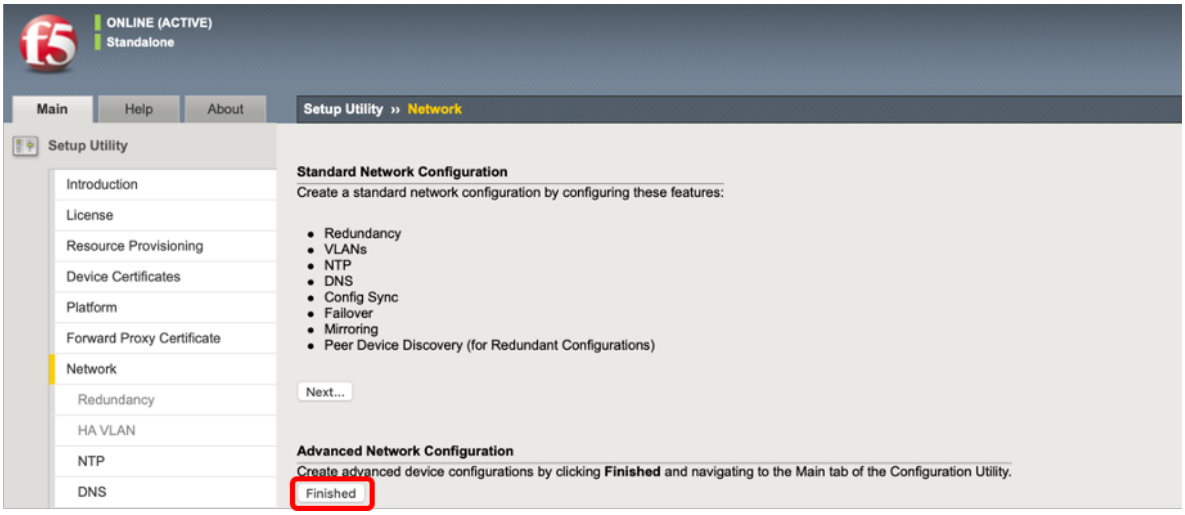


6. SSLO でサーバ証明書を書き換える際に利用する　**CA 証明書**、**CA 鍵** を選択し、**任意の名前** を設定し、

Next ボタンを押します。

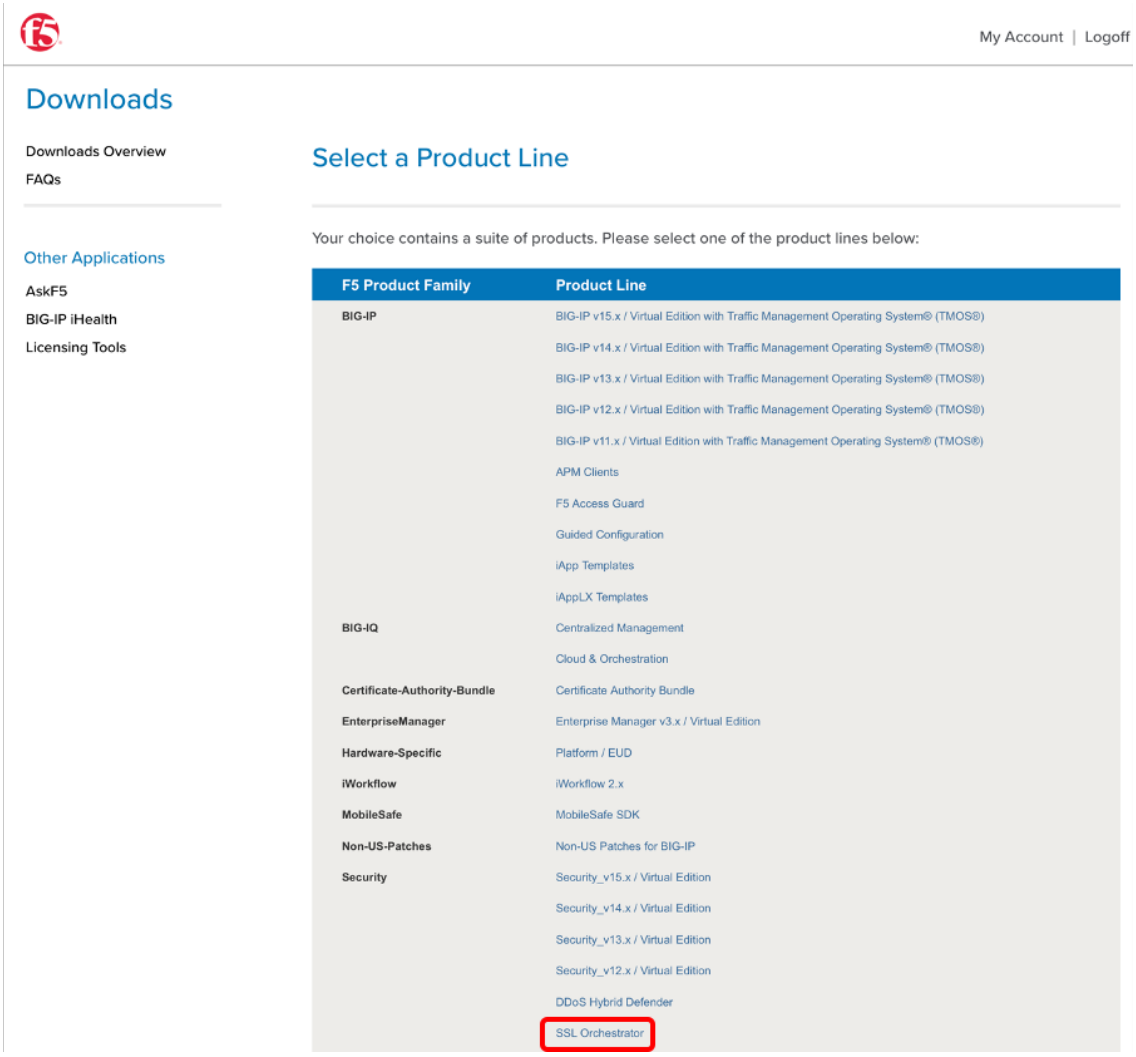


7. *Finished* ボタンを押します。

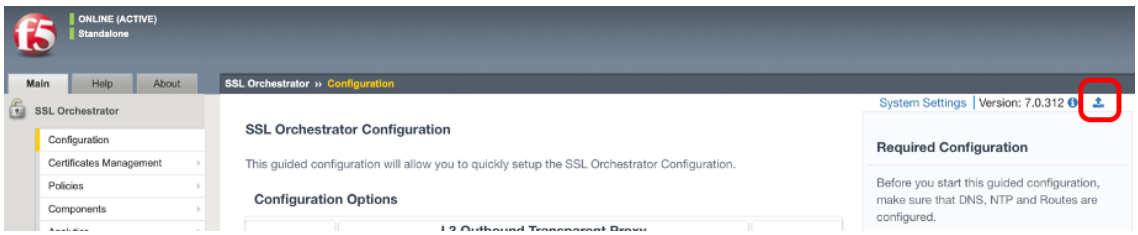


1.1.4 最新版の SSL Orchestrator RPM へのアップグレード

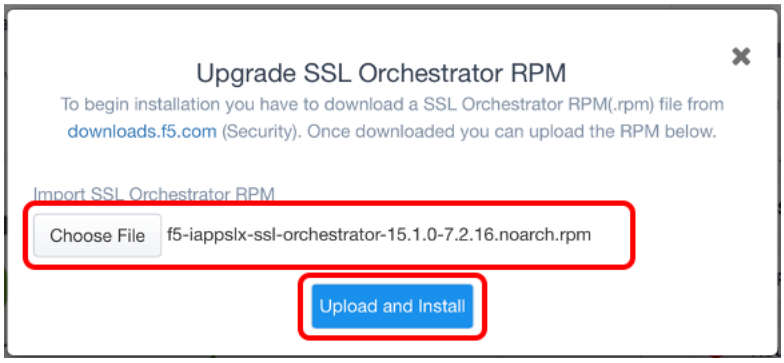
1. AskF5(https://support.f5.com/) より最新版の SSLO RPM をダウンロードします。(SSLO にプリインストールされている RPM のバージョンが最新の場合は、以下の操作は不要です。)



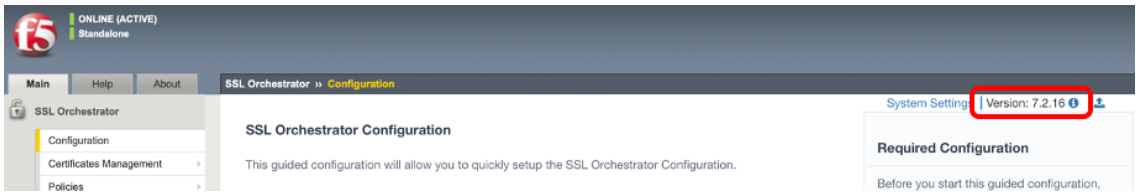
2. **SSL Orchestrator >> Configuration** の画面にて、右上の **アップグレードボタン** を押します。



3. *Choose File* にて、先程ダウンロードした RPM を選択し、*Upload and Install* を押します。

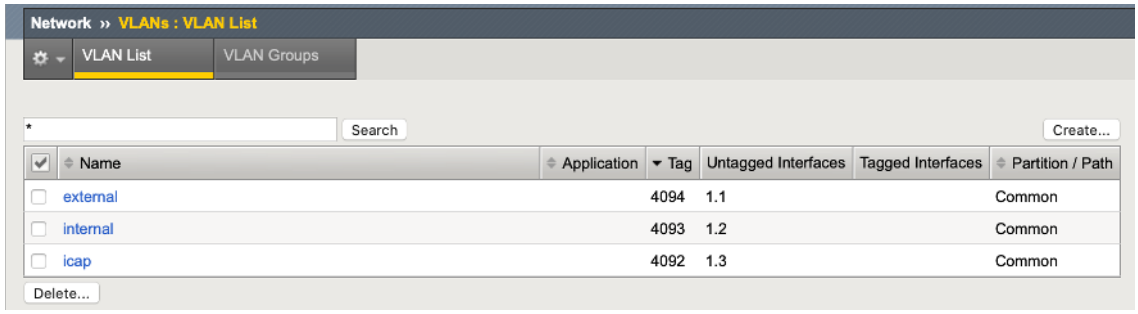


4. バージョンがアップグレードされていることを確認します。

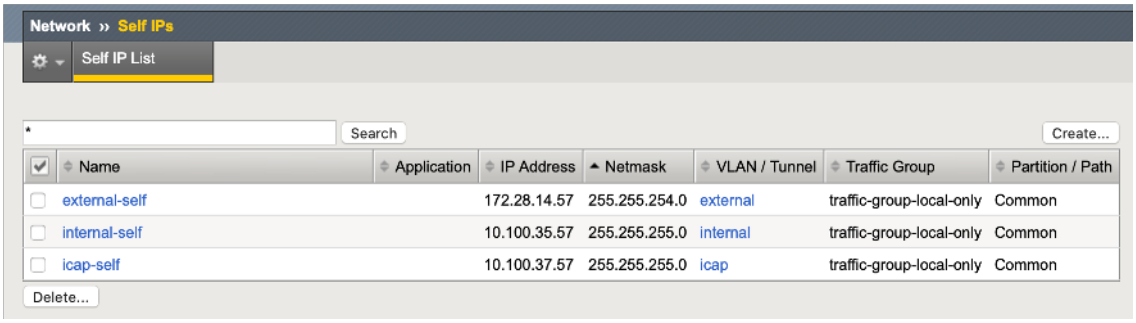


1.1.5 Network の基本設定

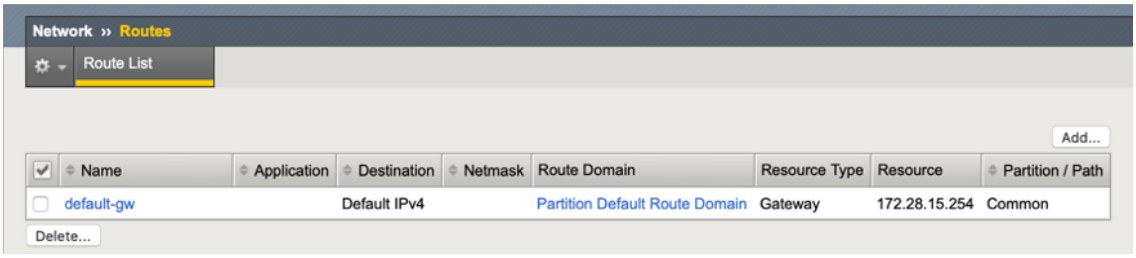
1. VLAN の設定を行います。



2. Self IP の設定を行います。

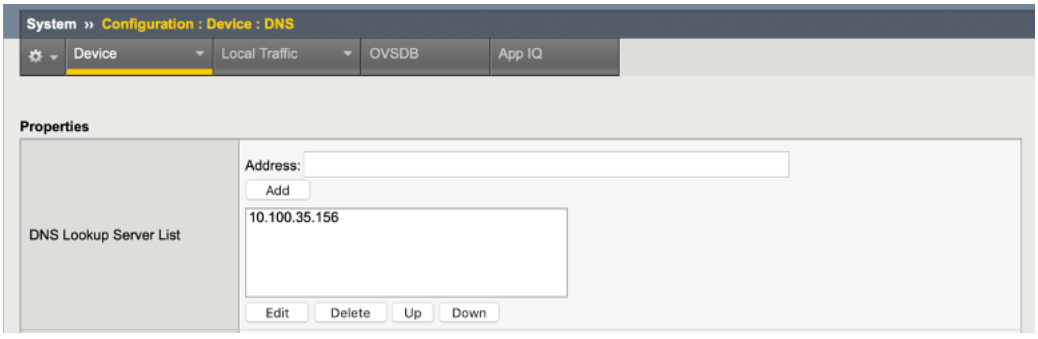


3. デフォルトゲートウェイの設定を行います。

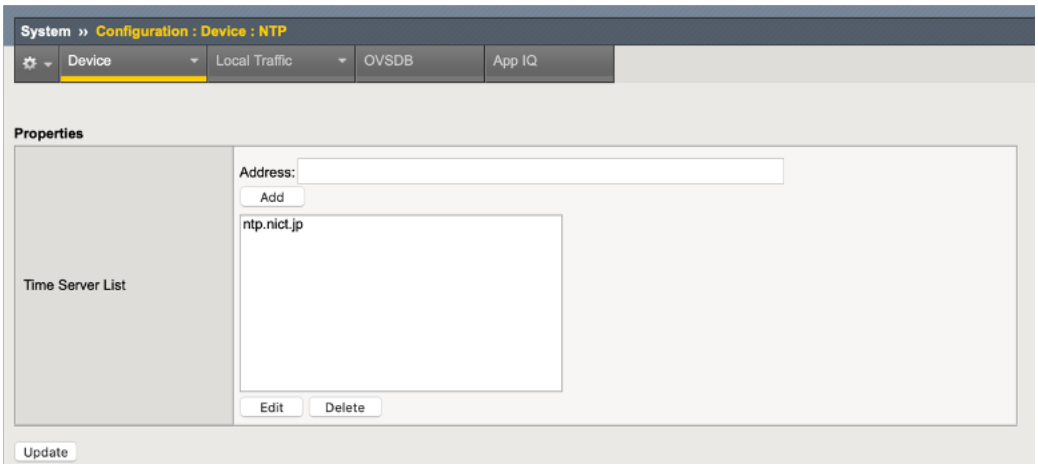


1.1.6 DNS, NTP の設定

1. DNS の設定を行います。



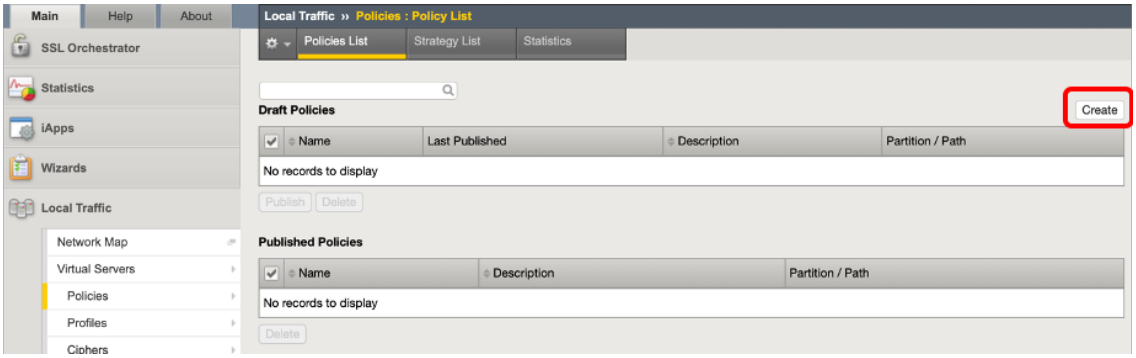
2. NTP の設定を行います。



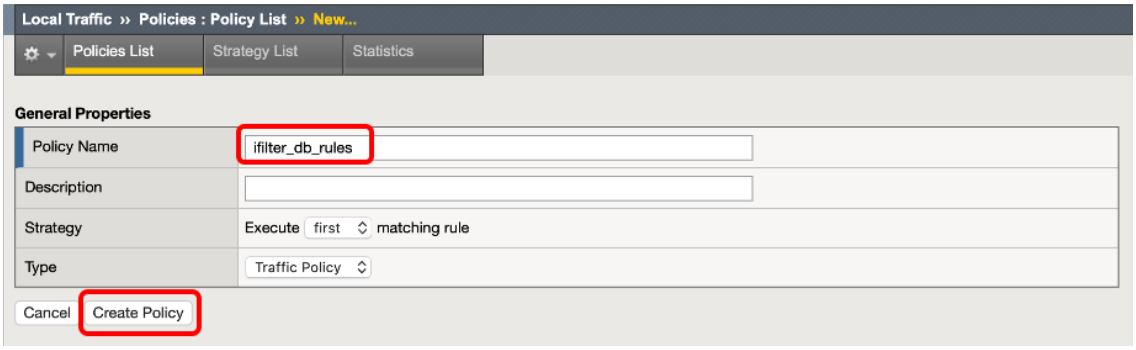
1.1.7 i-FILTER にて HTTP/HTTPS 判別するための設定（Local Traffic Policy の設定）

i-FILTER ICAP 版は、ICAP のリクエストヘッダの一部で HTTP サーバへの通信か HTTPS サーバへの通信かを判別しています。i-FILTER が HTTP/HTTPS 判別可能となるように Local Traffic Policy にてルールを作成します。同時に i-FILTE は ICAP レスポンスはチェックしないので、ICAP レスポンスチェックを無効にします。また、i-FILTER による URL Filtering チェック後に、リクエストヘッダを元の値に戻すための Local Traffic Policy ルールも作成します。こちらのルールでは実際に ICAP 通信は行わないので、ICAP 通信を無効にします。

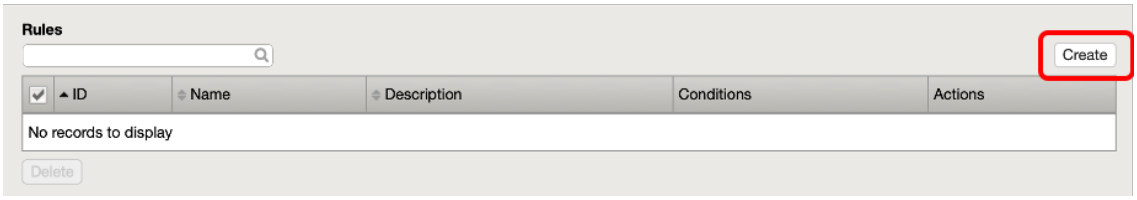
1. **Local Traffic >> Policies >> Policies List** にて、*Create* ボタンを押します。



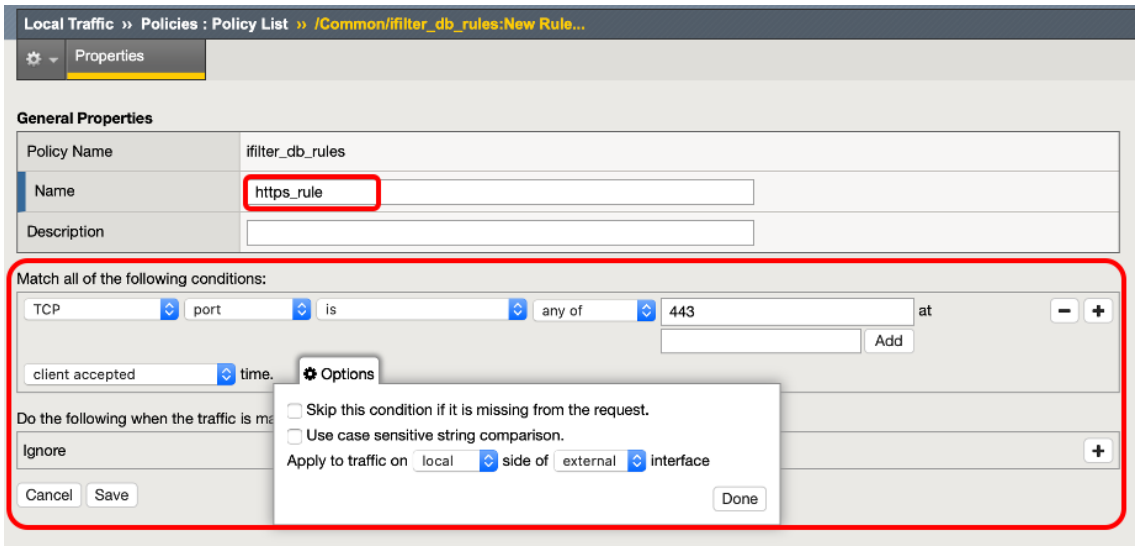
2. **任意のポリシー名** を入力し、*Create Policy* ボタンを押します。



3. HTTPS 用のルールを作成します。Rules の *Create* ボタンを押します。

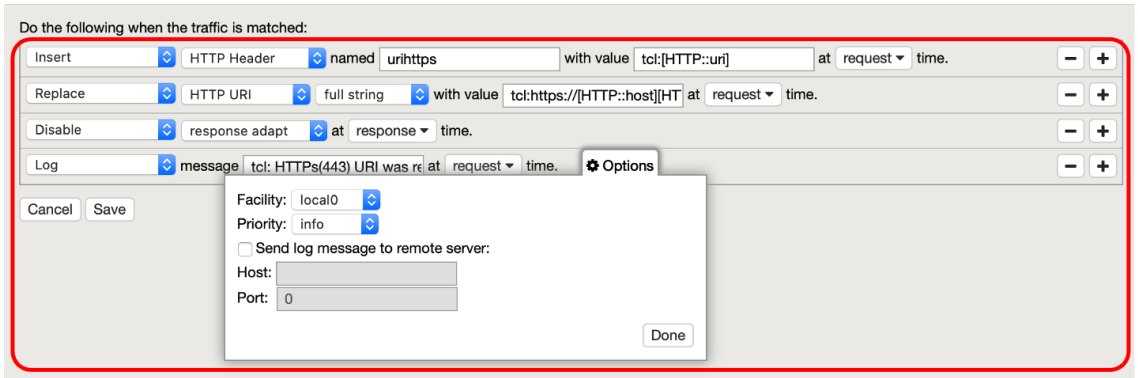


4. **任意の Rule 名** を入力し、**Match all of the following conditions:** の ＋ マークをクリックし、以下のように入力します。



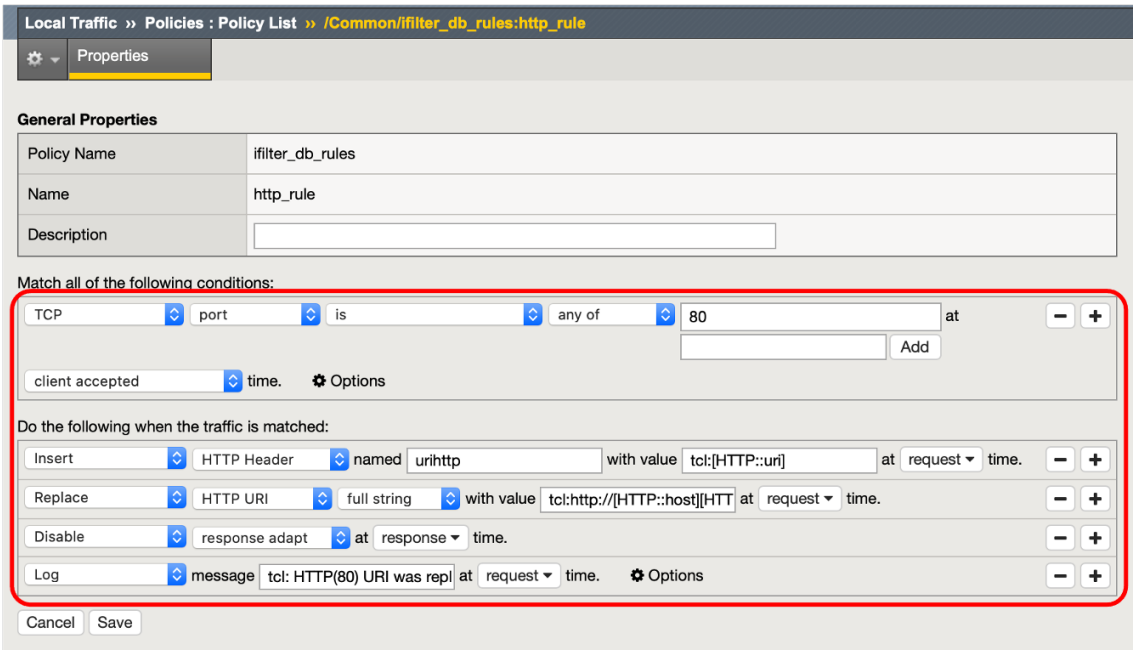
Match all of the following conditions:	必要有無
TCP port is any of 443 at client accepted time.	必須
Apply to traffic on local side of external interface	必須

5. 同様に、**Do the following when the traffic is matched:** の ＋ マークをクリックし、以下のように入力し、*Save* ボタンを押します。（デバック用のログルールは任意で追加します。）



Do the following when the traffic is matched:	必要有無
Insert HTTP Header named urihttps with value tcl:[HTTP::uri] at request time.	必須
Replace HTTP URI full string with value tcl:https://[HTTP::host][HTTP::uri] at request time.	必須
Disable response adapt at response time.	必須
Log message tcl: HTTPs(443) URI was replaced to: [HTTP::uri] at request time.	任意
Facility: local0 Priority: info	任意

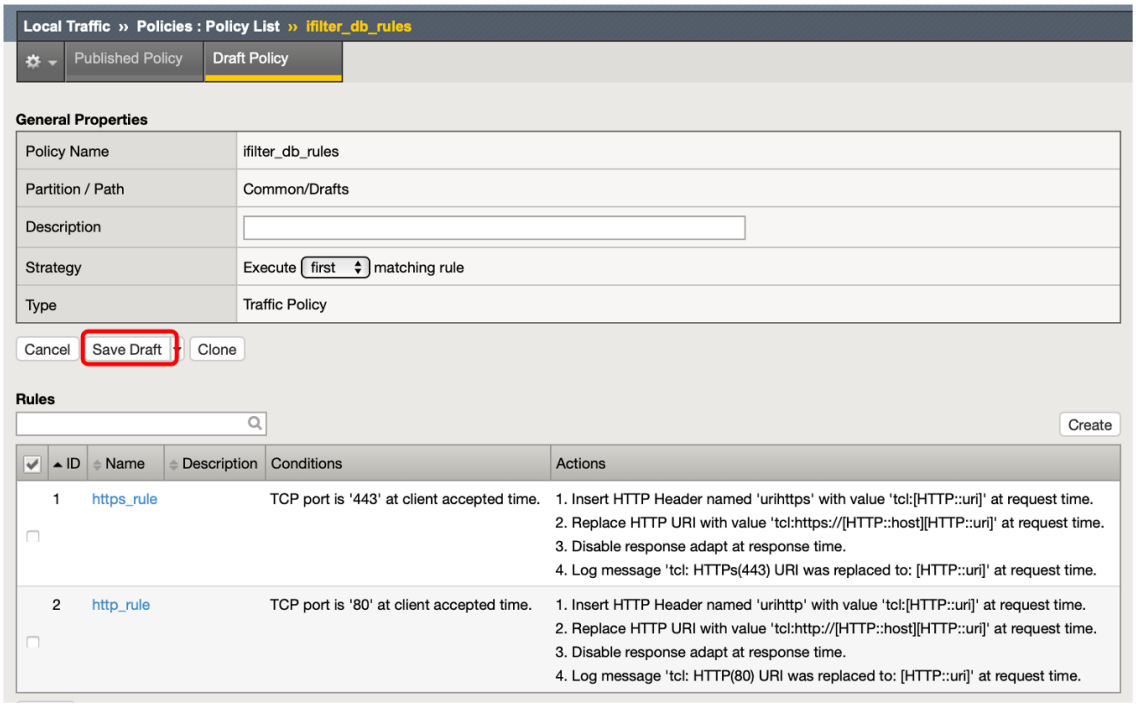
6. 同様に HTTP 用のルールを作成します。



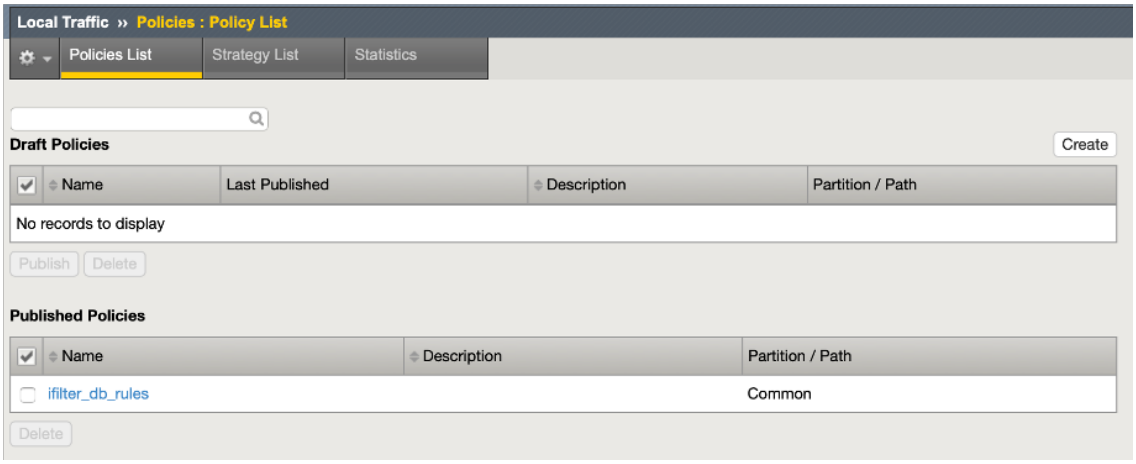
Match all of the following conditions:	必要有無
TCP port is any of 80 at client accepted time.	必須
Apply to traffic on local side of external interface	必須

Do the following when the traffic is matched:	必要有無
Insert HTTP Header named urihttp with value tcl:[HTTP::uri]** at **request time.	必須
Replace HTTP URI full string with value tcl:http://[HTTP::host][HTTP::uri] at request time.	必須
Disable response adapt at response time.	必須
Log message tcl: HTTP(80) URI was replaced to: [HTTP::uri] at request time.	任意
Facility: local0 Priority: info	任意

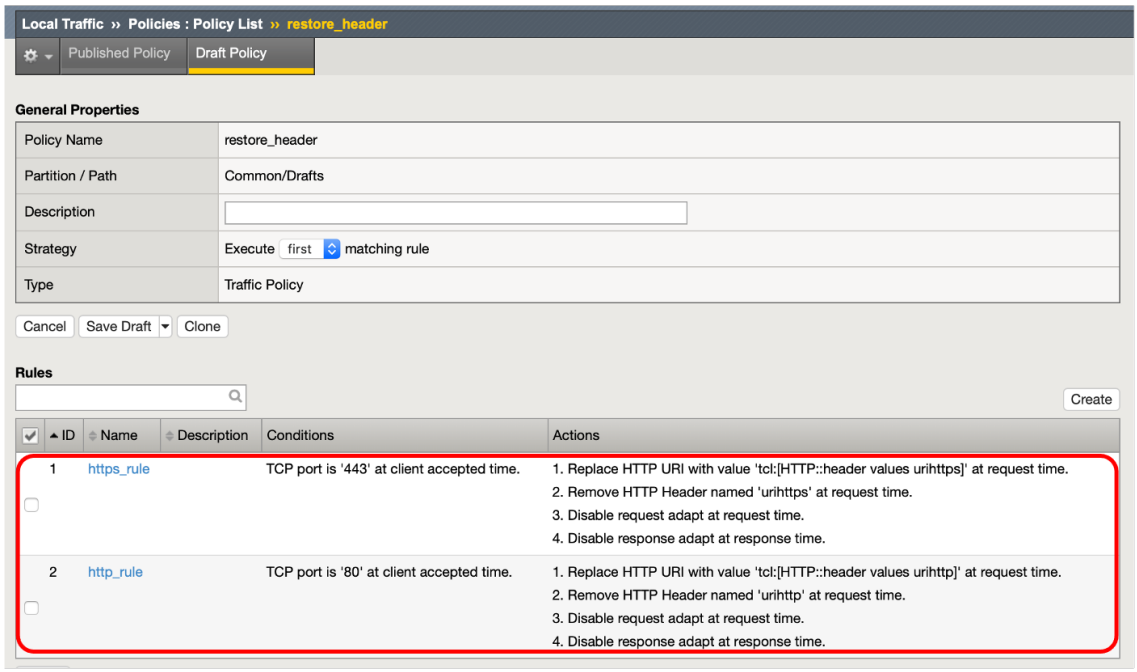
7. 2つのルール作成後は、以下のようになります。*Save Draft* ボタンを押します。



8. *Publish* ボタンを押すと、以下のようになります。



9. 上記手順と同様に、以下のようなリクエストヘッダをもとに戻す Local Traffic Policy ルールを作成します。



Match all of the following conditions:	必要有無
TCP port is any of 443 at client accepted time.	必須
Apply to traffic on local side of external interface	必須

Do the following when the traffic is matched:	必要有無
Replace HTTP URI full stringÂăwith valueÂătcl:[HTTP::header values urihttps]ÂăatÂărequestÂătime.	必須
Remove HTTP Header named urihttps at request time.	必須
Disable request adapt at request time.	必須
Disable response adapt at response time.	必須

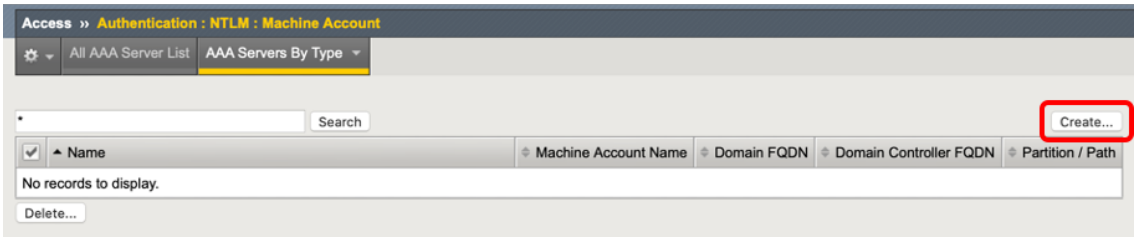
Match all of the following conditions:	必要有無
TCP port is any of 80 at client accepted time.	必須
Apply to traffic on local side of external interface	必須

Do the following when the traffic is matched:	**必要有無**
`Replace HTTP URI full string`Âăwith valueÂă`tcl:[HTTP::header values urihttp]`ÂăatÂă`request`Âătime.	必須
`Remove HTTP Header` named `urihttp` at `request` time.	必須
`Disable request adapt` at `request` time.	必須
`Disable response adapt` at `response` time.	必須

1.1.8 NTLM 認証設定

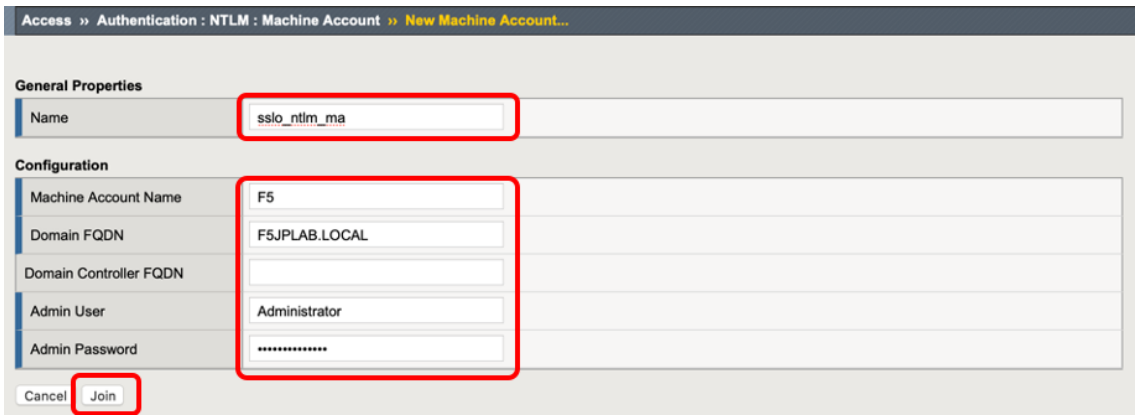
APM で行う認証の設定を行います。ここでは、NTLM 認証の設定を行います。AD サーバ（ドメインコントローラ、DNS）の構築済みであることを前提としております。

1. **Access >> Authentication >> NTLM >> Machine Account** を選択し、*Create* ボタンを押します。

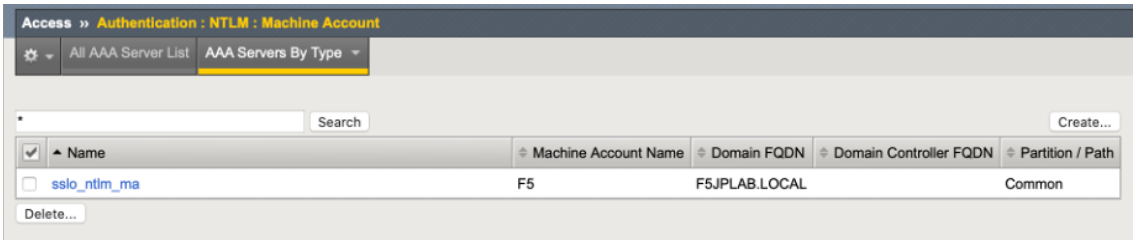


2. **任意の名前** を設定し、**Machine Account Name** には **任意の SSLO 識別名** を入力し、**Domain FQDN**、

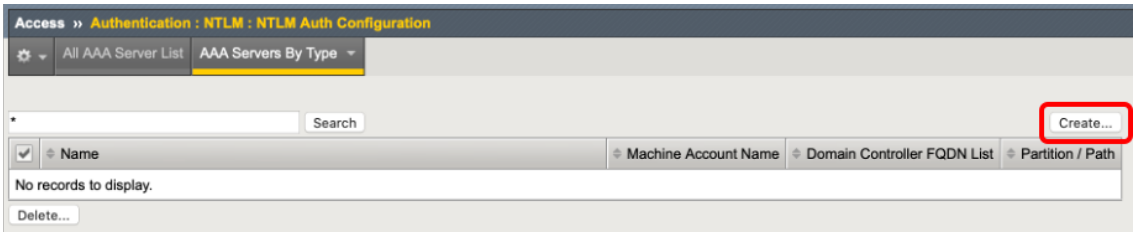
Admin ユーザ名 と **パスワード** を入力し、*Join* を押します。



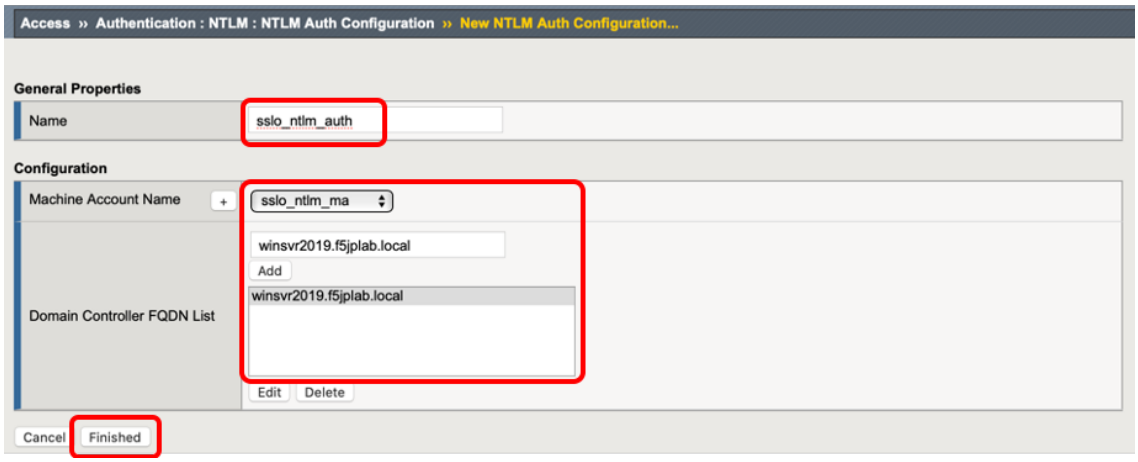
3. 以下のようになります。



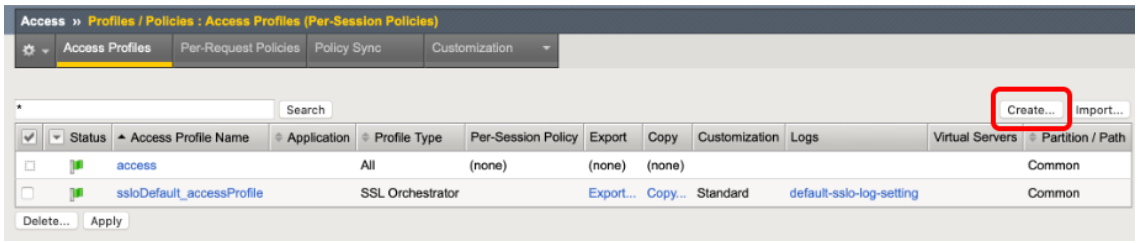
4. **Access >> Authentication >> NTLM >> NTLM Auth Configuration** を選択し、*Create* ボタンを押します。



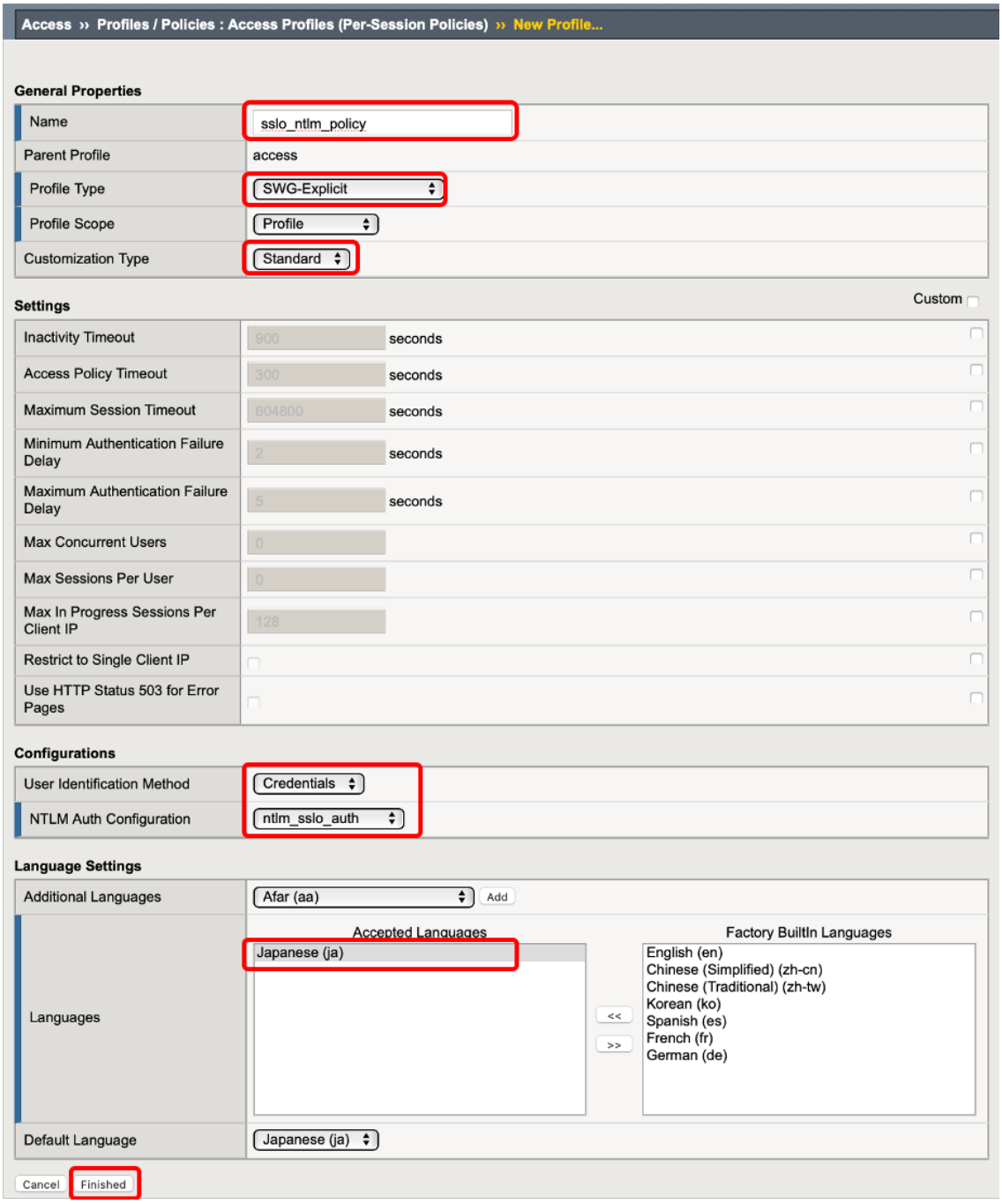
5. **任意の名前** を設定し、**Machine Account Name** は **先程作成したもの** を選択します。**Domain Controller FQDN List** には **ご利用のドメイン名（FQDN 名）** を指定し、*Finished* ボタンを押します。



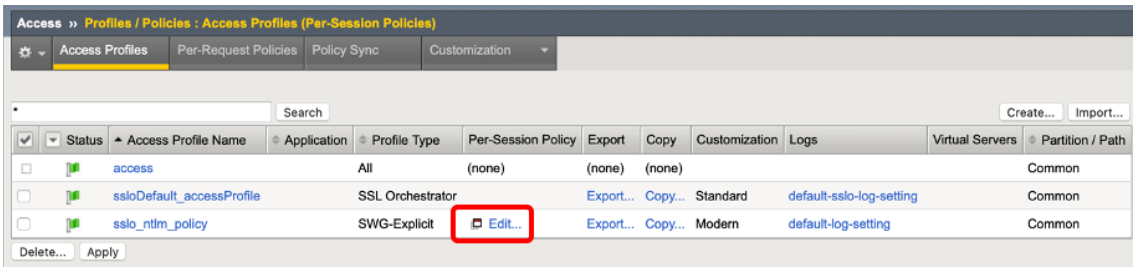
6. **Access >> Profiles/Policies >> Access Profiles(Per-Session Policies)** を選択し、*Create* ボタンを押します。



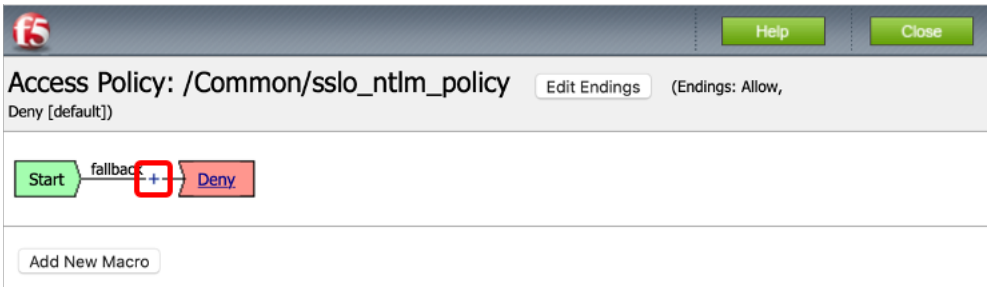
7. **任意の名前** を設定し、**Policy Type** に **SWG-Explicit** を選択し、**Customization Type** に **Standard** を選択し、**User Identification Method** にて、**Credential** を選択し、**NTLM Auth Configuration** に **先程作成したもの** を選択し、**Languages** は **ご利用の言語** を選択して、*Finished* ボタンを押します。



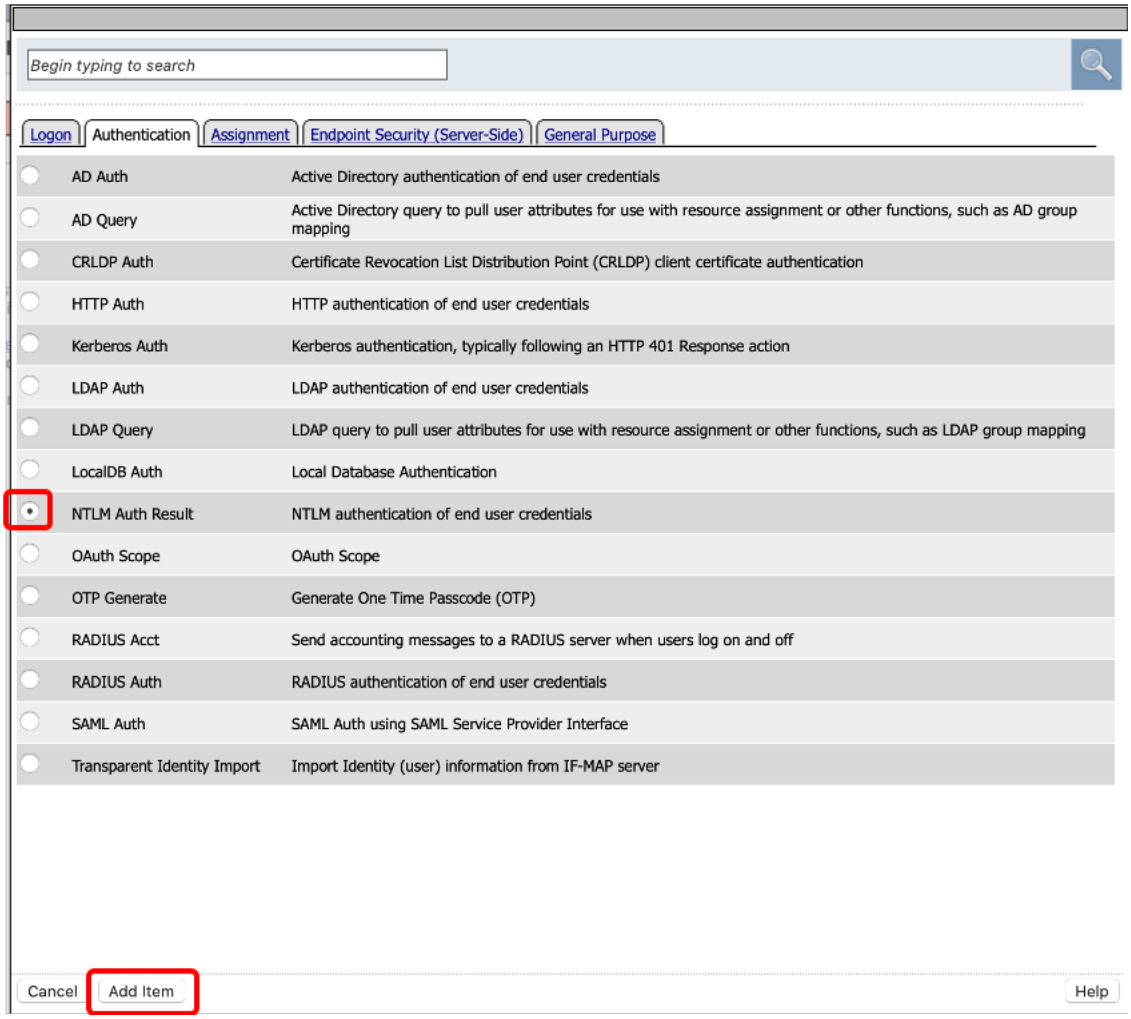
8. 作成された Access Profile の一覧から先程作成したプロファイルを見つけ、Per-Session Policy 欄の *Edit* を押します。



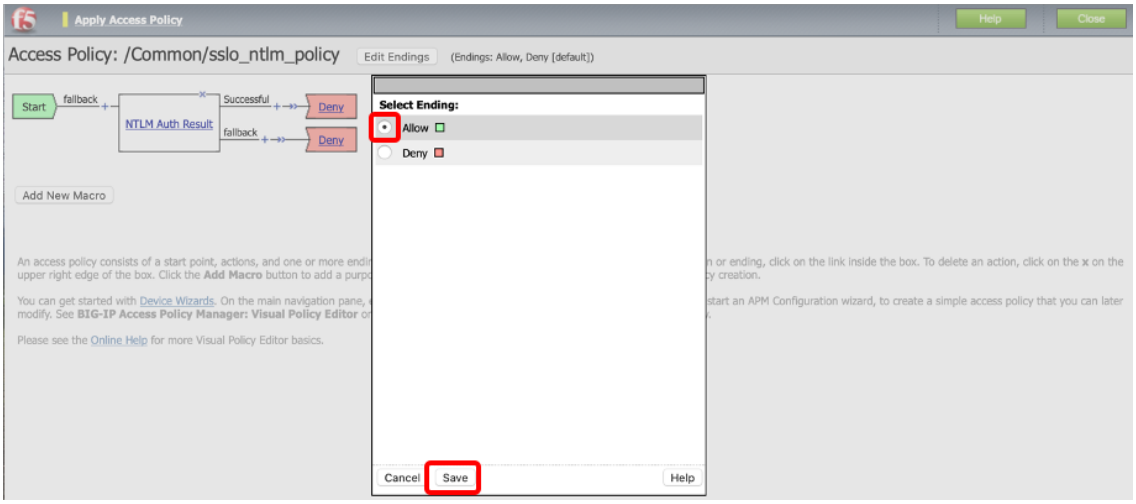
9. ブラウザの別タブに VPE が表示されます。Start の右隣の ＋ ボタンを押します。



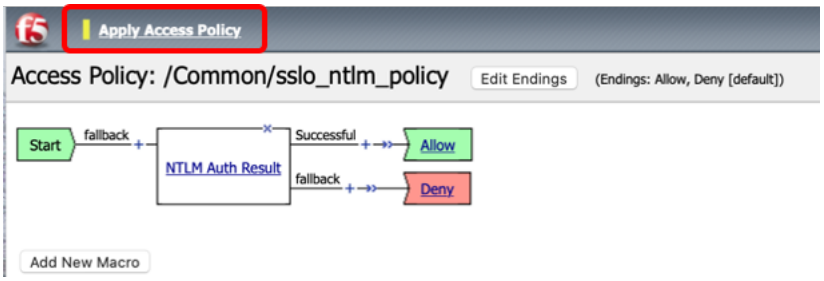
10. Authentication タブの **NTLM Auth Result** を選択し、*Add Item*、*Save* を押します。



11. NTLM Auth Result の右の Successful につながるフローを **Deny** から **Allow** に変更します。



12. 左上の *Apply Access Policy* を押し、ブラウザの VPE タブを閉じます。



1.1.9 認証ヘッダ挿入用 iRule の作成

ここでは、ユーザ認証ヘッダ（X-Authenticated-User）と APM で認証されたユーザ名（session.logon.last.username）を ICAP リクエストに挿入するための iRule を作成します。ICAP は HTTP をカプセリングしますので、認証情報を挿入するためには以下の 2 つの iRule が必要となります。また、i-FILTER ICAP 版に認証ユーザ名を理解してもらうために、ユーザ名の手前に **ldap:///** を加え、**Base64 エンコーディング** をする必要があります。

1. **Local Traffic >> iRules** にて、*Create* ボタンを押します。**任意の名前** を入力して、**Definition** に以下サンプル iRule を入力し、*Finished* ボタンを押します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。）

例１）HTTP リクエストヘッダに認証情報を挿入するための iRule サンプル

```
###  Add this iRule to Virtual Server:ssloS_XXXX-t-4 ###

when HTTP_REQUEST {

    # Create the HDR list and insert the X-Server-IP value
    set hdr [list "X-Server-IP:[IP::local_addr]"]

    # Add the authenticated user information if it exists
    if { [ACCESS::session data get session.logon.last.username] ne "" } {
        # In case of  NLTM/Basic Auth
        lappend hdr "X-Authenticated-User:[b64encode ldap:///[ACCESS::session␣
↪data get session.logon.last.username]]"
    }

    # Add above info to the table temporarily and add an ICAP-X header
    set guid [expr {int(rand()*1e12)}]
    table set ${guid} ${hdr} 10 10
    HTTP::header replace ICAP-X ${guid}

}
```

例２）ICAP リクエストヘッダに認証情報を挿入するための iRule サンプル

```
### Add this iRule to ssloS-XXXX-req ###

when ICAP_REQUEST {

    if { [HTTP::header exists "ICAP-X"] } {

        # Get the header and the value from table
        set hdrs [table lookup [HTTP::header "ICAP-X"]]

        # Add it to ICAP header
        foreach x ${hdrs} {
            ICAP::header add [lindex [split ${x} ":"] 0] [lindex [split ${x}
↪":"] 1]
        }

    # Remove the temporary http header. But the data would show up in the␣
↪ICAP header.
    HTTP::header remove "ICAP-X"
    }
}
```

２つ目の iRule において、ICAP-X ヘッダは HTTP ヘッダ上からは削除されますが、ICAP ヘッダに残

ります。このヘッダは無害ですが、気になる場合は、i-FILTER のヘッダコントロールにて削除して下さい。
また、接続テストをされる際には、変数情報をログ出力するなどして意図する値が入っているか確かめて頂くことをおすすめします。

例１）認証ユーザ名を確認するデバッグ

```
log local0. "Username: [ACCESS::session data get session.logon.last.username]"
```

例２）エンコードされた内容を確認するデバッグ

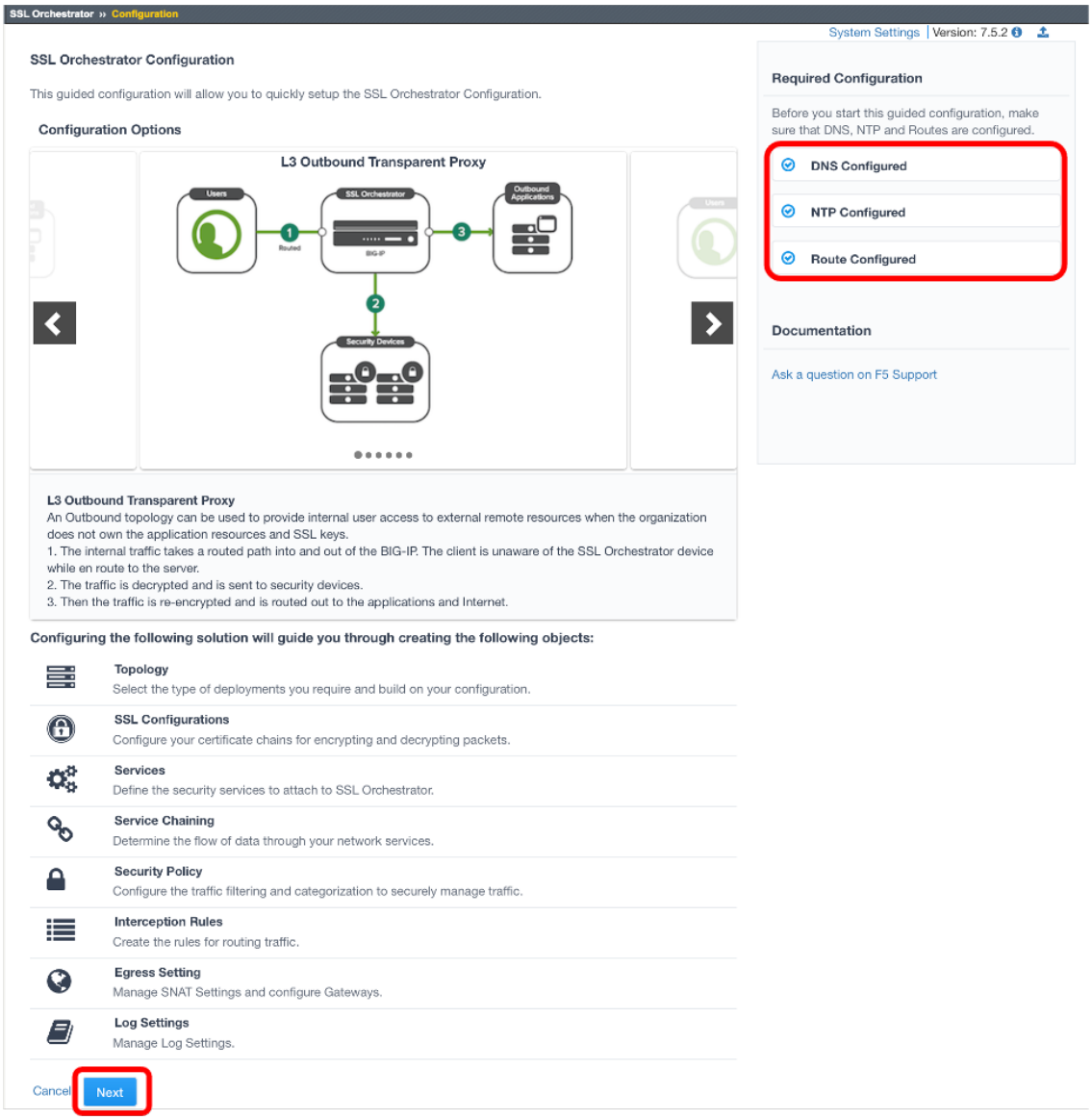
```
log local0. "Encoded username: b64encode ldap:///{[}ACCESS::session data get
session.logon.last.username]]"
```

例３）ヘッダー内容を確認するデバッグ

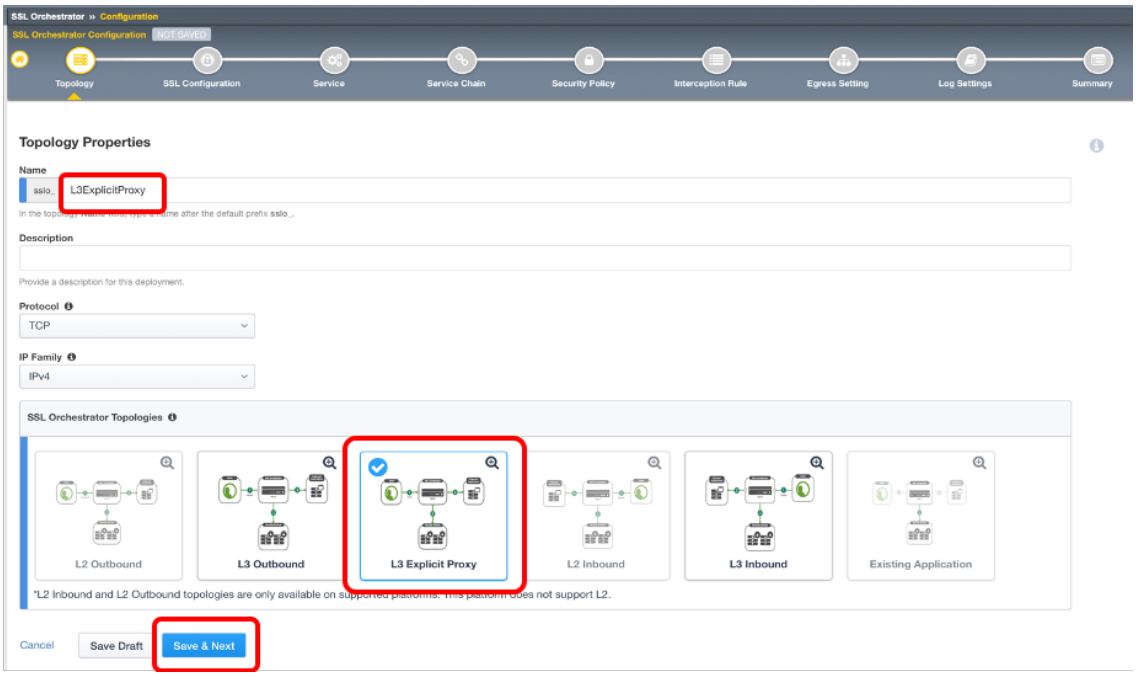
```
foreach attr "[HTTP::header names]" {
        log local0. "$attr : [HTTP::header value $attr]"
}
```

1.1.10 SSLO Guided Configuration による SSLO の設定

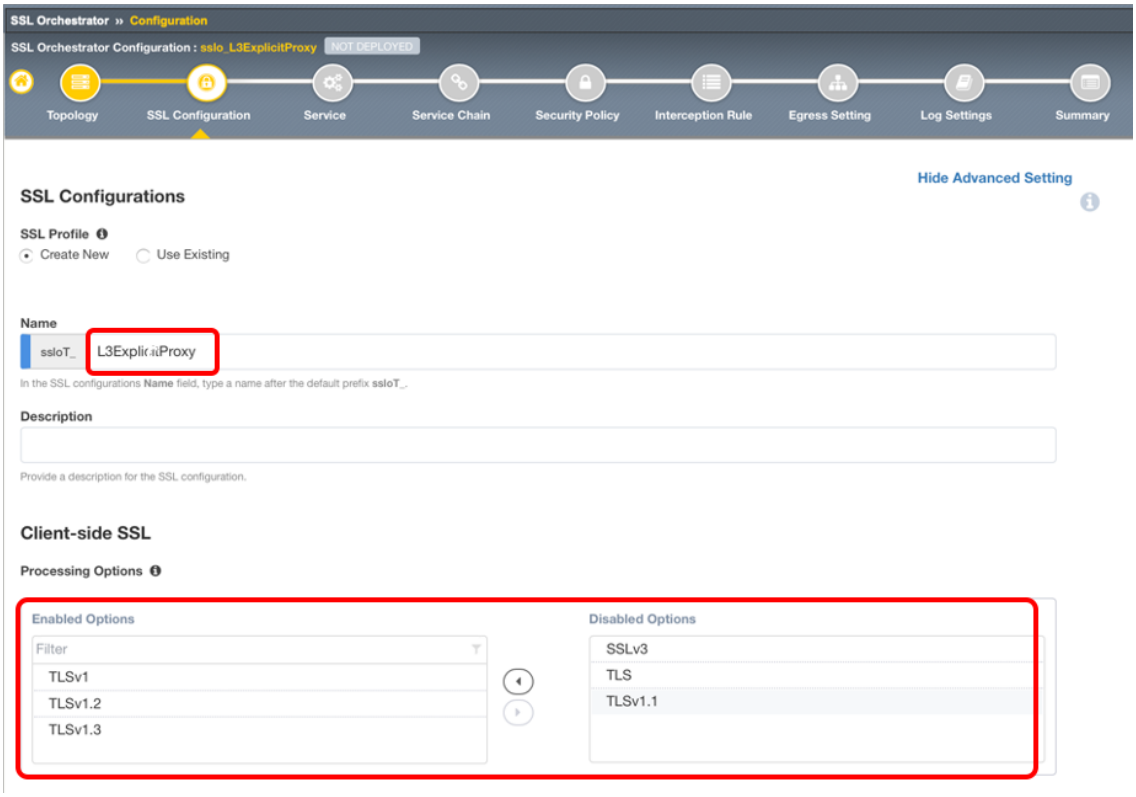
1. **SSL Orchestrator >> Configuration** を選択します。**DNS** と **NTP** と **Route** が **Configure** となっているのを確認し、*Next* ボタンを押します。



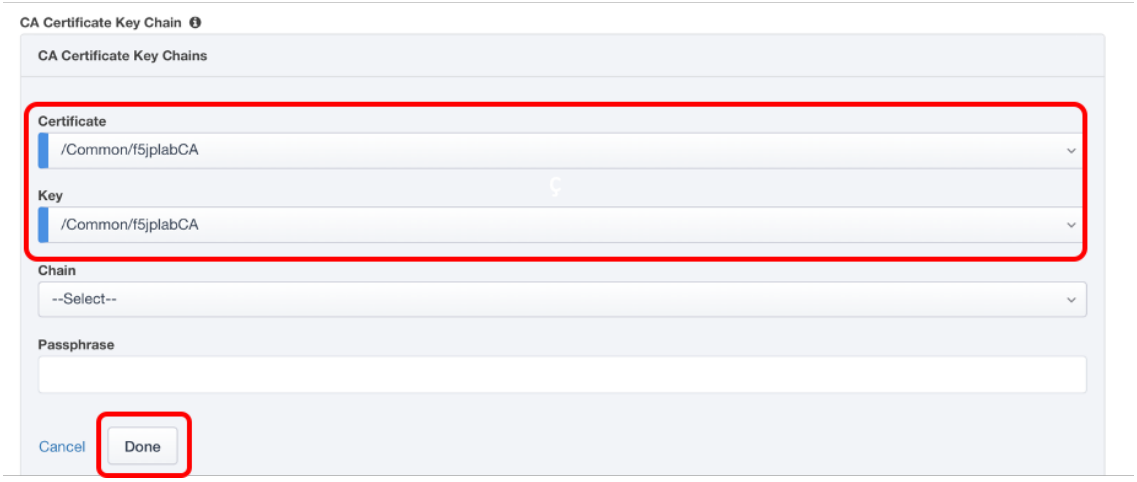
2. **任意の名前** を設定し、SSL Orchestrator Topologies として、**L3 Explicit Proxy** を選択し、*Save* **&** *Next* ボタンを押します。



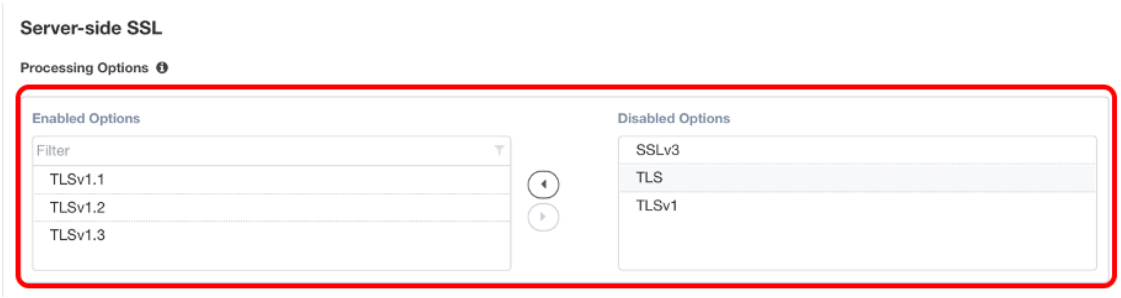
3. **任意の名前** を設定し、右上の *Show Advanced Setting* をクリックし、**Client-side SSL** にて、利用したい **TLS のバージョン** を選択します。



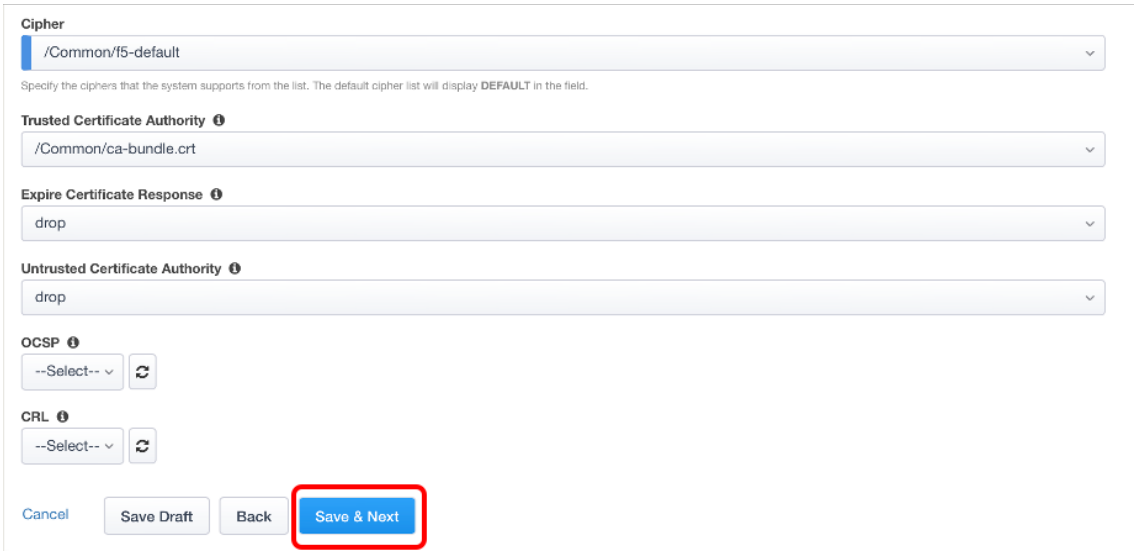
4. **CA Certificate KeyChain** にて、既にインポート済みの CA ファイルを選択して *Done* を押します。



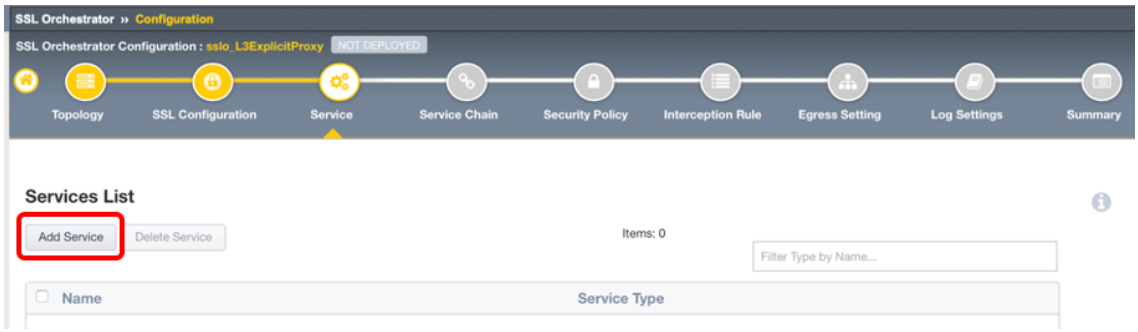
5. **Server-side SSL** も同様に利用したい **TLS バージョン** を選択します。



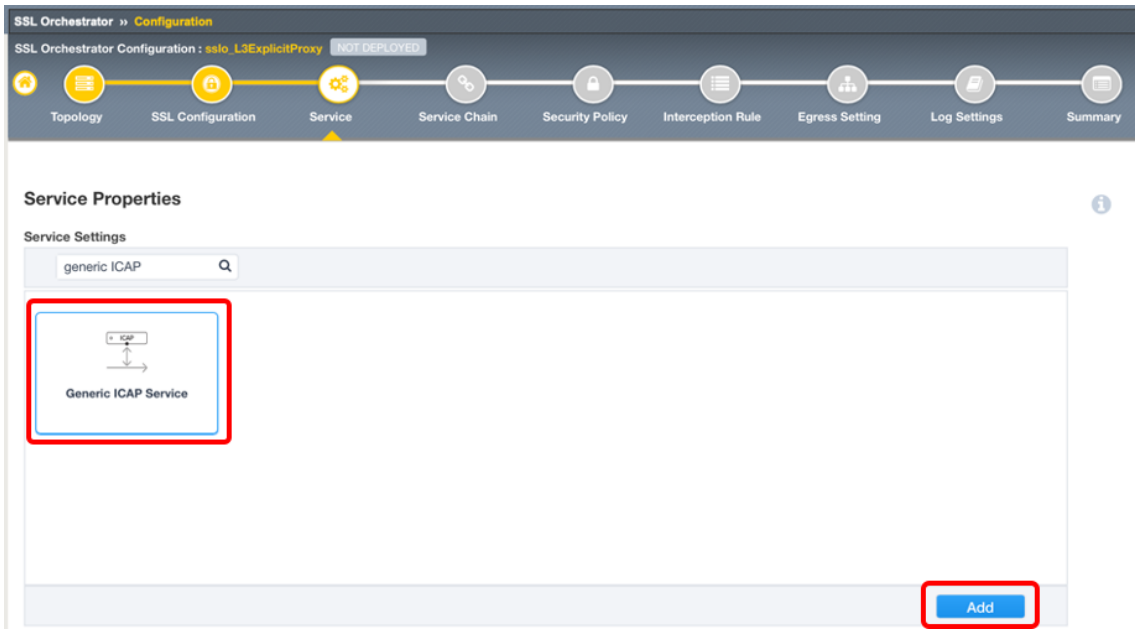
6. 期限切れの証明書や自己署名証明書に対しての動作も確認し、*Save* **&** *Next* を押します。



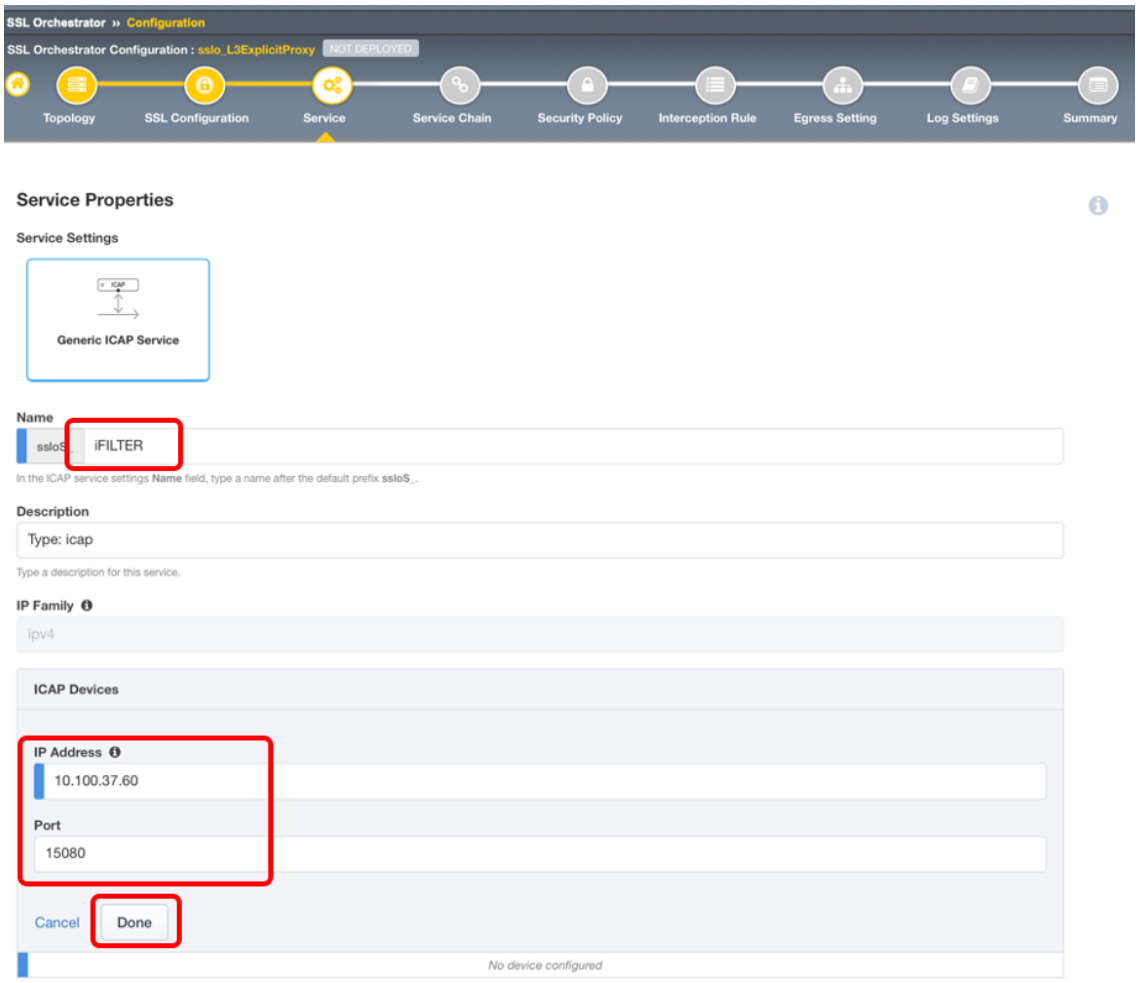
7. *Add Service* を押します。



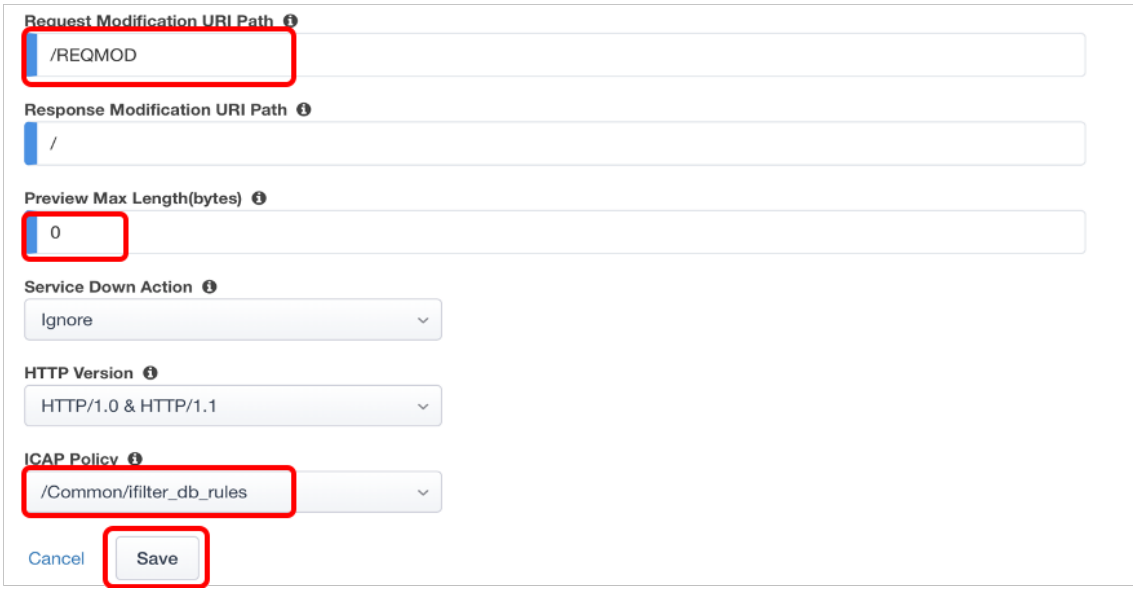
8. **Generic ICAP Service** を選択し、*Add* ボタンを押します。



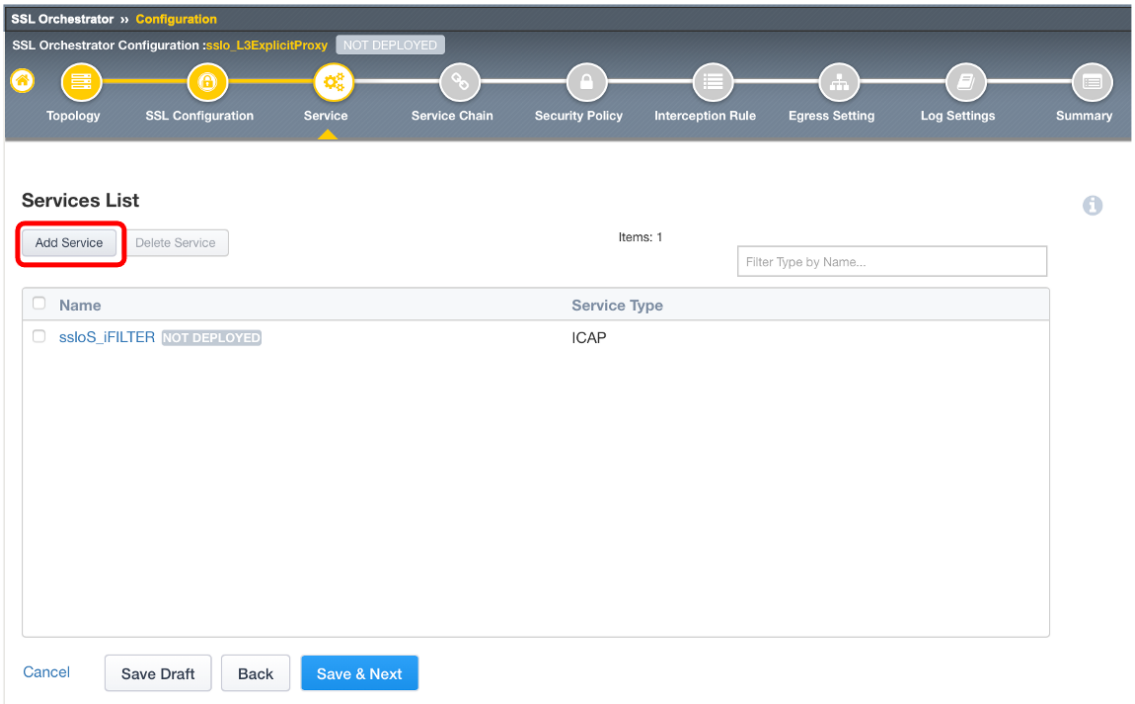
9. **任意の名前** を設定し、**ICAP Devices** に **i-FILTER の IP アドレス** と **ポート番号** を設定し、*Done* を押します。



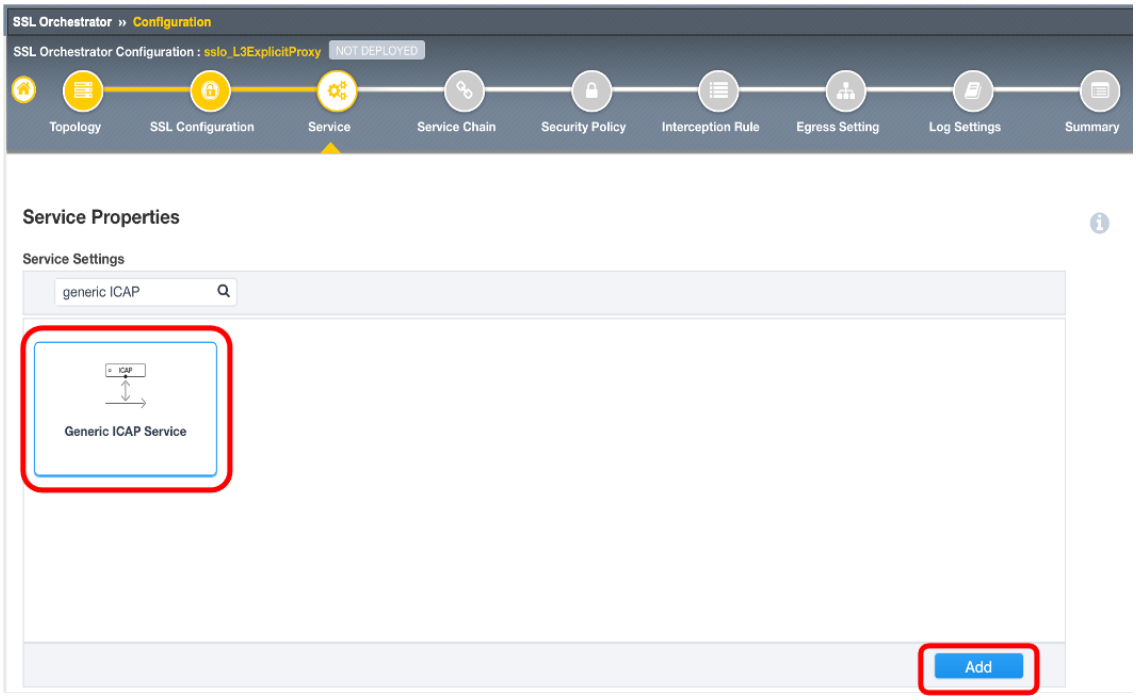
10. **Request Modification URI Path** に **/REQMOD** を入力し、**Preview Max Length** に **0** を入力し、**ICAP Policy** に既に作成済みの **Local Traffic Policy** を選択し、*Save* を押します。(i-FILTER ICAP 版は **ICAP Preview** に対応していないので、**0** を入力します。)



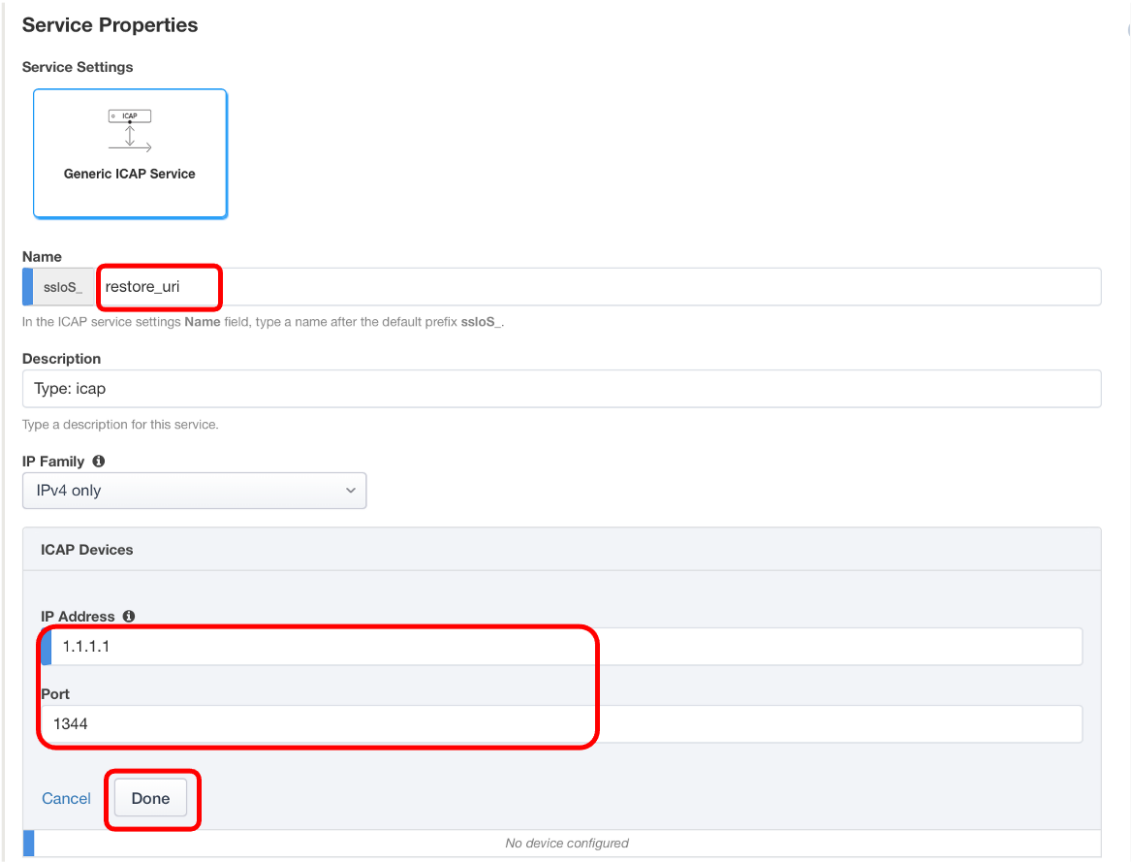
11. *Add Service* を選択します。



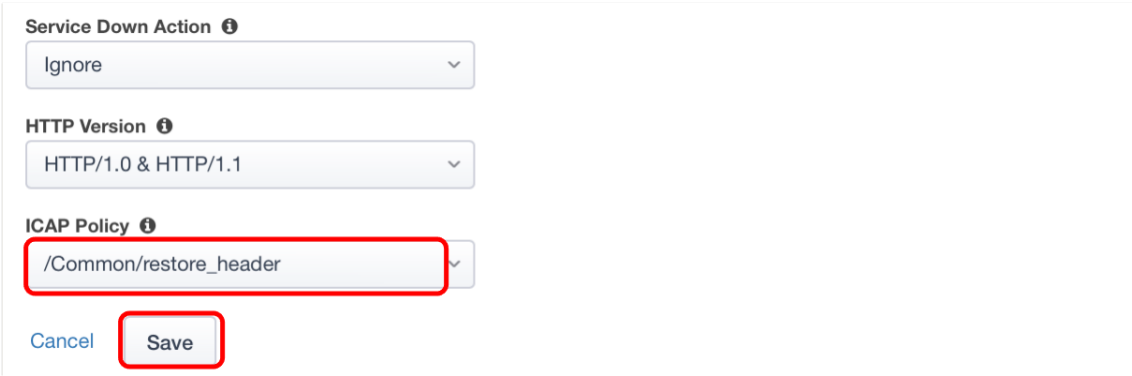
12. **Generic ICAP Service** を選択し、*Add* ボタンを押します。



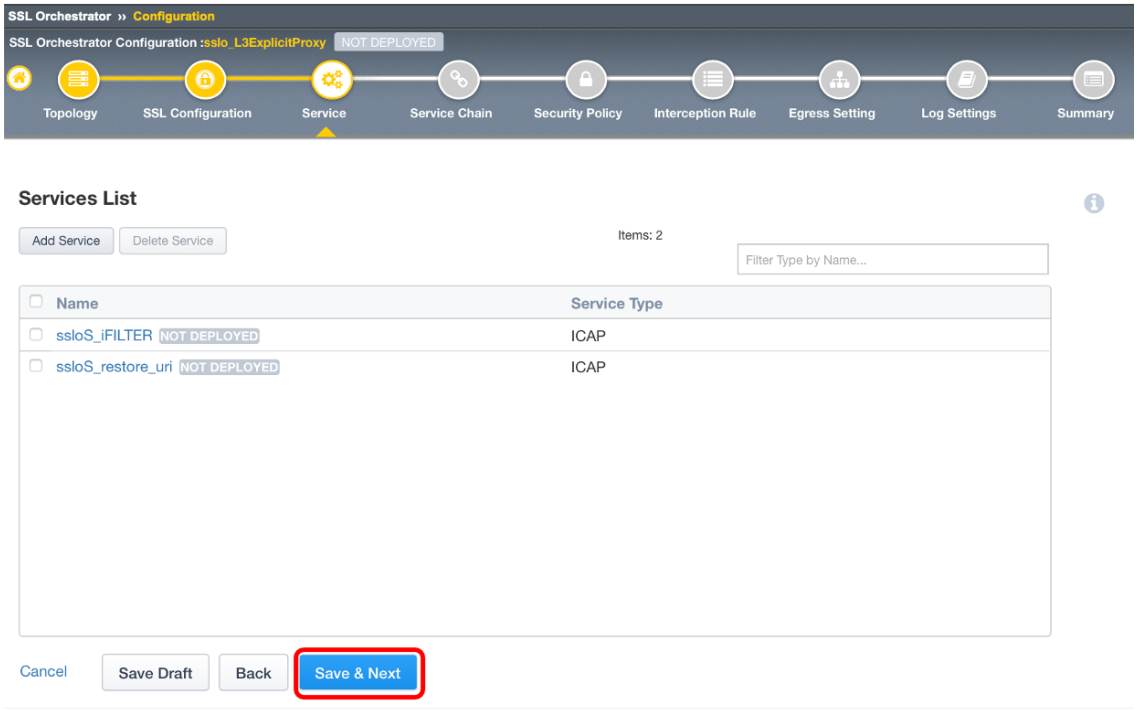
13. **任意の名前** を設定し、**ICAP Devices** に **任意の IP アドレス** と **任意のポート番号** を設定し、*Done* を押します。（こちらのサービスは実際には利用しませんので、設定する情報は何でも構いません。）



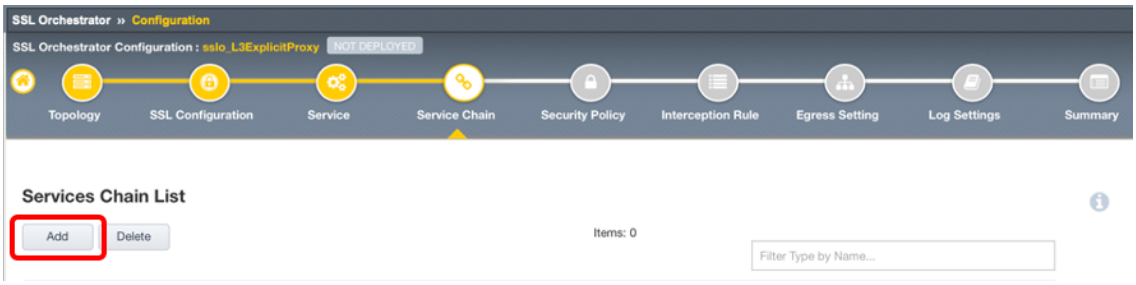
14. **ICAP Policy** に既に作成済みのヘッダをもとに戻すための **Local Traffic Policy** を選択し、*Save* を押します。



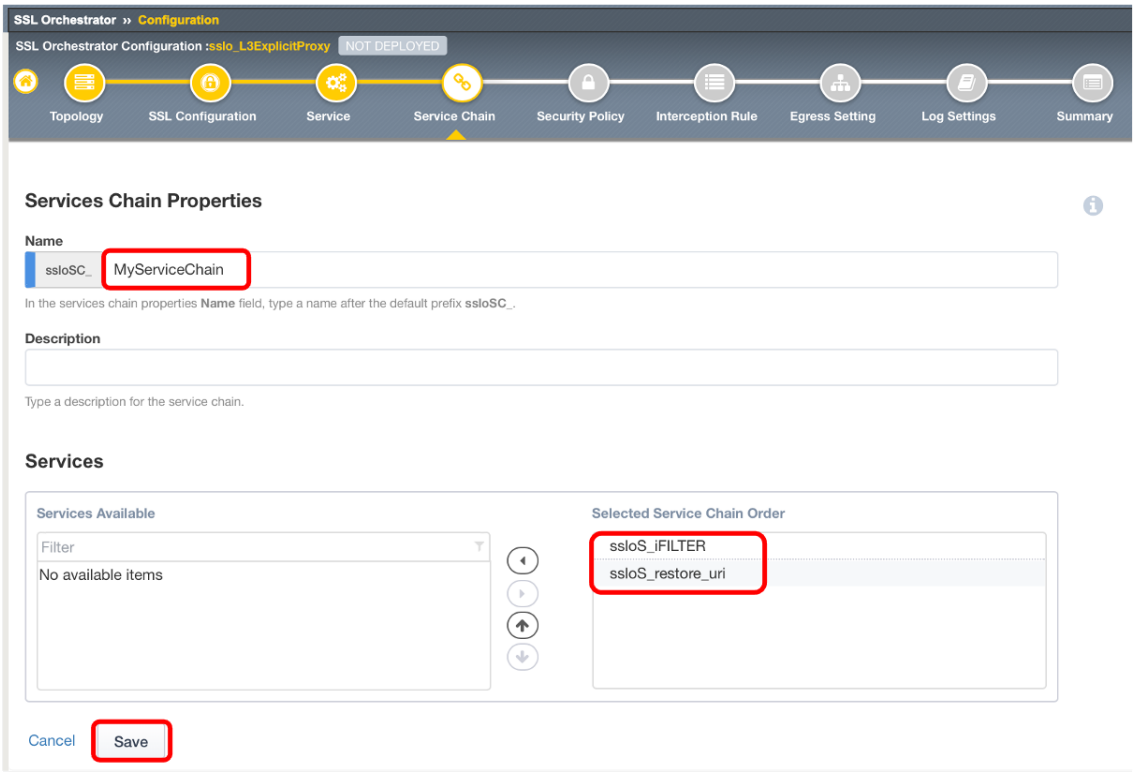
15. *Save* **&** *Next* ボタンを押します。



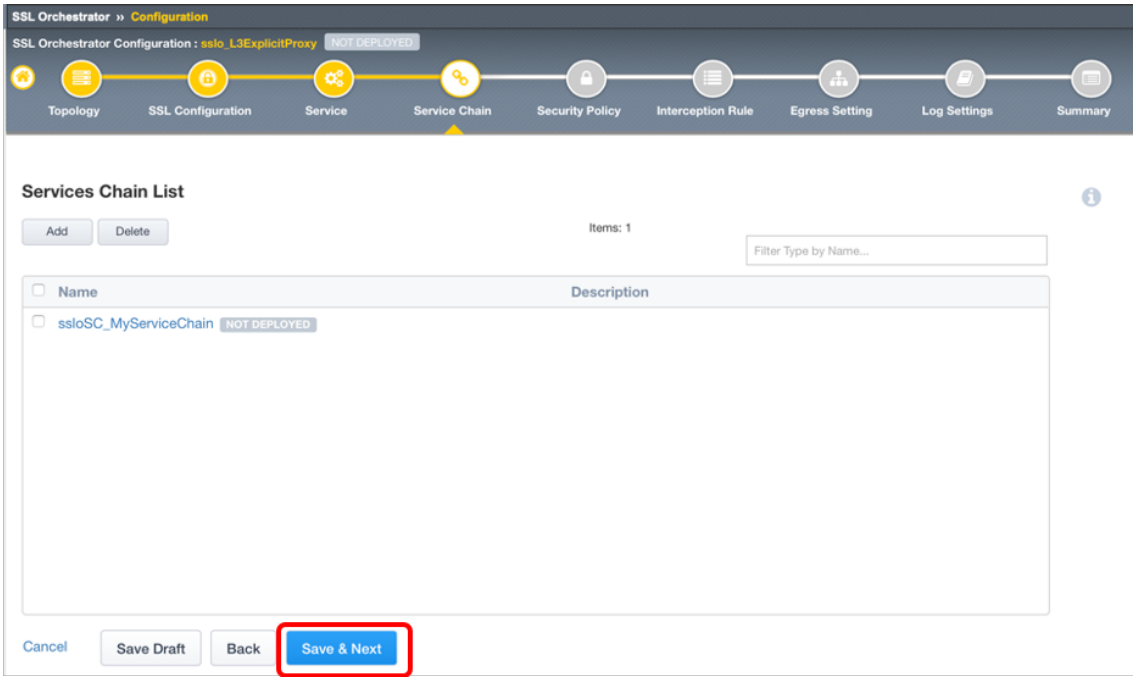
16. **Service Chain List** で *Add* を押します。



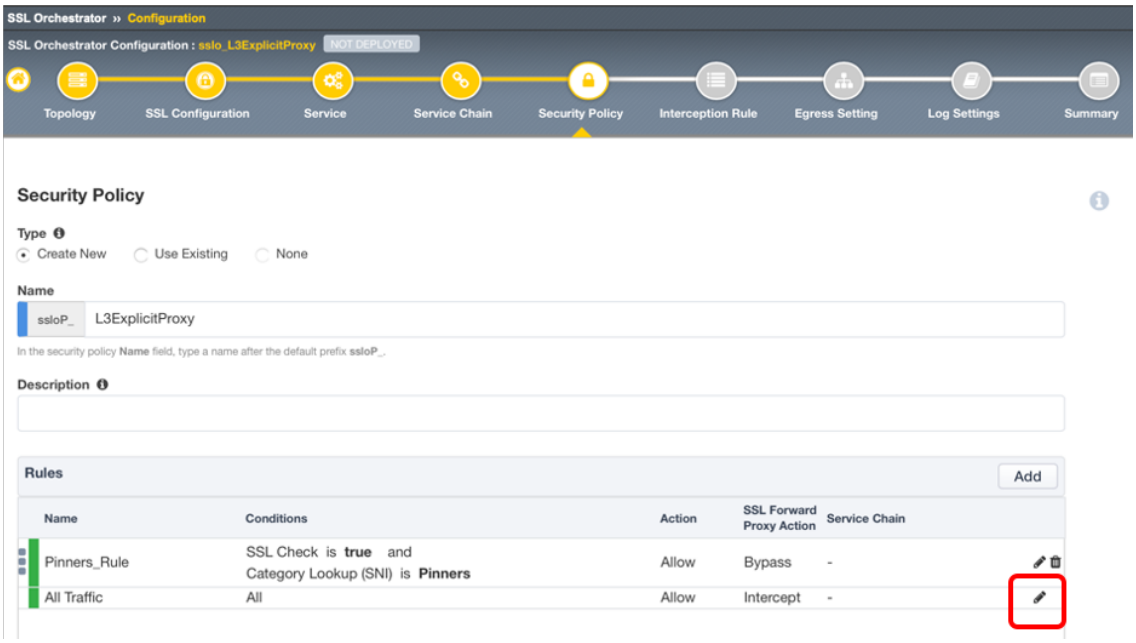
17. **任意の名前** を設定し、先程作成したサービスを右に移動させ、*Save* ボタンを押します。(i-FILTER のサービスが上にくるようにします。)



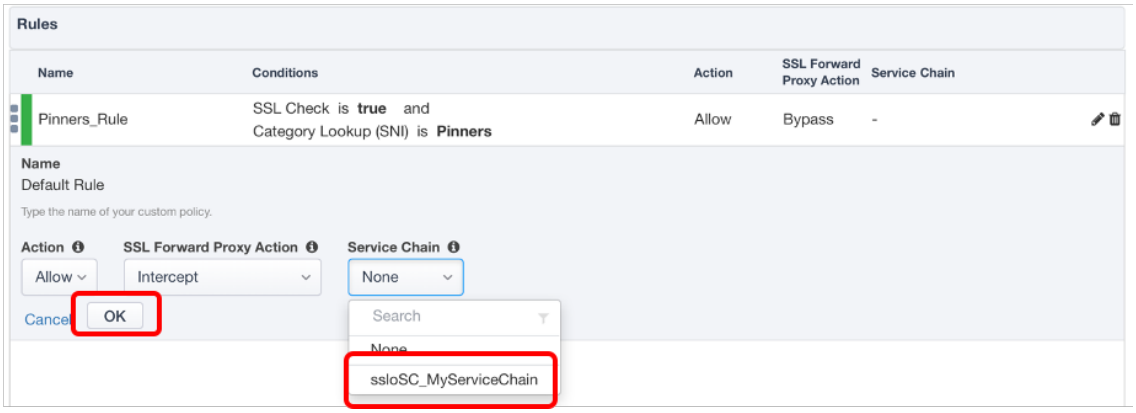
18. **Service Chain** ができたことを確認し、*Save* **&** *Next* ボタンを押します。



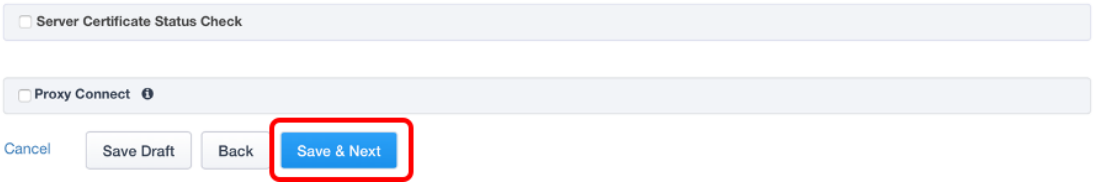
19. All Traffic の **ペンマーク** をクリックします。



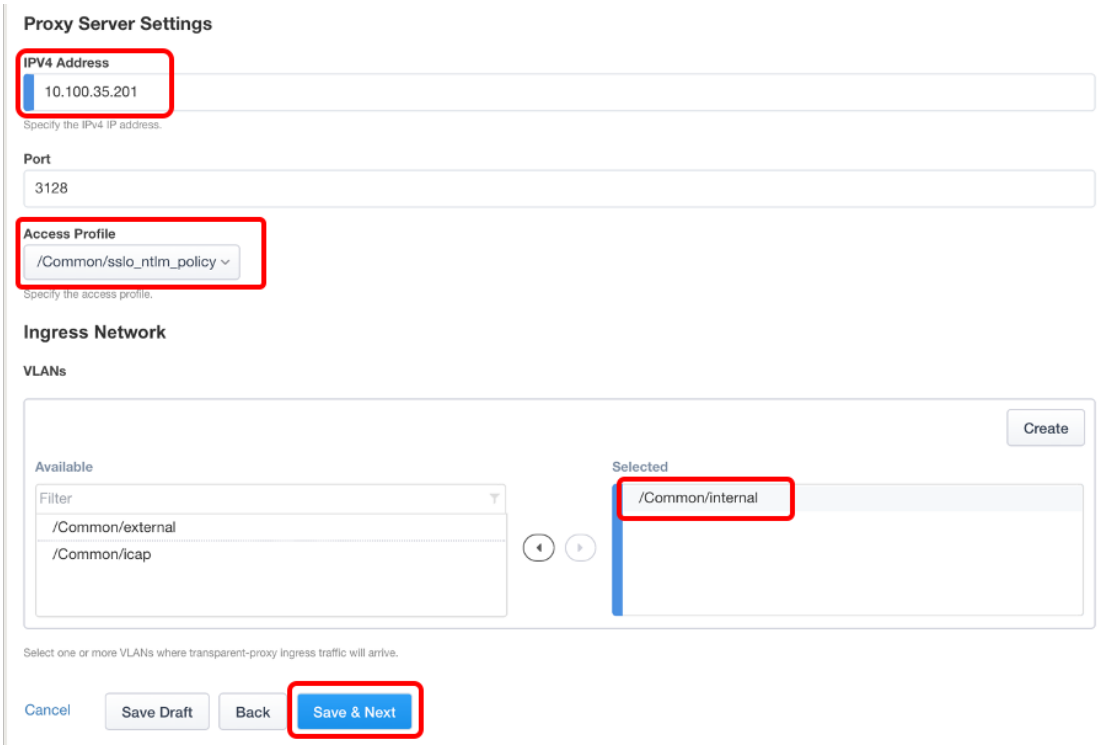
20. 先程作成した **Service Chain** を選択し、*OK* ボタンを押します。



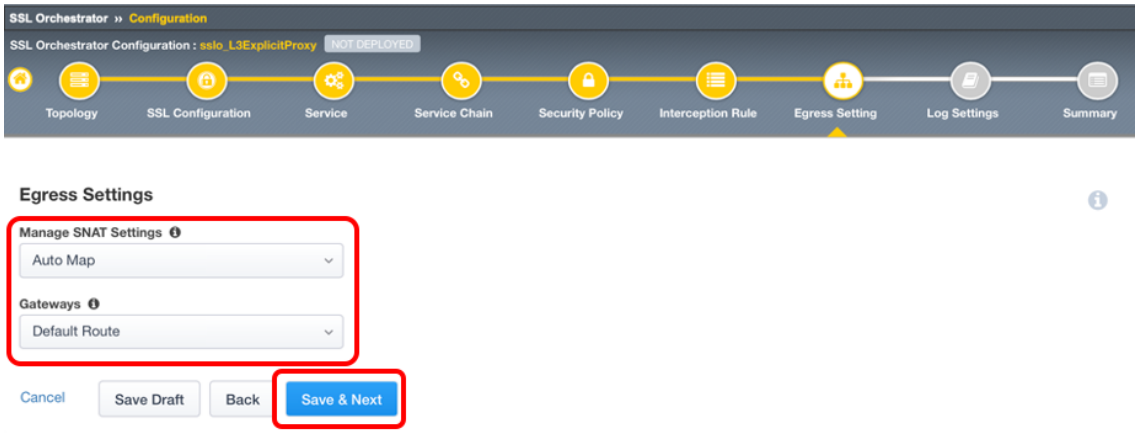
21. *Save* **&** *Next* ボタンを押します。



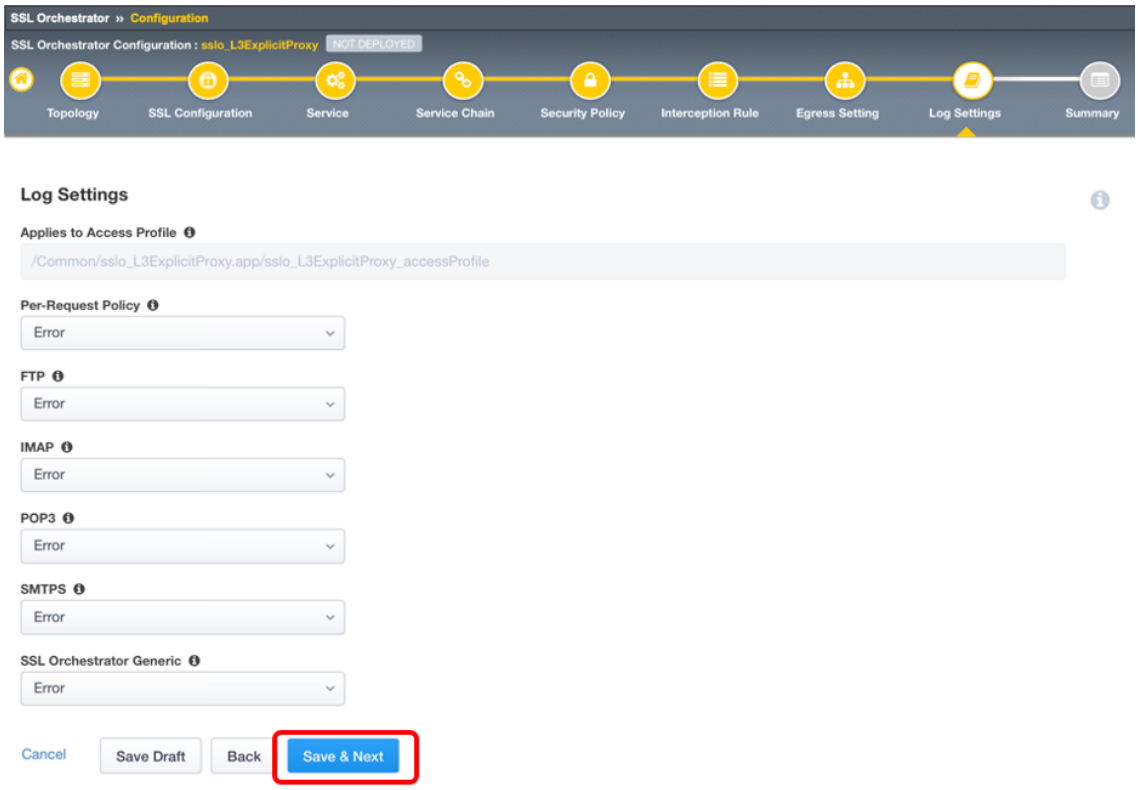
22. **Proxy Server Settings** にクライアントからプロキシとしてアクセスさせる IP アドレスを入力し、既に作成済みの **Access Profile** を選択し、**Ingress Network** として、クライアントからアクセス可能な **VLAN** を選択し、*Save* **&** *Next* ボタンを押します。



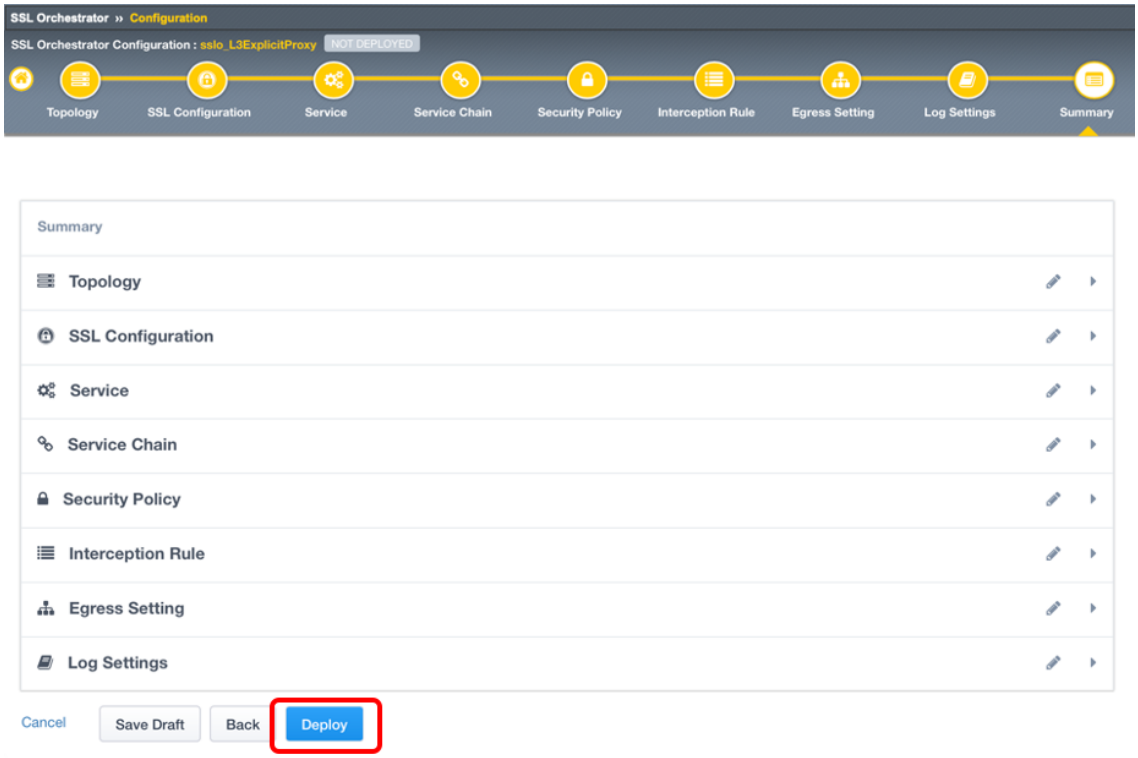
23. 本テスト構成では、**Manage SNAT Settings** で **Auto Map**、**Gateways** で **Default Route** を選択し、*Save* **&** *Next* ボタンを押します。(設定は検証環境に合わせてください。)



24. *Save* **&** *Next* ボタンを押します。

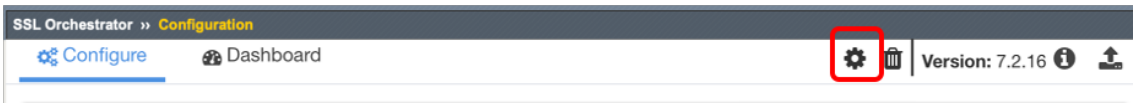


25. 必要に応じて、設定内容を見直し、 *Deploy* ボタンを押します。

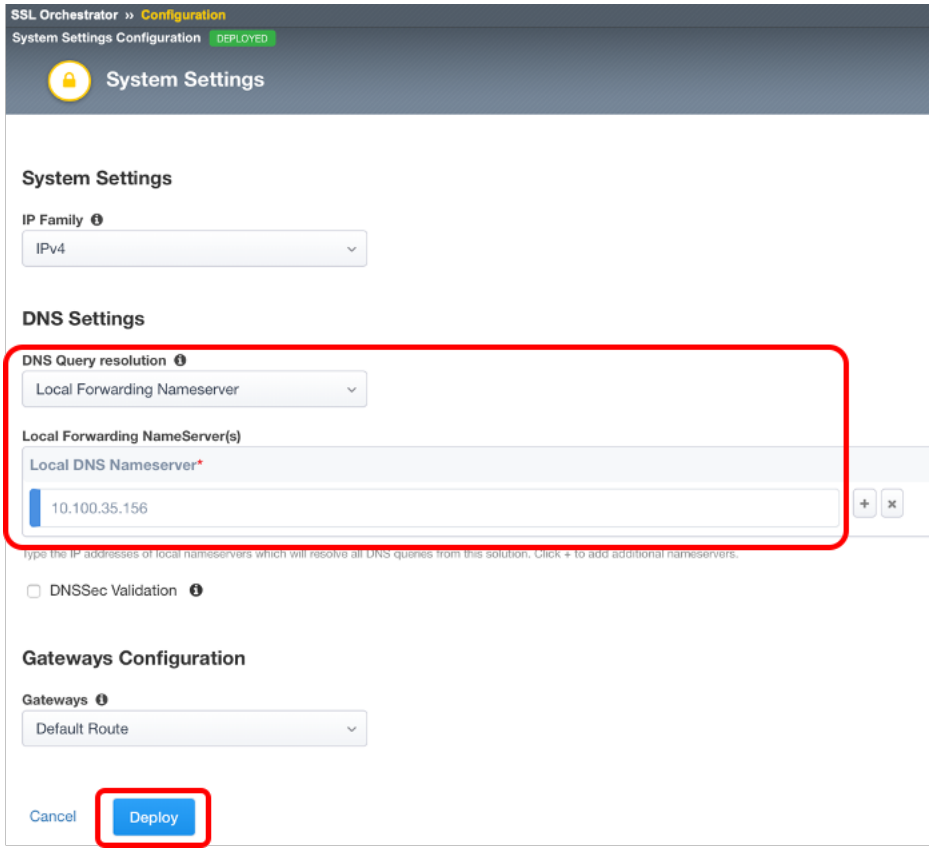


26. *OK* ボタンを押し、Deploy に成功すると以下のような緑色の **DEPLOYED** マークが表示されます。

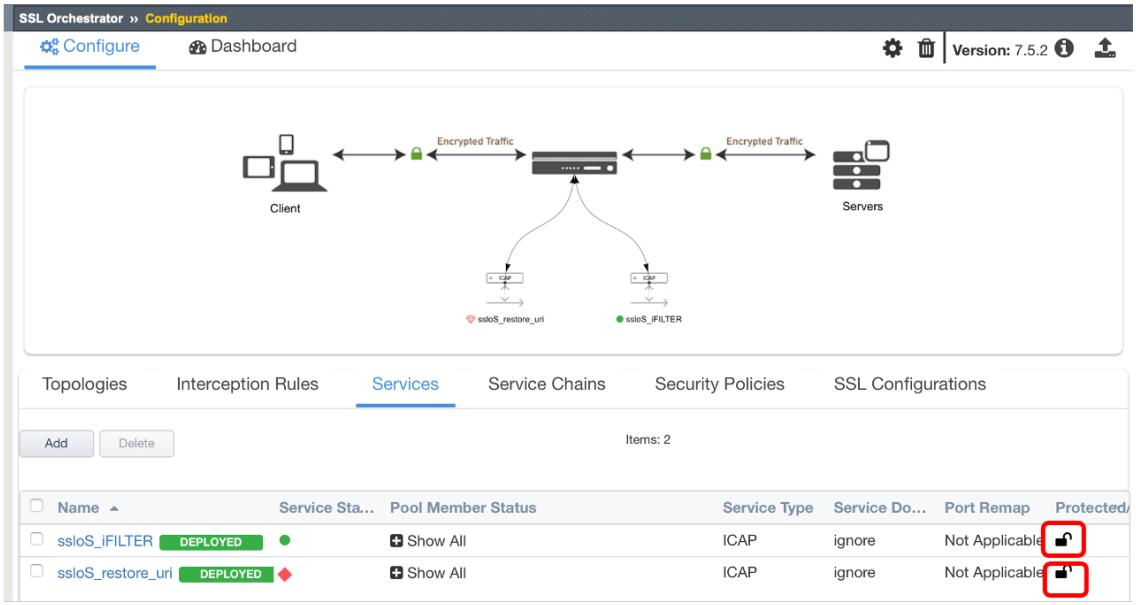
27. 右上の **System Settings** アイコンを選択します。



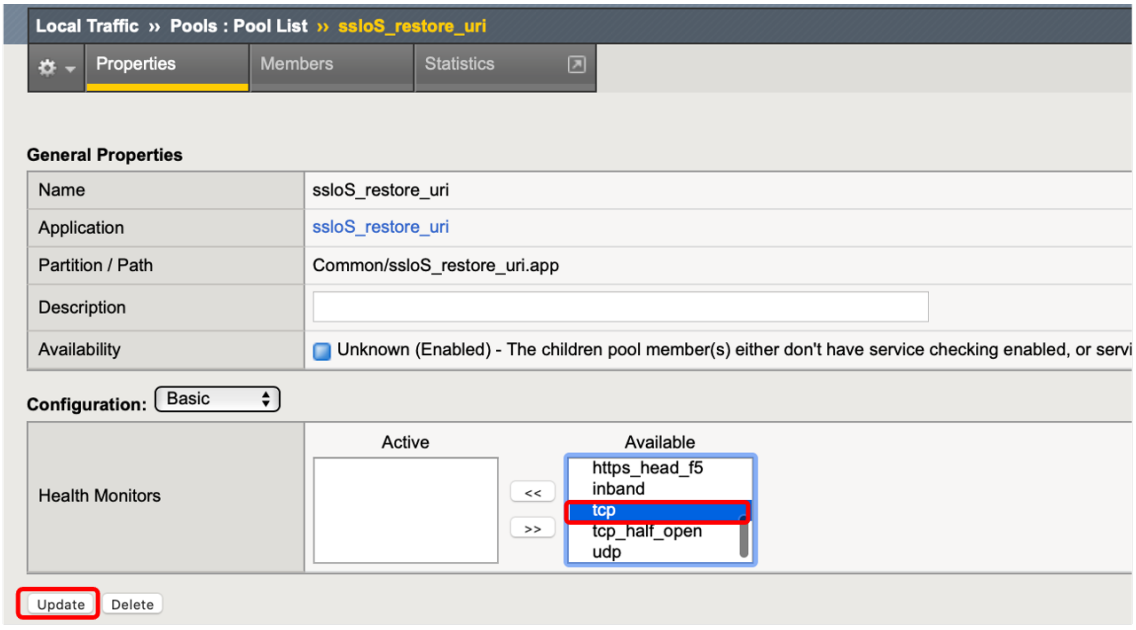
28. SSLO が Explicit Proxy として利用する **DNS** を設定し、*Deploy* を押します。



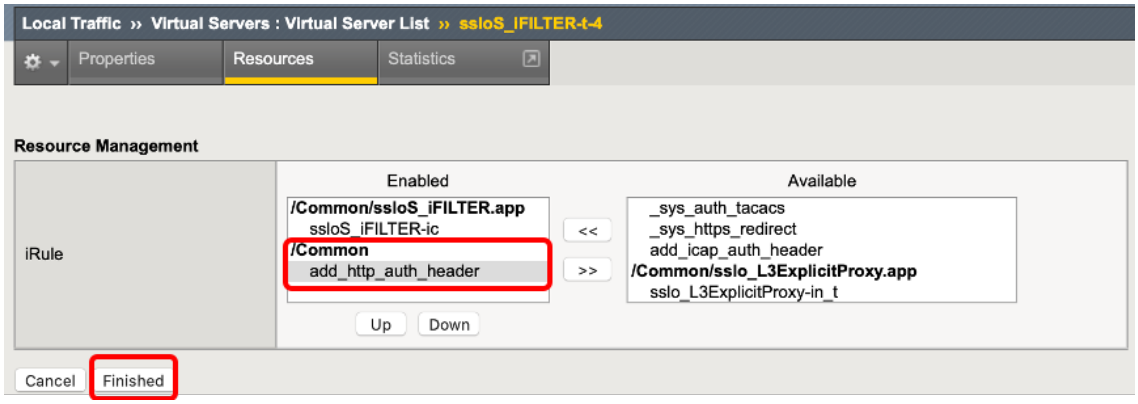
29. i-FILTER と連携するサービスに関しては認証ヘッダを挿入するために、また、ヘッダをリストアするサービスに関してはモニタを停止するため、Protected の **鍵** マークを外します。



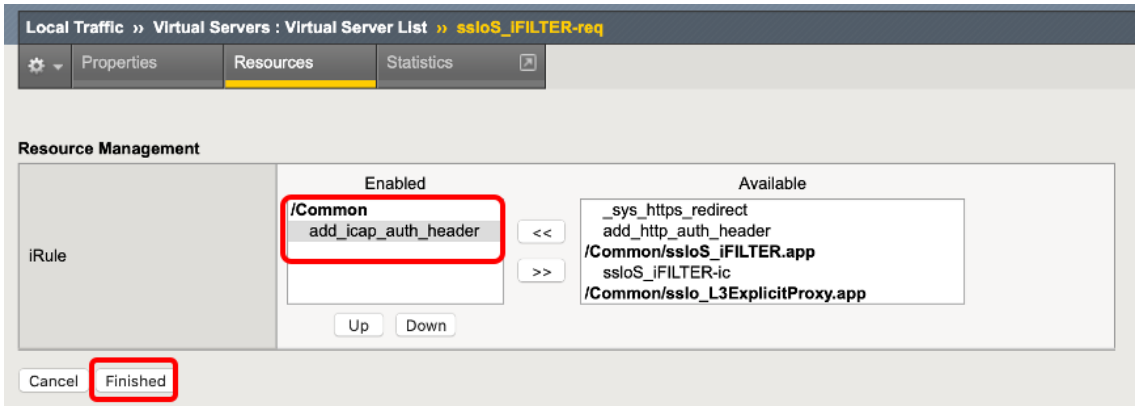
30. ヘッダをリストアするサービスに紐づく Pool のモニターの情報をはずします。（設定を外す前は存在しないサーバに対して、モニターを投げているので、Availability が Offline となっているはずです。）



31. 既に作成済みの **iRule** を追加します。**Local Traffic >> Virtual Servers>>** において、**ssloS_XXXX(任意)-t-4** という名称の Virtual Server を選択し、**Resources** タブを選択、HTTP リクエストヘッダに認証情報を挿入するための **iRule** を選択し、*Finished* を押します。



32. ICAP リクエストヘッダに加えるための **iRule** を追加します。**Local Traffic >> Virtual Servers** において、**ssloS_XXXX(任意)-req** という名称の Virtual Server を選択し、**Resources** タブを選択、ICAP リクエストヘッダに認証情報を挿入するための **iRule** を選択し、*Finished* を押します。



注釈:

- セキュリティデバイスが ICAP サービス、HTTP サービスの場合、SSL 復号していないトラフィックをサービスチェーンに流せません。

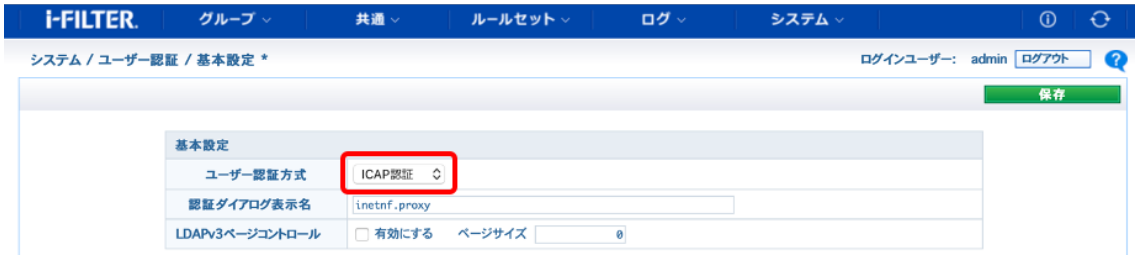
1.1.11 i-FILTER ICAP 版（Linux 版）の設定

i-FILTER のインストールは終了しているものとします。

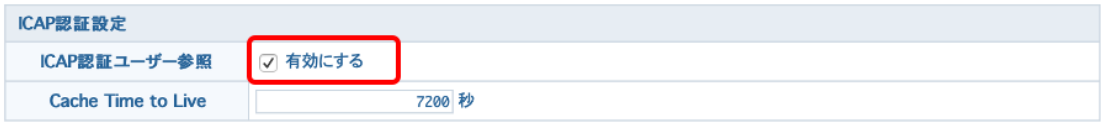
ユーザ認証の設定

1. **システム／ユーザ認証／基本設定 を選択します。**

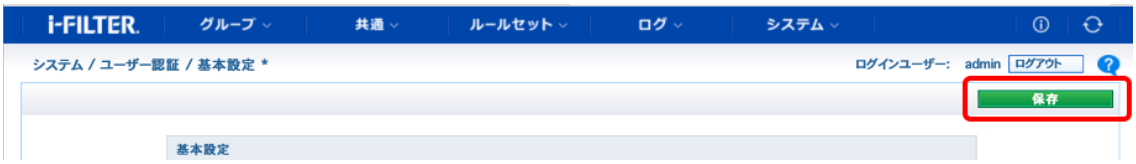
2. **基本設定** の **ユーザ認証方式** において **ICAP 認証** を選択します。



3. **ICAP 認証設定** の **ICAP 認証ユーザ参照** において、**有効にする** をチェックします。



4. 右上の **保存** ボタンを押します。



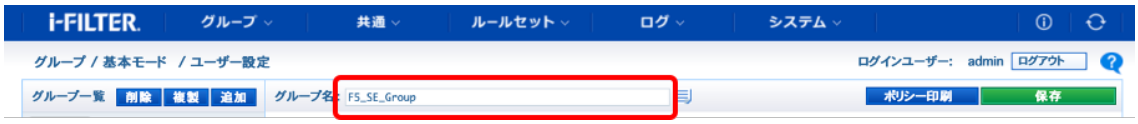
5. 次に、**システム／ユーザ認証／ LDAP サーバ設定** を選択します。本ガイドでは認証ユーザ情報を AD サーバから取り込むため、以下のように設定し、**保存** ボタンを押します。

項目	設定例
アドレス:ポート番号	ご利用の AD 情報を登録
BIND	BIND 有効
BIND DN	ご利用の管理者 BIND DN 情報を登録 l 設定例：cn=Administrator,cn=Users,dc=f5jplab,dc=local
BIND パスワード	ご利用のドメイン管理者のパスワードを登録
Search Base	ご利用のドメイン情報を登録 l 設定例：dc=f5jplab,dc=local
Search Filter	(sAMAccountName=%1)

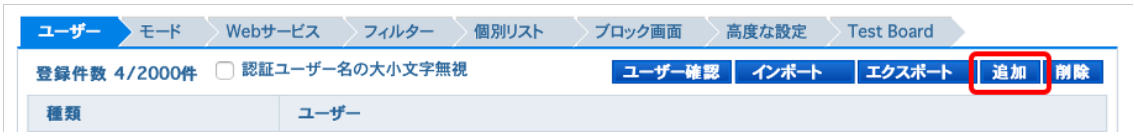
グループポリシーの作成、ユーザ情報の取り込み

1. **グループ／基本モード／モード設定** において、**追加** ボタンを押します。

2. **グループ名** を分かりやすいグループ名に変更します。



3. **ユーザ** タブにおいて、**追加** ボタンをクリックします。



4. **認証ユーザ参照** を選択します。

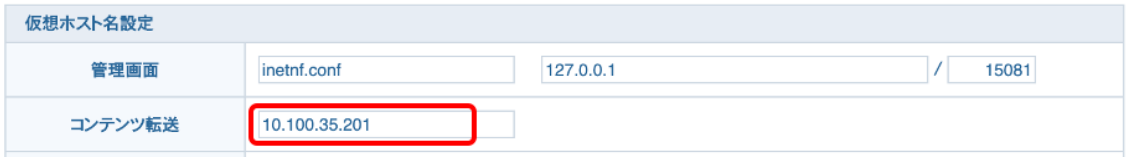
5. 検索しやすい検索条件を設定し、AD から取り込みたい **ユーザ名** を追加します。

6. **Web サービス** タブを選択し、制御したいサービスを選択し、ユーザに利用不許可とする機能に対し、**ブロック** を選択します。（注：下記イメージはあくまでも一例です。）

7. **個別リスト** タブを選択し、個別ブロックしたい URL/URI を追加し、右上の **保存** ボタンを押します。（注：下記イメージはあくまでも検証目的で設定した内容です。）

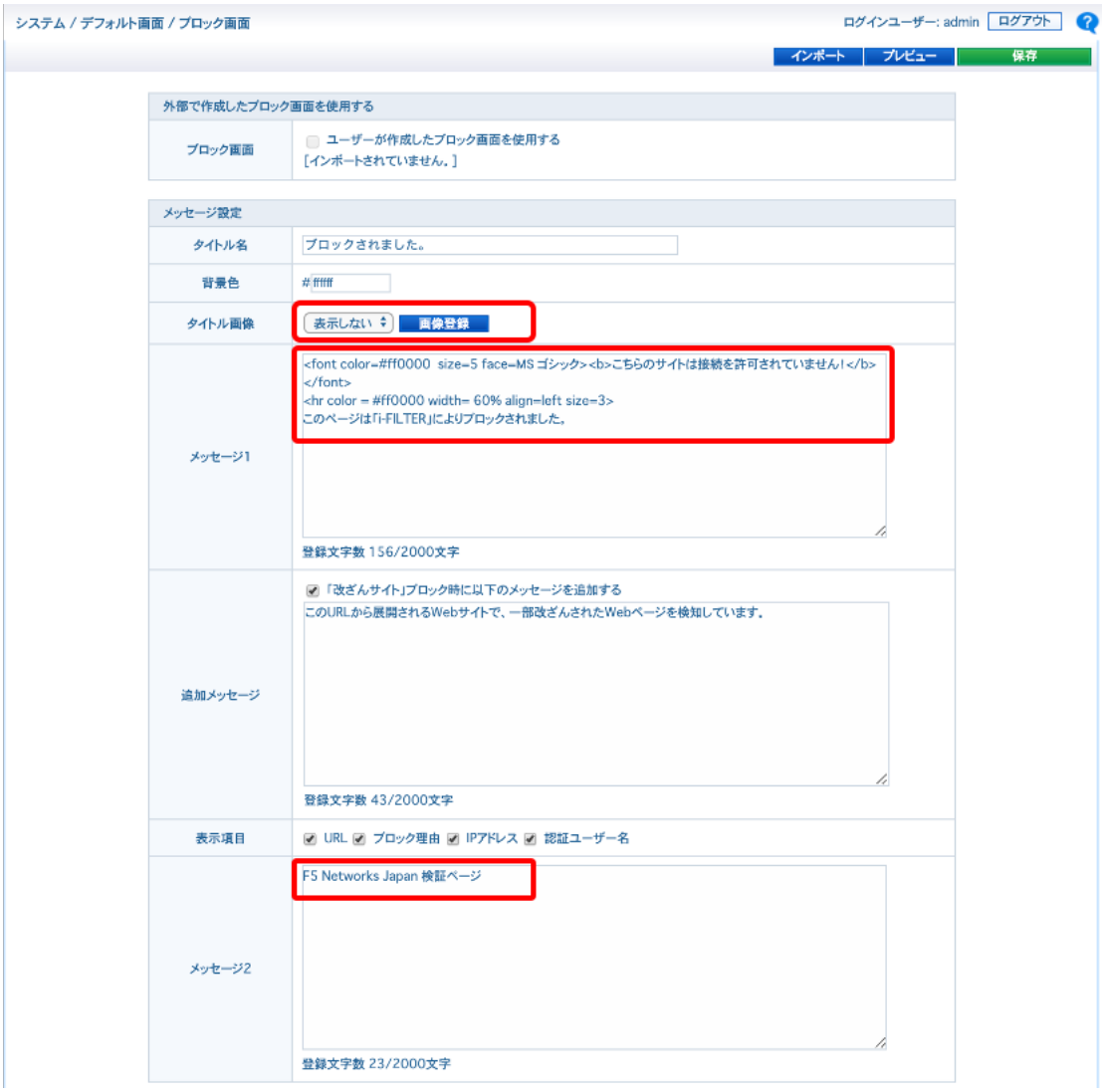
ブロック画面のタイトル画像連携設定

1. **システム／システムパラメーター／動作設定** において、**仮想ホスト名設定** の **コンテンツ転送** の値を、で設定した SSLO の Interception Rule の Destination Address の IP アドレスに変更します。



2. **i-FILTER v10.40R01 より、WEB 接続ブロック時にユーザに表示するブロック画面中のタイトル画像は、HTTPS 接続アク**

3. ここでは、簡易的にブロック画面中にタイトル画像を表示せず、**システム／デフォルト画面／ブロック画面**において、タイトル画像を **表示しない** を選択し、**保存** ボタンを押します。必要に応じて、メッセージも変更します。



1.1.12 クライアントからの接続テスト

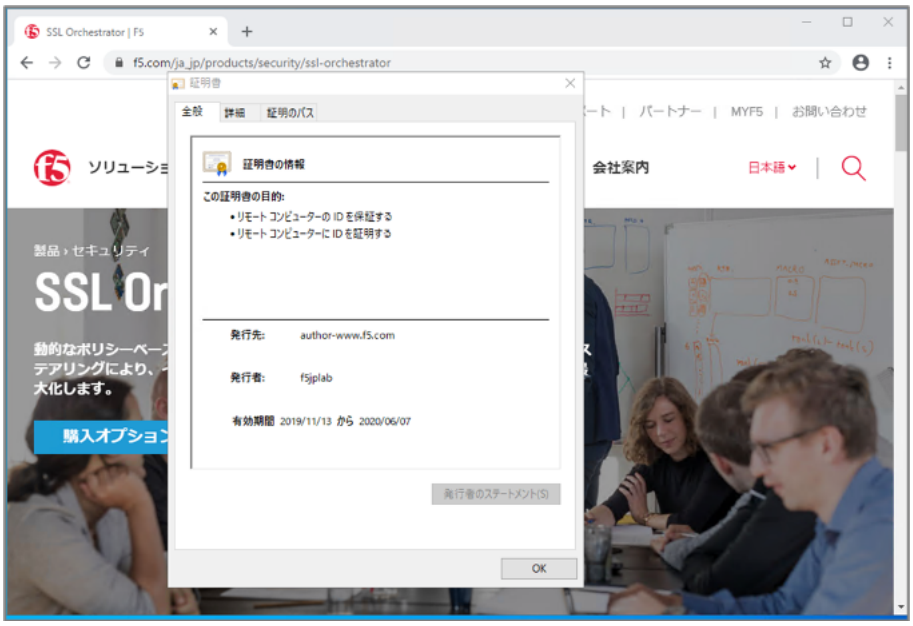
1. クライアントに SSLO の **CA 証明書** をインストールします。

2. コマンドラインで確認する場合は、以下を利用します。

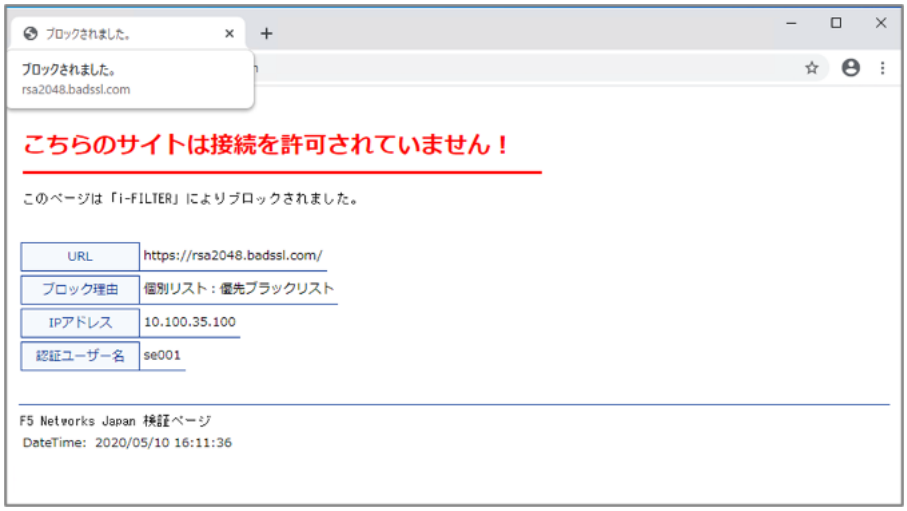
```
curl -vk --proxy 10.100.35.201:3128 -proxy-ntlm -proxy-user 'se001:ilovef5!'␣
↪https://xxxx.xxx
```

3. ブラウザで確認する場合は、プロキシ設定に SSLO を加えてから WEB 接続を確認します。

4. WEB 通信が成功するか確認します。サーバ証明書が SSLO の CA 証明書で **書き換えられている** ことを確認します。



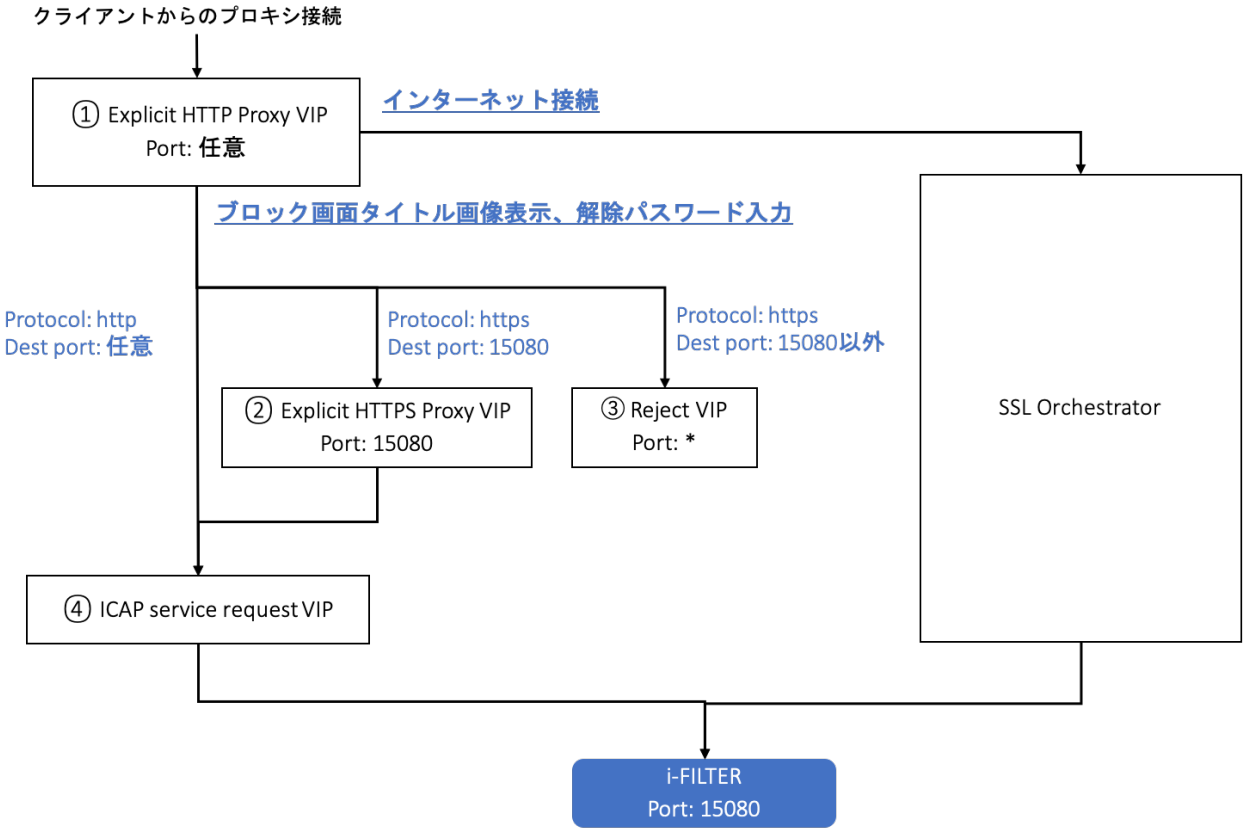
5. ブラックリストの宛先への通信が **ブロック** されることを確認します。



6. SSL 復号除外サイトの通信が成功するか確認します。サーバ証明書が SSLO の CA 証明書で **書き換えられていない** ことを確認します。

1.1.13 i-FILTER ブロック画面のタイトル画像の表示、パスワードブロック解除の設定

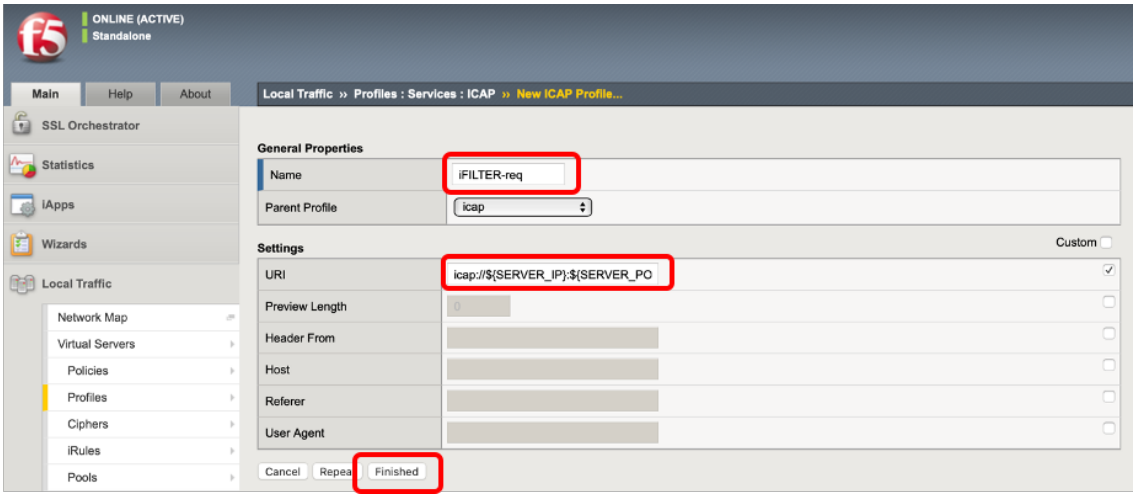
1.11.3. にて、SSLO 連携をするとブロック画面のタイトルが標準では表示されないと記載しましたが、以下の設定を追加することによって、表示させることが可能です。本設定を行うことにより、パスワード入力によるブロック解除も連携が可能となります。上記を実現するためには、下図の 4 つの Virtual Server が追加で必要となります。ここでは、下図の âŠčâŚąâŚăâŚć の順で作成していきます。



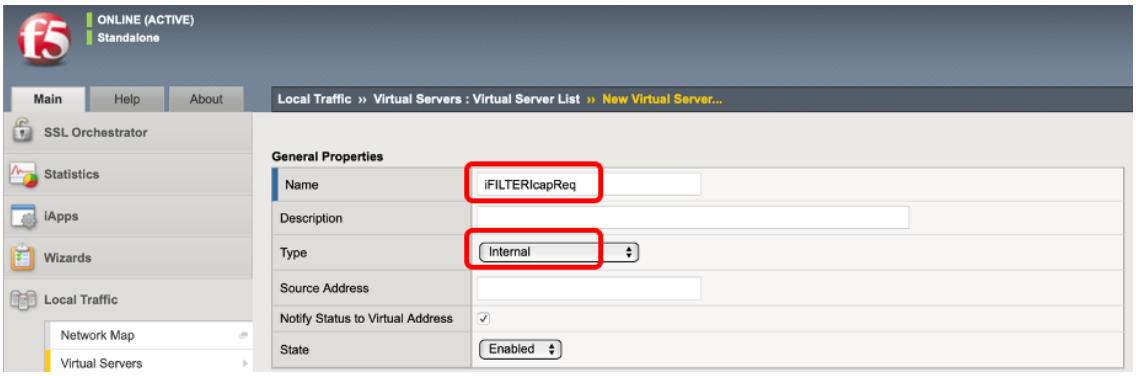
ICAP service request 用 Virtual Server の作成

1. まずは ICAP 用のプロファイルを以下のように作成します。 **Local Traffic >> Profiles >> Services >> ICAP** にて、*Create* ボタンを押します。**任意の名前** を入力し、**URI** には以下のように入力し、*Finished* ボタンを押します。

   icap://${SERVER_IP}:${SERVER_PORT}/REQMOD

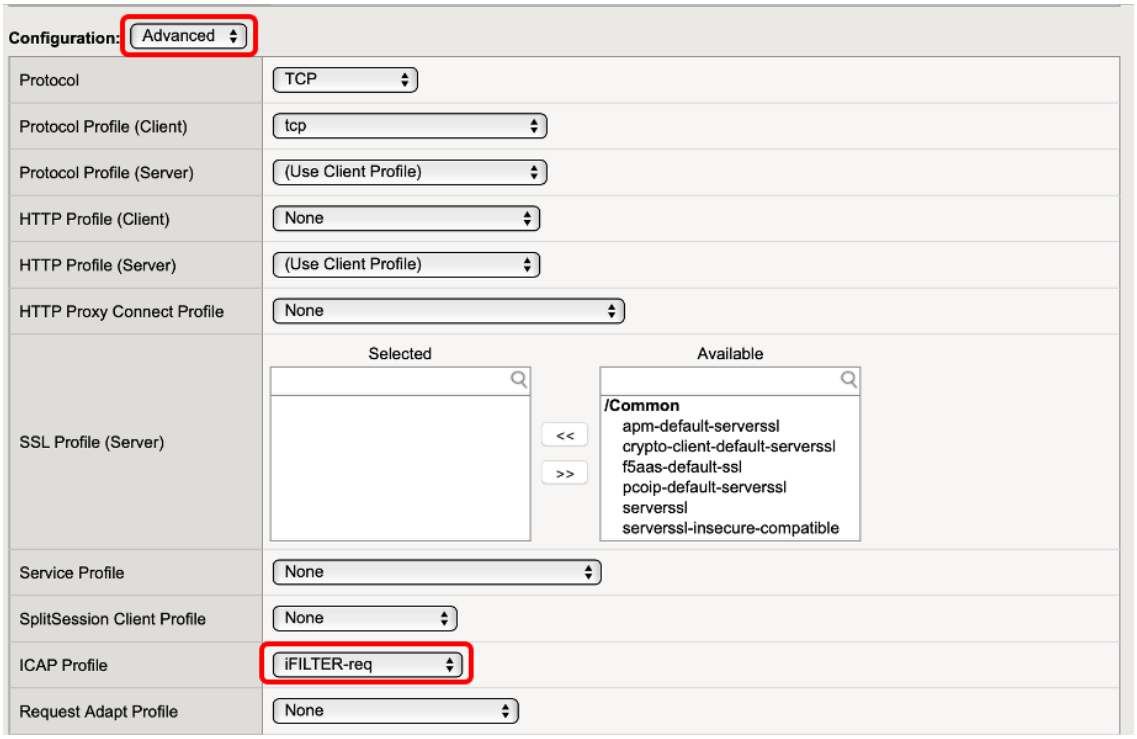


2. 次に ICAP service request 用の Virtual Server を作成します。**Local Traffic >> Virtual Servers** にて、*Create* ボタンを押します。**任意の名前** を入力し、**Type** は **Internal** を選択します。

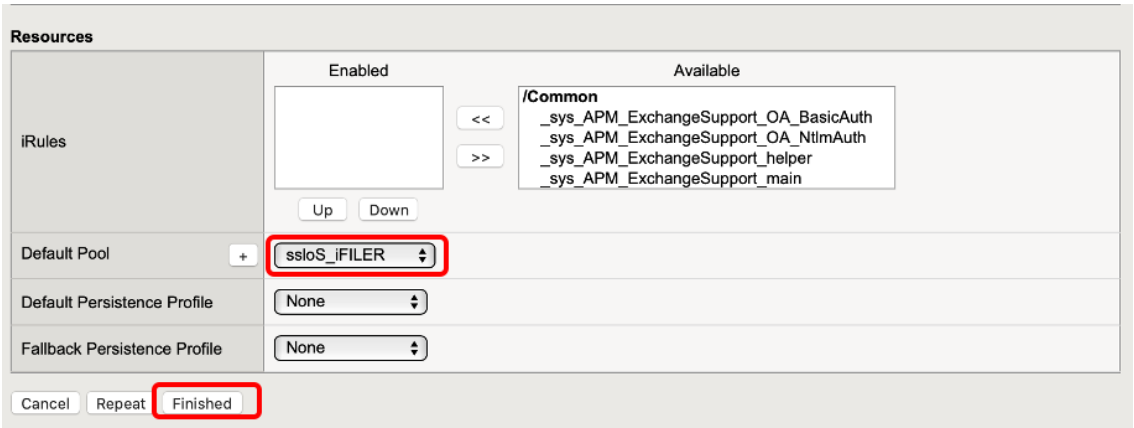


3. **Configuration** にて **Advanced** を選択し、**ICAP Profile** にて、作成済みの **ICAP プロファイル** を選択し

ます。

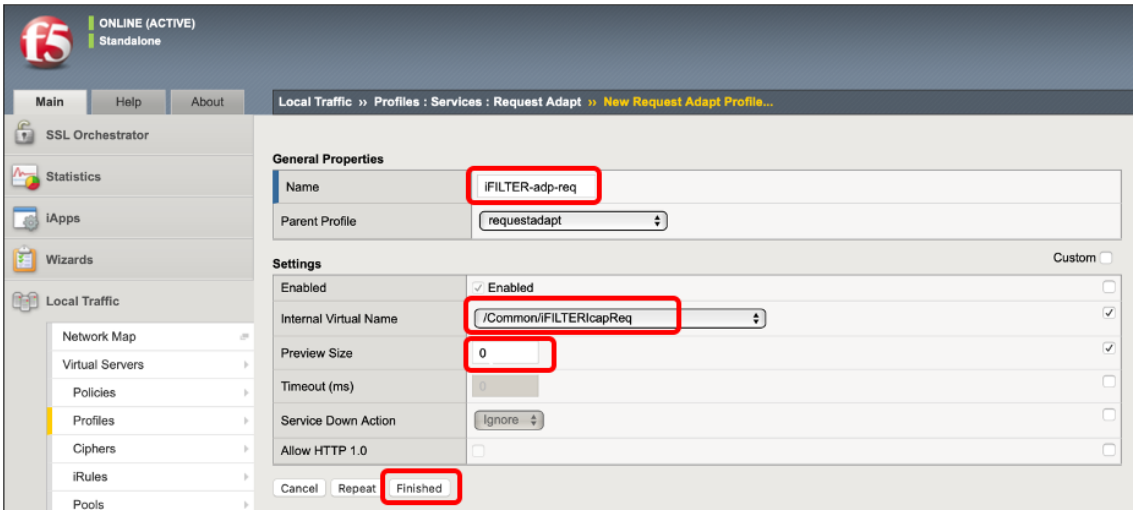


4. 最後に、**Default Pool** にて、SSLO Guided Configuration で作成済みの **Pool** を選択して、*Finished* ボタンを押します。

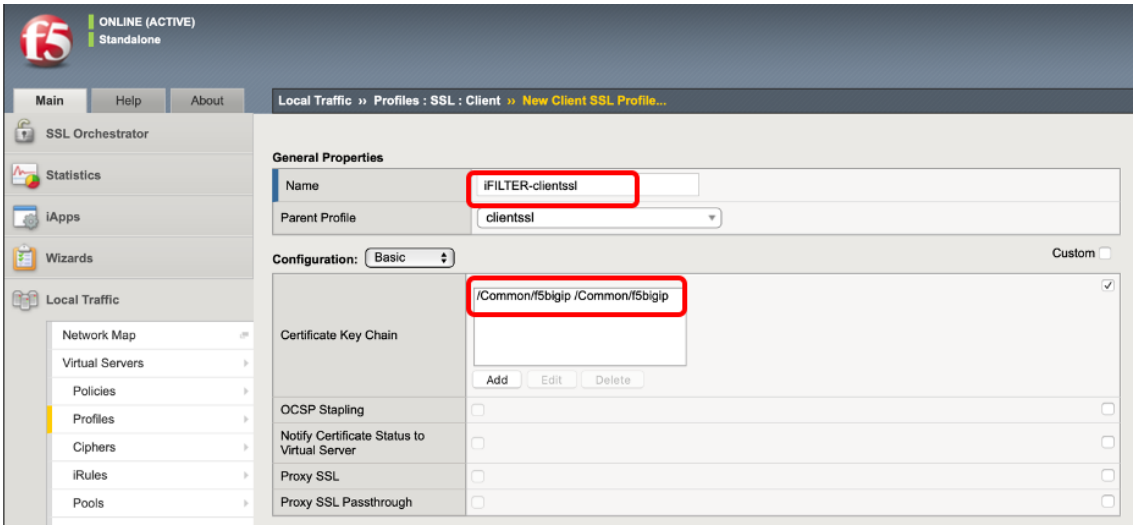


Explicit HTTPS Proxy 用 Virtual Server の作成

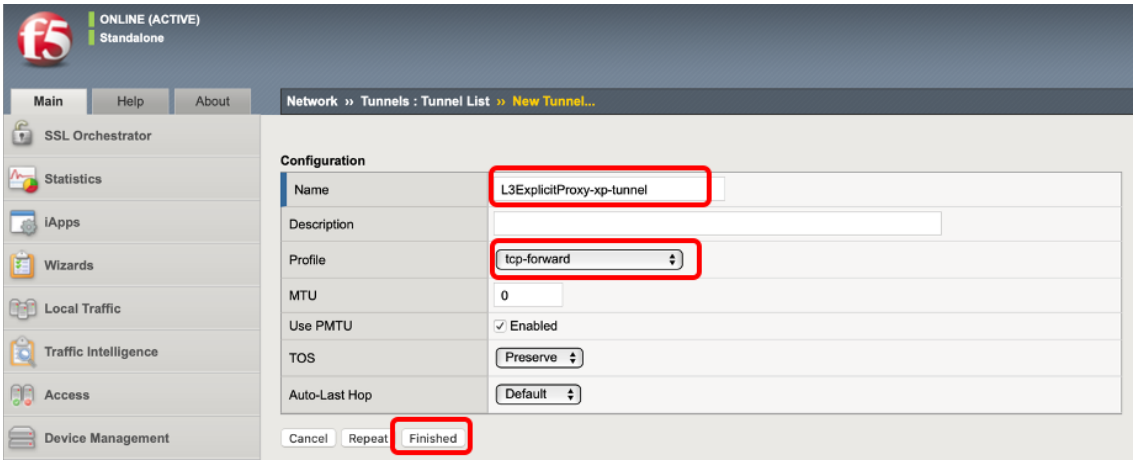
1. まず、ICAP 用の Virtual Server を呼び出すための Request Adapt プロファイルを作成します。**Local Traffic >> Profiles >> Services >> Request Adapt** にて、*Create* ボタンを押します。**任意の名前** を入力し、**Internal Virtual Name** にて、作成済みの **Virtual Sever** を選択し、**Preview Size** にて、**0** を入力し、*Finished* ボタンを押します。



2. 次に、SSL サイトにおけるブロック時に、ブロック画面内のタイトル画像に SSL 接続するための ClientSSL プロファイルを作成します。**Local Traffic >> Profiles >> SSL >> Client** にて、*Create* ボタンを押します。**任意の名前** を入力し、タイトル画像を SSL 経由で表示させるための **サーバ証明書** と **秘密鍵** を選択し、*Finished* ボタンを押します。（BIG-IP にて SSL オフロードを行うため、**System >> Certificate Management >> Traffic Certificate Management** にて、予め **サーバ証明書** と **秘密鍵** の登録が必要となります。）



3. 次に、Proxy 用の Tunnel を作成します。**Network >> Tunnel List** にて、*Create* ボタンを押します。**任意の名前** を入力し、**Profile** にて、**tcp-forward** を選択し、*Finished* ボタンを押します。

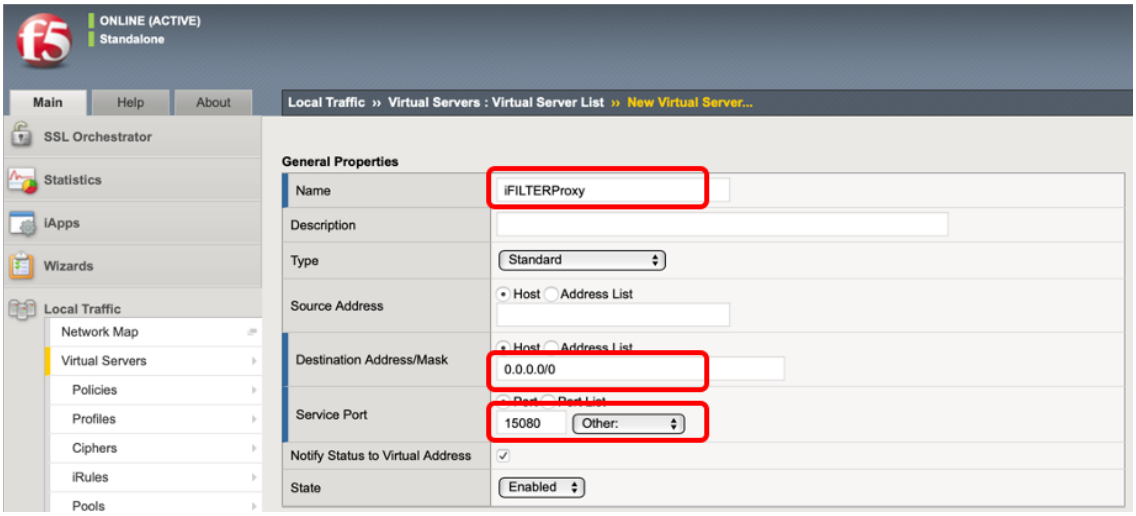


4. 次に、HTTPS トラフィックにおける URI 書き換え用の iRule を作成します。**Local Traffic >> iRules** にて、*Create* ボタンを押します。 **任意の名前** を入力して、 **Definition** に以下サンプル **iRule** を入力し、*Finished* ボタンを押します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。）

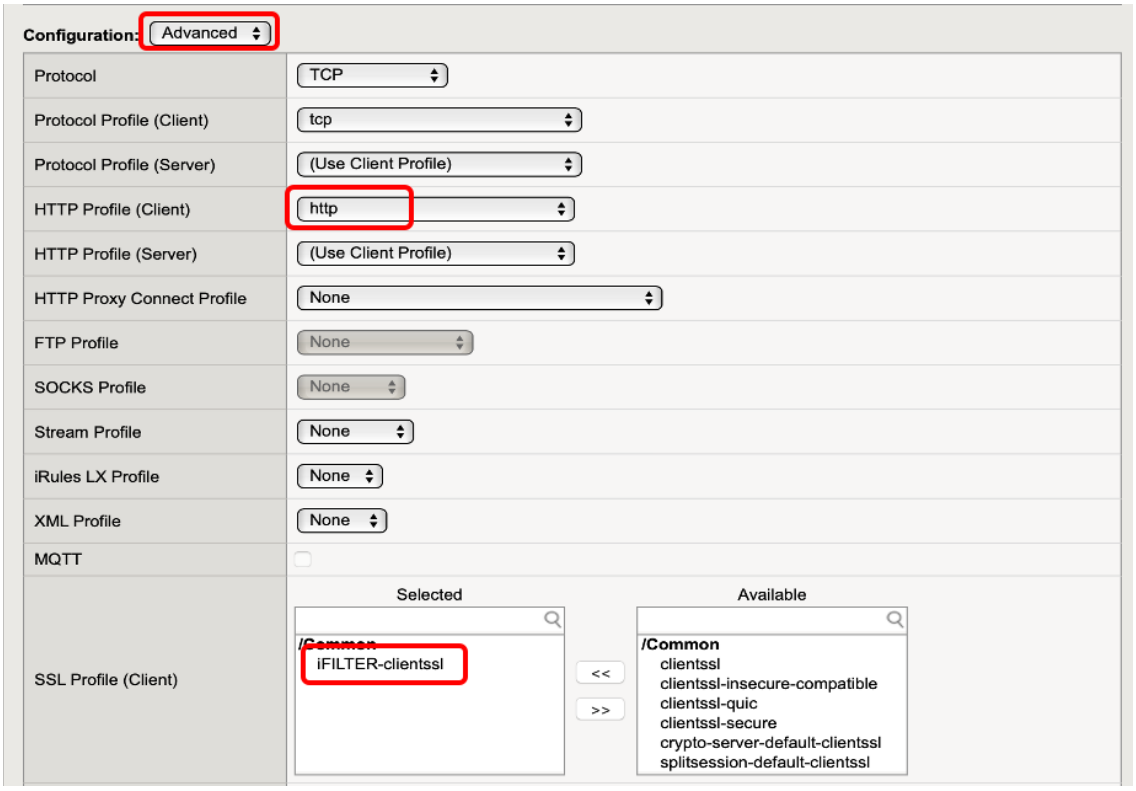
例）URI 書き換え用の iRule

```
###  Add this iRule to Explicit HTTPS Virtual Server ###
when HTTP_REQUEST {
    set OLDURI [HTTP::uri]
    HTTP::uri "https://[HTTP::host]$OLDURI"
}
```

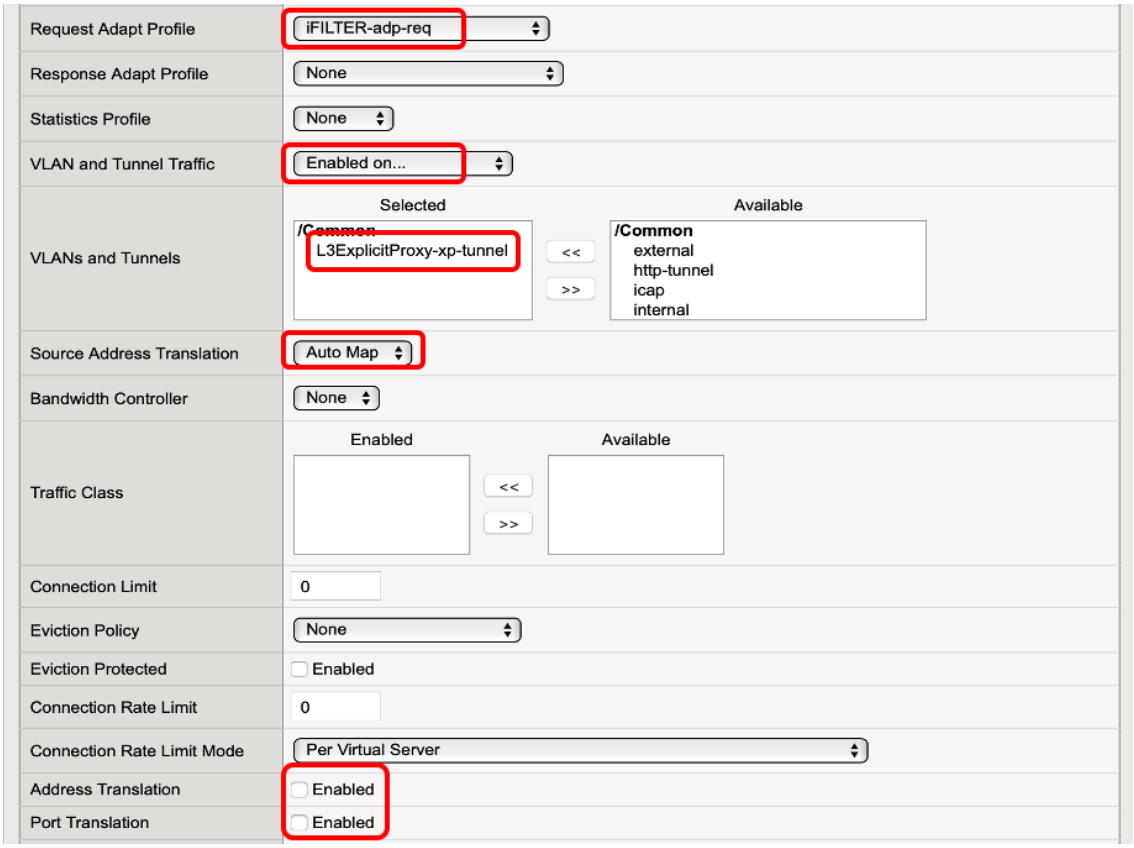
5. 次に、SSL 用の Explicit HTTPS Virtual Server を作成します。**Local Traffic >> Virtual Servers** にて、*Create* ボタンを押します。**任意の名前** を入力し、**Destination Address/Mask** にて、**0.0.0.0/0** を入力、**Service Port** にて、**15080** と入力します。



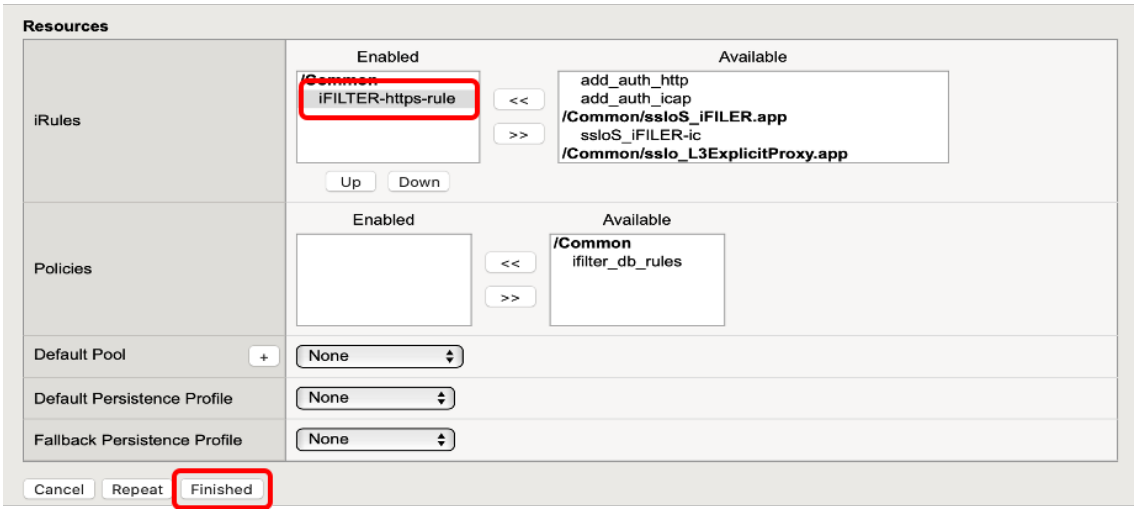
6. **Configuration** にて、**Advanced** を選択し、**HTTP Profile** にて、**http** を選択、**SSL Profile (Client)** にて、作成済みの **clientssl プロファイル** を選択します。



7. **Request Adapt Profile** にて、作成済みの **プロファイル** を選択し、**VLAN and Tunnel Traffic** にて、**Enable on...** を選択し、**VLANs and Tunnels** にて作成済みの **Tunnel** を選択し、**Source Address Translation** にて **Auto Map** を選択し、**Address Translation** と **Port Translation** のチェックをはずします。

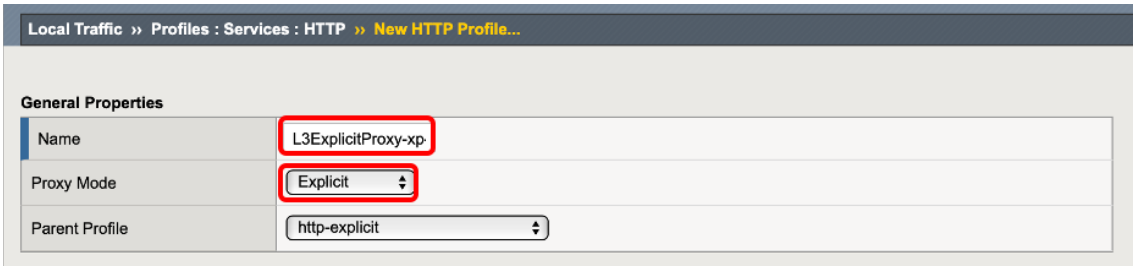


8. **iRules** にて、作成済みの **iRule** を選択し、*Finished* ボタンを押します。



Explicit HTTP Proxy 用 Virtual Server の作成

1. まず、プロキシ用の Explicit Profile を作成します。**Local Traffic >> Profiles >> Services >> HTTP** にて、*Create* ボタンを押します。**Proxy Mode** にて、**Explicit** を選択します。



2. **DNS Resolver** にて、SSLO Guided Configuration で作成した **DNS Resolver** を選択し、**Tunnel Name** にて、作成済みの **Tunnel** を選択し、*Finished* ボタンを押します。

3. 次に、以下の 2 つの iRule を作成します。

    - インターネット接続用の HTTP/HTTPs トラフィックと i-FILTER ブロックタイトル画面接続トラフィックを分ける iRule
    - 上記後者のトラフィックにおいて、i-FILTER 向けに URI を書き換える iRule

    Local Traffic >> iRules にて、*Create* ボタンを押します。**任意の名前** を入力して、**Definition** に以下サンプル **iRule** を入力し、*Finished* ボタンを押します。本 iRule では、トラフィックの内容をみて、ICAP プロファイルを紐付けるかの判断もしています。また、iRule 内のホスト名は、環境にあわせた FQDN/IP アドレスに変更して頂く必要があります。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。また、以下の 2 つの iRule は、1 つのファイルにしていただいても構いません。）

    例）トラフィックを分ける用の iRule

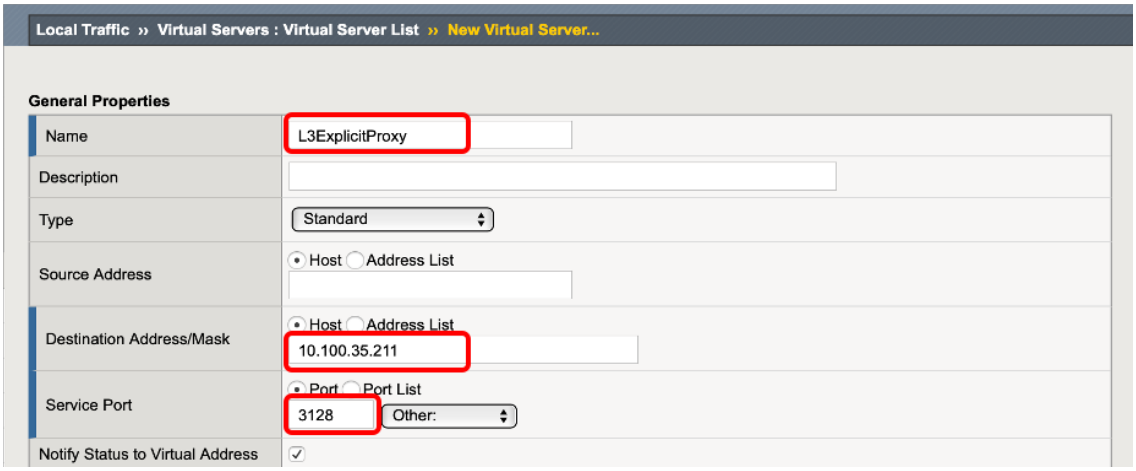
```
###  Add this iRule to Explicit Virtual Server ###
when HTTP_PROXY_REQUEST {
    set F5PROXY "bigip.f5jplab.local"
    if { [HTTP::host] contains $F5PROXY} {
        HTTP::proxy enable
        ADAPT::enable request false
    } else {
        HTTP::proxy disable
        virtual sslo_L3ExplicitProxy.app/sslo_L3ExplicitProxy-xp-4
        snat automap
    }
}
```

    例）URI 書き換え用の iRule

```
###  Add this iRule to Explicit Virtual Server ###
when HTTP_REQUEST {
    set F5PROXY "bigip.f5jplab.local"
    if { [HTTP::host] contains $F5PROXY} {
        set OLDURI [HTTP::uri]
        HTTP::uri "http://[HTTP::host]$OLDURI"
        ADAPT::enable request true
    } else {
        ADAPT::enable request false
    }
}
```

4. 次に Explicit HTTP Proxy 用の Virtual Server を作成します。**Local Traffic >> Virtual Servers** にて、*Create*

ボタンを押します。**任意の名前** を入力し、**Destination Address/Mask** にて、プロキシ接続用の **IP アドレス** を入力、**Service Port** にて、プロキシとして利用する **ポート番号** を入力します。



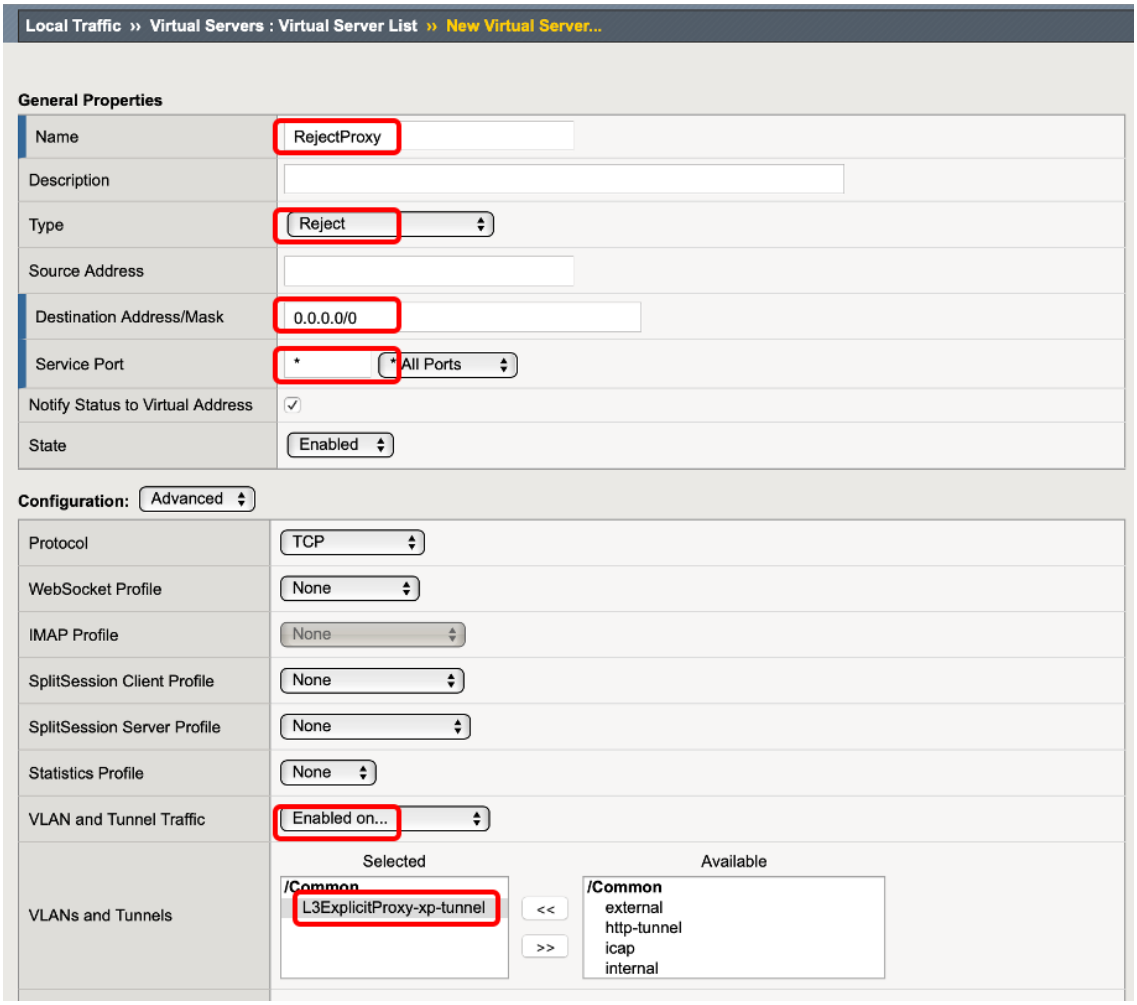
5. **HTTP Profile(Client)** にて、作成済みの **HTTP Explicit Profile** を選択します。

6. **Request Adapt Profile** にて、作成済みのプロファイルを選択し、**Source Address Translation** にて、**Auto Map** を選択します。

7. **iRules** にて作成済みの 2 つの iRule を選択して、 *Finished* ボタンを押します。

Reject 用 Virtual Server の作成

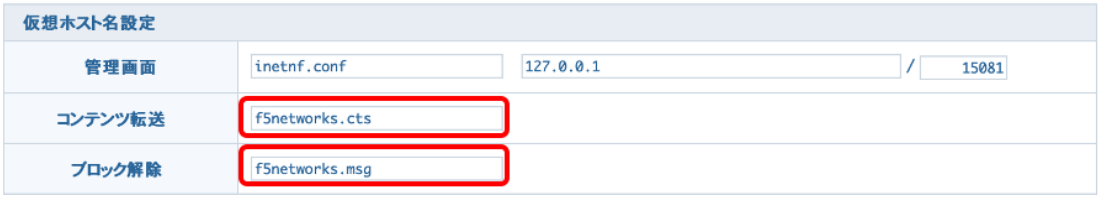
1. 最後に、**15080 ポート以外はブロック** をする Virtual Server を作成します。（本 Virtual Server がなくても動作はします。）**任意の名前** を入力し、**Type** にて **Reject** を選択し、**Destination Address/Mask** にて、**0.0.0.0/0** と入力し、**Service Port** にて * と入力します。**VLAN and Tunnel Traffic** にて、**Enabled on...** を選択し、**VLANs and Tunnels** にて作成済みの Tunnel を選択し、*Finished* ボタンを押します。



i-FILTER 側の設定

1. **システム／システムパラメーター／機能設定** にて、**仮想ホスト転送モード** を有効にし、**コンテンツ転送用アドレス** に **BIG-IP の FQDN** を入力し、**ポート番号** に **15080** を入力します。（IP アドレスを入力しても動作はしますが、SSL 接続時にブラウザでプライバシー保護のエラーが表示されます。）

2. **システム／システムパラメーター／動作設定** にて、**コンテンツ転送**、**ブロック解除** のそれぞれに **任意の値** を設定します。（ブロック画面タイトル表示時、パスワードブロック解除時の URI の一部として利用されます。）

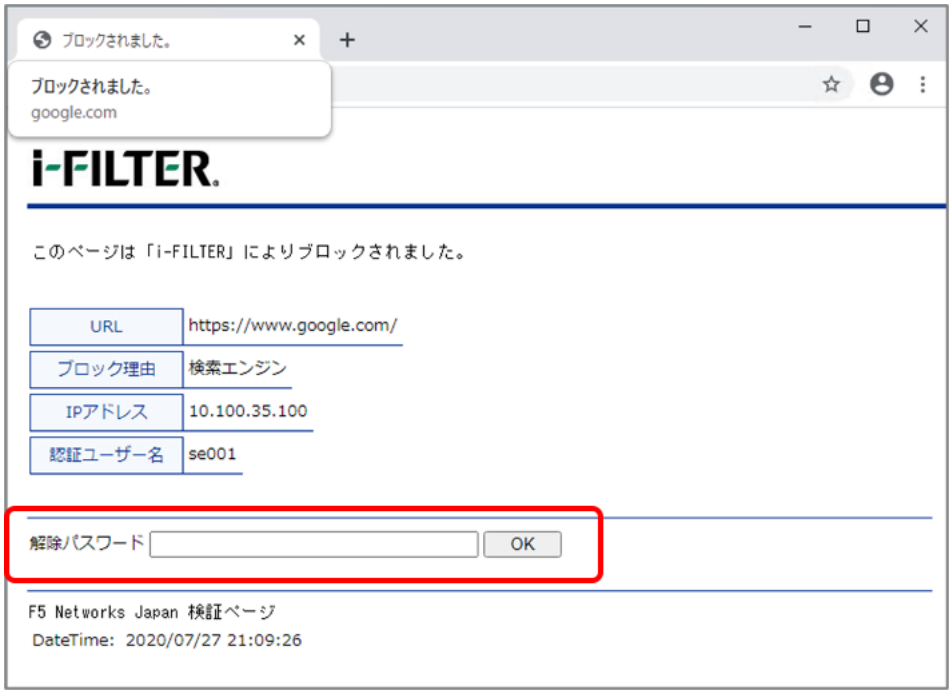


クライアントからの接続テスト

1. ブラウザの **プロキシ設定** にて、作成済みの Explicit Proxy の IP アドレスに紐づく **FQDN** または、IP アドレスに変更します。

2. ブラックリストの宛先への通信がブロックされ、タイトル画像が表示されることを確認します。

3. パスワードブロック解除が設定されている宛先へ接続し、パスワード入力画面が表示されることを確認します。



4. 解除パスワード入力後、無事 WEB 接続ができる事を確認します。

1.1.14 その他

各種ログ

1. **テストの際には、以下の** SSLO **におけるログを参照して下さい。**

   - SSLO **全般ログや** iRule **で** local0 **で指定したログ**　/var/log/ltm
   - APM**（認証）のログ**　/var/log/apm
   - SSL Guided Configuration**（**iAppLX/REST**）に関わるログ**　/var/log/restnoded/restnoded.log
     /var/log/restjava.0.log
   - **その他、**SSLO **に関するトラブルシューティングは以下の** AskF5 **記事をご参照下さい。**
     K26520133:　Troubleshooting　SSL　Orchestrator(SSLO)　https://support.f5.com/csp/article/K26520133

2. **テストの際には、以下の** i-FILTER **におけるログを参照して下さい。**

    - **アクセスログ**　/usr/local/ifilter10/logs/access.log0000

フィルター除外設定（必須ではございません）

1. 本テストでは、httpbin.org を接続先ホストとして利用させて頂いておりますが、現状デフォルトではブロックされるので、**システム／システムフィルター／推奨フィルター設定 – 除外設定** にて、テストで利用しているホスト名を **除外設定** に追加します。

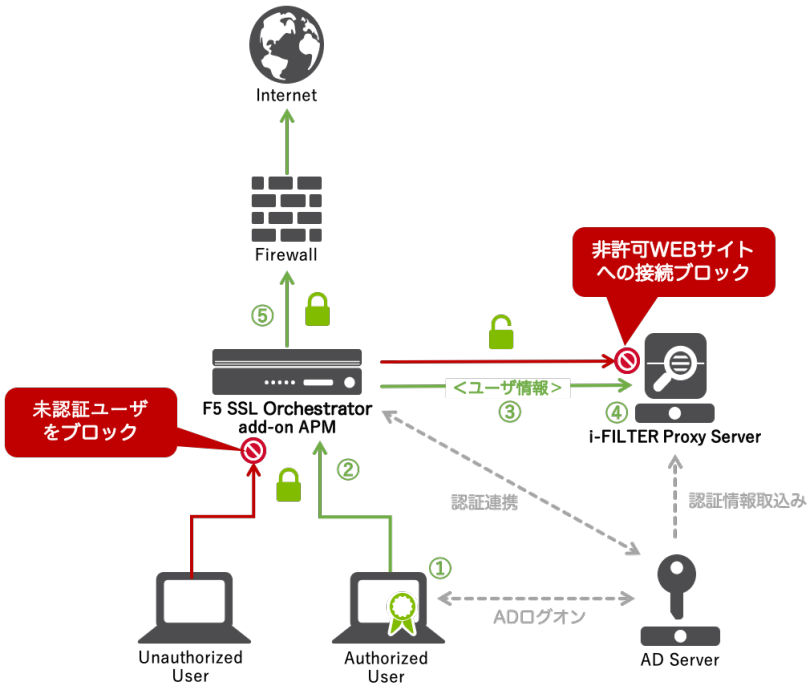
1.2 F5 SSLO (L3 Explicit Proxy) と i-FILTER Proxy 版の連携設定

本章では、SSLO の可視化ゾーンに i-FILTER Proxy 版を配置して連携する構成について、ご紹介致します。

1.2.1 F5 SSLO と i-FILTER PROXY 版連携の流れ

F5 SSLO と i-FILTER PROXY 版の連携の流れは以下の通りです。

1. クライアントがパソコン (AD) にログオンします。



2. SSLO でユーザ認証を実施し (NTLM 認証、Kerberos 認証の場合は自動的に認証が実施されます)、SSL 通信を復号します。
3. WEB 接続の情報と SSLO で認証が成功したユーザ情報を i-FILTER に HTTP で送ります。
4. ユーザ情報を元に URL フィルタリングルールを適用し、接続が許可されていない WEB サイトの場合は、ブロック画面をクライアントに表示します。
5. 許可された通信に関しては、SSL 通信を再暗号化し、WEB サーバへ接続します。

1.2.2 本ガイドの利用バージョンと構成イメージ

本ガイドは、以下の製品バージョンを利用して、作成しております。

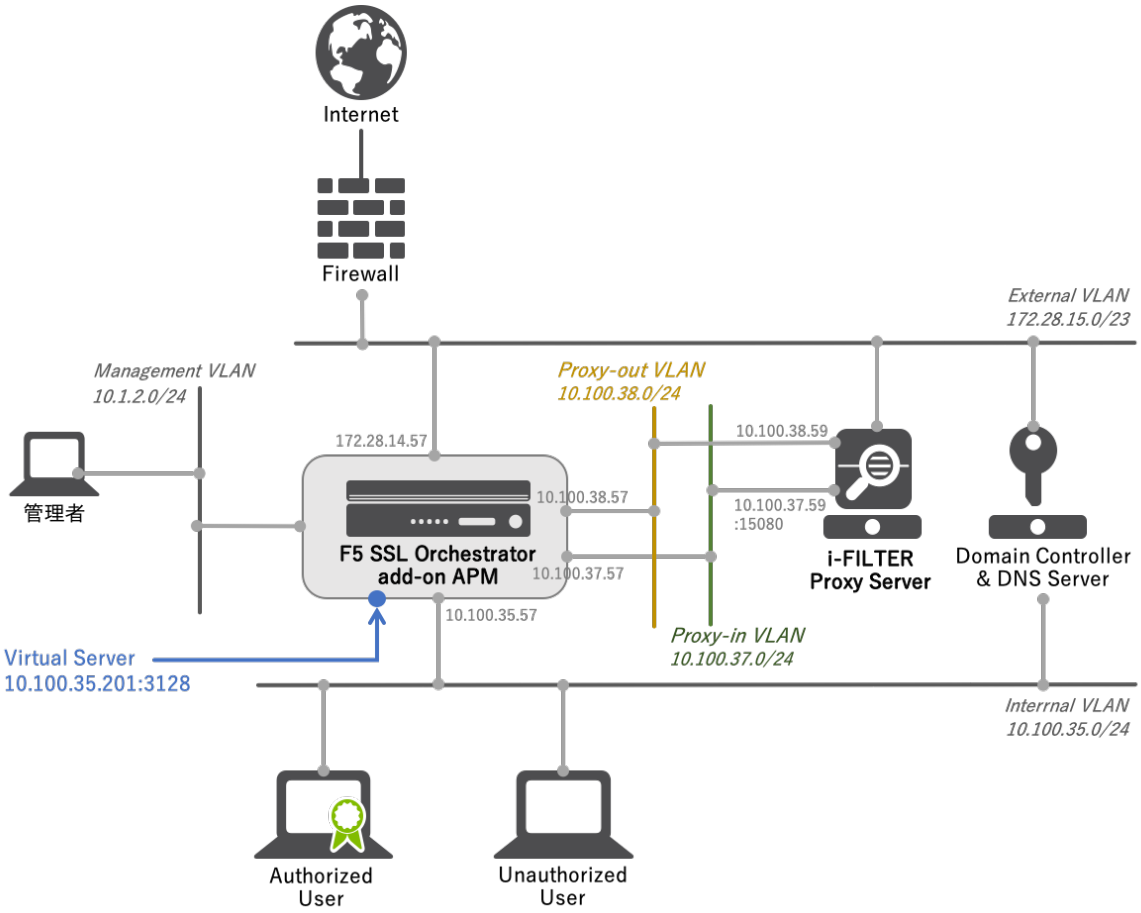
1. 利用バージョン

製品名	バージョン
F5 SSL Orchestrator	v15.1.0.5-0.0.8 âĂŠ 7.4.9
i-FILTER Proxy Server (Linux)	Ver.10.41R01-20200807 (CentOS v7.7.1908 ※) ※ デジタルアーツ様による製品サポート OS は Red Hat Enterprise Linux となります。

注釈:

- （各 F5 代理店でサポート可能な範囲において、）極力最新のバージョンを適用頂くことをおすすめ致します。最新のバージョンは AskF5 でご確認下さい。
- Proxy 認証を行いたい場合、i-FILTER のログにユーザ名を出力したい場合は、APM のライセンスが必要となります。

2. 本ガイドにおける構成イメージ



1.2.3 ライセンスアクティベーション、プロビジョニング、CA 証明書/鍵登録

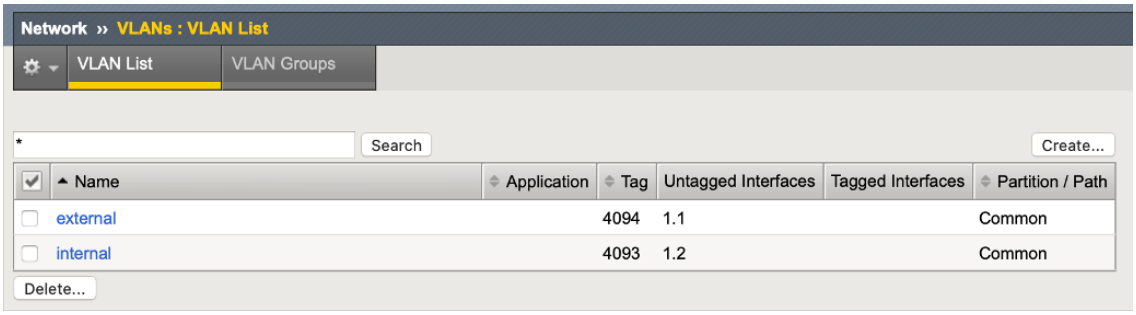
1- 1.3 と同じです。

1.2.4 最新版の SSL Orchestrator RPM へのアップグレード

1- 1. 4 と同じです。

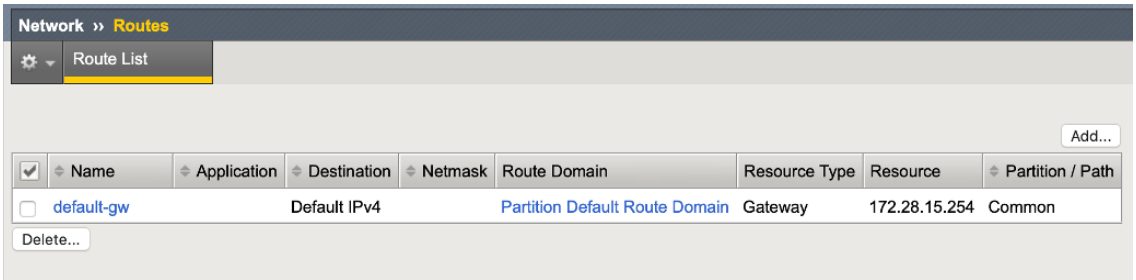
1.2.5 Network の基本設定

1. VLAN の設定を行います。



2. Self IP の設定を行います。

3. デフォルトゲートウェイの設定を行います。



1.2.6 DNS, NTP の設定

1- 1.6 と同じです。

1.2.7 i-FILTER にて HTTP/HTTPS 判別するための iRule の作成

i-FILTER PROXY 版は、HTTP のリクエストヘッダで HTTP サーバへの通信か HTTPS サーバへの通信かを判別しています。i-FILTER が HTTP/HTTPS 判別可能となるための iRule を作成します。**X-Forwarded-Proto** というヘッダーに利用プロトコル（http or https）を加えて、HTTP リクエストに挿入します。ここではあわせて、クライアントの IP アドレスも **X-Forwarded-For** に挿入しています。

1. **Local Traffic >> iRules** にて、*Create* ボタンを押します。**任意の名前** を入力して、**Definition** に以下サンプル iRule を入力し、*Finished* ボタンを押します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。）

   例）HTTP リクエストに i-FILTER 連携のヘッダを挿入するための iRule サンプル

```
###  Add this iRule to Virtual Server:ssloS_XXXX-t-4 ###

when HTTP_REQUEST {

    if { [TCP::local_port] == 80 } {
        # For HTTP protocol
        HTTP::header replace "X-Forwarded-Proto" "http"

    } elseif { [TCP::local_port] == 443 } {
        # For HTTPS protocl
        HTTP::header replace "X-Forwarded-Proto" "https"
    }

    sharedvar ctx
    # For Client IP address
    lappend ctx(headers) "X-Forwarded-For" [IP::client_addr]

    if { [expr { [llength $ctx(headers)] % 2 }] == 0 } {
        foreach {h_name h_value} $ctx(headers) {
            HTTP::header replace ${h_name} ${h_value}
        }
    }
}
```

   接続テストをされる際には、変数情報をログ出力するなどして意図する値が入っているか確かめて頂くことをおすすめします。

例１）ヘッダー内容を確認するデバッグ

```
foreach attr "[HTTP::header names]" {
        log local0. "$attr : [HTTP::header value $attr]"
}
```

1.2.8 NTLM 認証設定

1- 1.8 と同じです。

1.2.9 認証ヘッダ挿入用 iRule の作成

ここでは、ユーザ認証ヘッダ（Proxy-Authorization）と APM で認証されたユーザ名（session.logon.last.username）を HTTP リクエストに挿入するための iRule を作成します。i-FILTER PROXY 版連携では、**Proxy-Authorization** というヘッダー名に、ユーザ名を Base64 エンコーディングした Basic 認証の形式で挿入します。

1. **Local Traffic >> iRules** にて、2- 2.7 で作成済みの iRule にユーザ名用のヘッダーを挿入するための箇所を追記します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。）

   例）HTTP リクエストに i-FILTER 連携のヘッダを挿入するための iRule サンプル

```
###  Add this iRule to Virtual Server:sslos_XXXX-t-4 ###

when HTTP_REQUEST {

    if { [TCP::local_port] == 80 } {
        # For HTTP protocol
        HTTP::header replace "X-Forwarded-Proto" "http"

    } elseif { [TCP::local_port] == 443 } {
        # For HTTPS protocl
        HTTP::header replace "X-Forwarded-Proto" "https"
    }

    sharedvar ctx
    # For Client IP address
    lappend ctx(headers) "X-Forwarded-For" [IP::client_addr]

    # For Username
    if { [ACCESS::session data get session.logon.last.username] ne "" } {
        set PROXYAUTH "Basic [b64encode [ACCESS::session data get session.
↪logon.last.username]:]"
```

(次のページに続く)

(前のページからの続き)

```
    } else {
        set PROXYAUTH ""
    }
    lappend ctx(headers) "Proxy-Authorization" $PROXYAUTH

    if { [expr { [llength $ctx(headers)] % 2 }] == 0 } {
        foreach {h_name h_value} $ctx(headers) {
        HTTP::header replace ${h_name} ${h_value}

    }
}
```

接続テストをされる際には、変数情報をログ出力するなどして意図する値が入っているか確かめて頂くことをおすすめします。

例１）認証ユーザ名を確認するデバッグ

log local0. "Username: [ACCESS::session data get session.logon.last.username]"

例２）エンコードされた内容を確認するデバッグ

log local0. "Encoded username: b64encode [ACCESS::session data get session.logon.last.username]"

例３）ヘッダー内容を確認するデバッグ

foreach attr "[HTTP::header names]" {

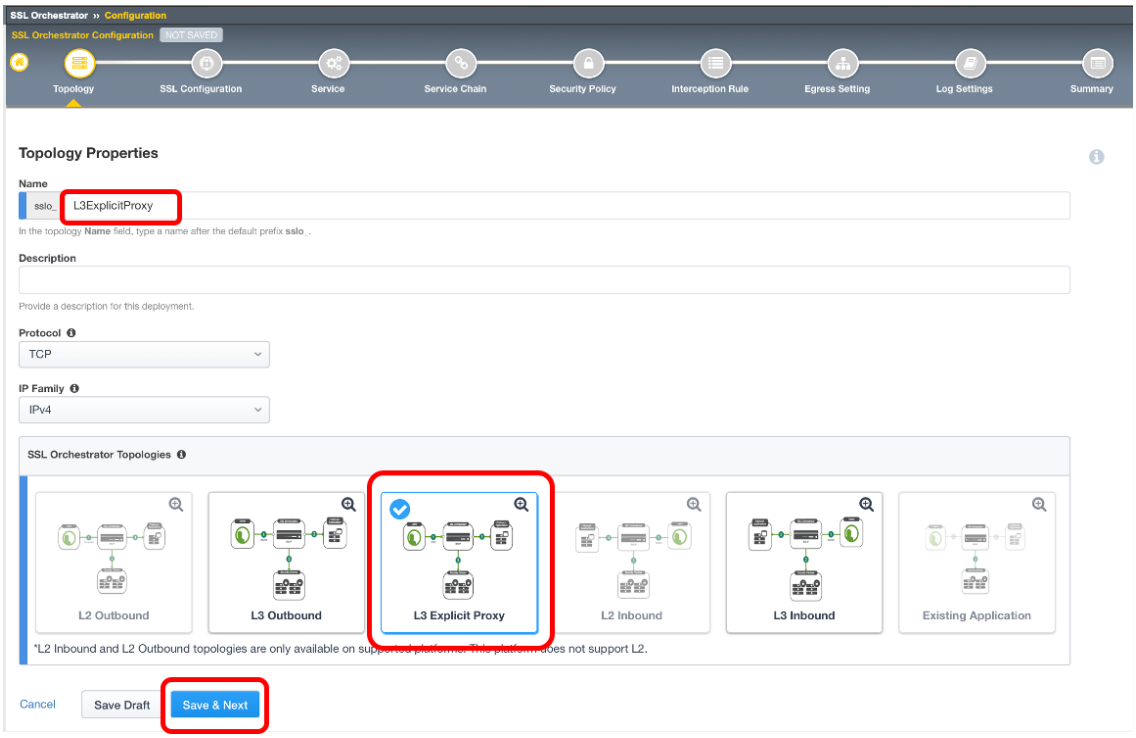
　　log local0. "$attr : [HTTP::header value $attr]"

}

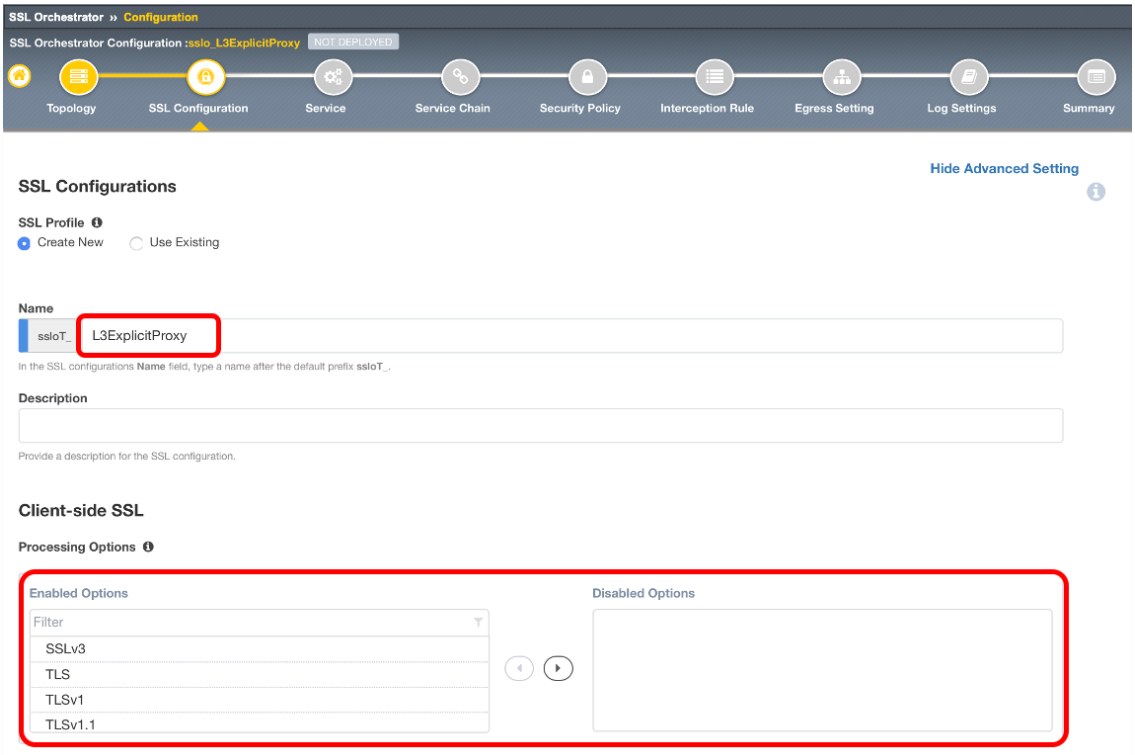
1.2.10 SSLO Guided Configuration による SSLO の設定

1. **SSL Orchestrator >> Configuration** を選択します。**DNS** と **NTP** と **Route** が **Configure** となっているのを確認し、*Next* ボタンを押します。

2. **任意の名前** を設定し、SSL Orchestrator Topologies として、**L3 Explicit Proxy** を選択し、*Save* **&** *Next* ボタンを押します。

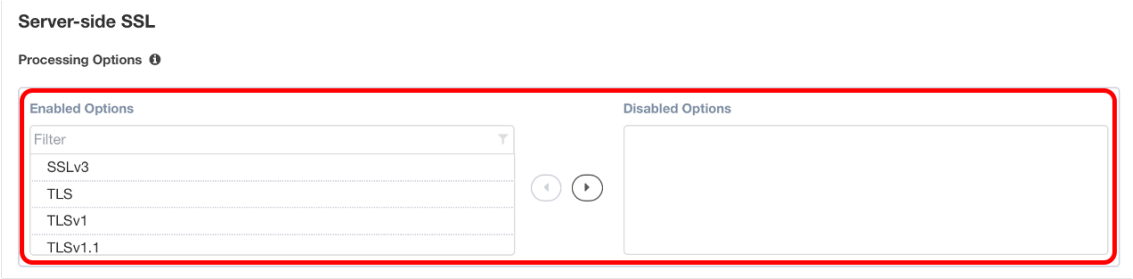


3. **任意の名前** を設定し、右上の *Show Advanced Setting* をクリックし、**Client-side SSL** にて、利用したい **TLS のバージョン** を選択します。

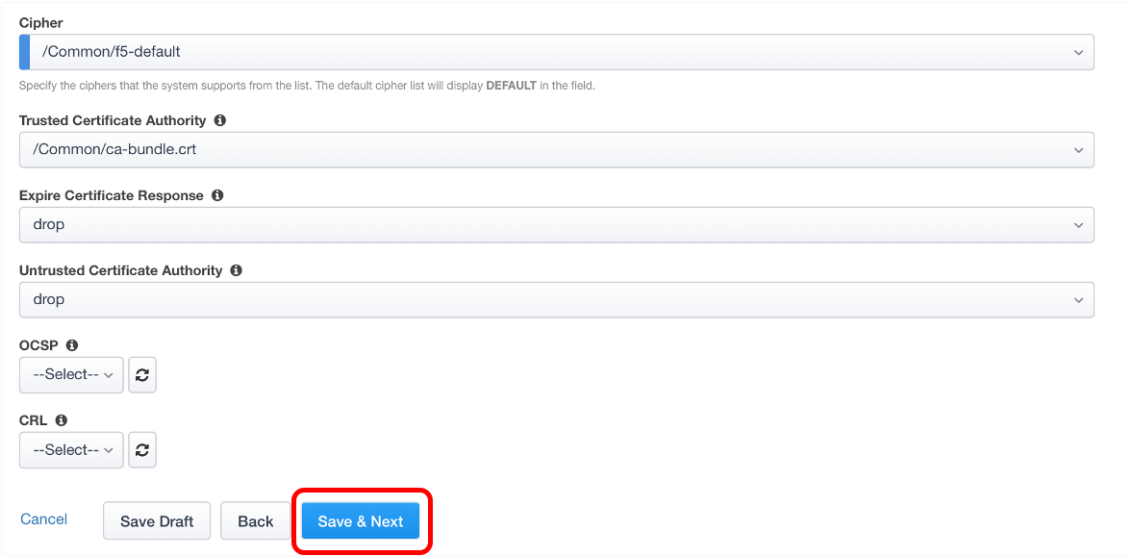


4. **CA Certificate KeyChain** にて、既にインポート済みの CA ファイルを選択して *Done* を押します。

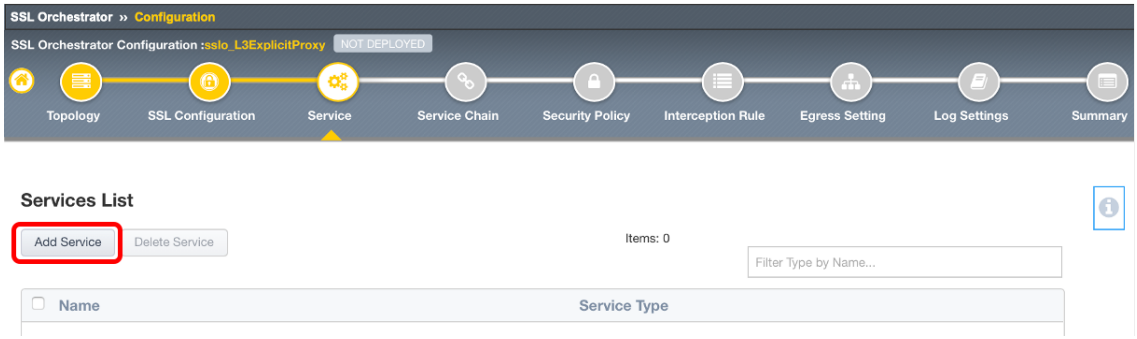
5. **Server-side SSL** も同様に利用したい **TLS バージョン** を選択します。



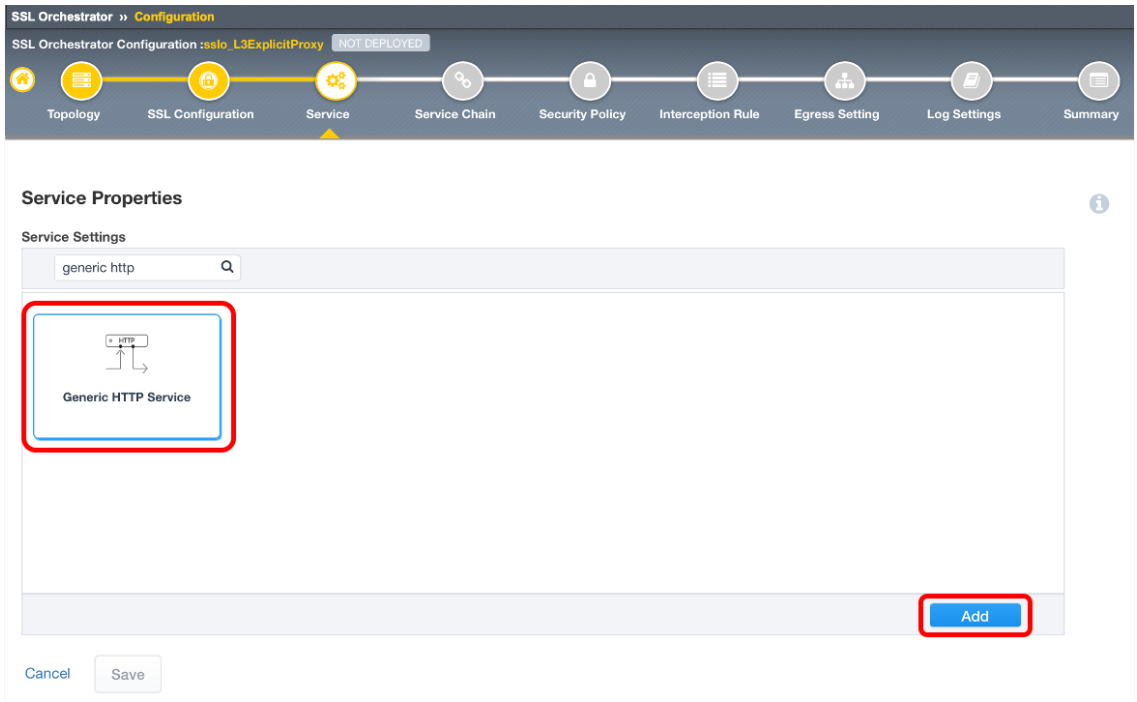
6. 期限切れの証明書や自己署名証明書に対しての動作も確認し、*Save* **&** *Next* を押します。



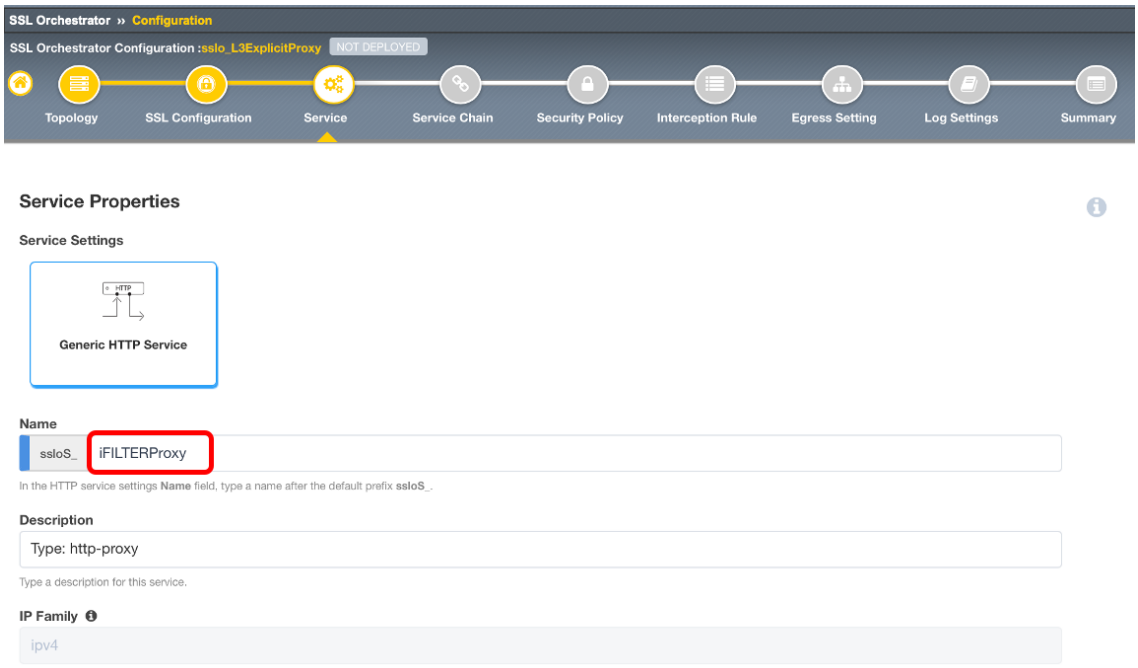
7. *Add Service* を押します。



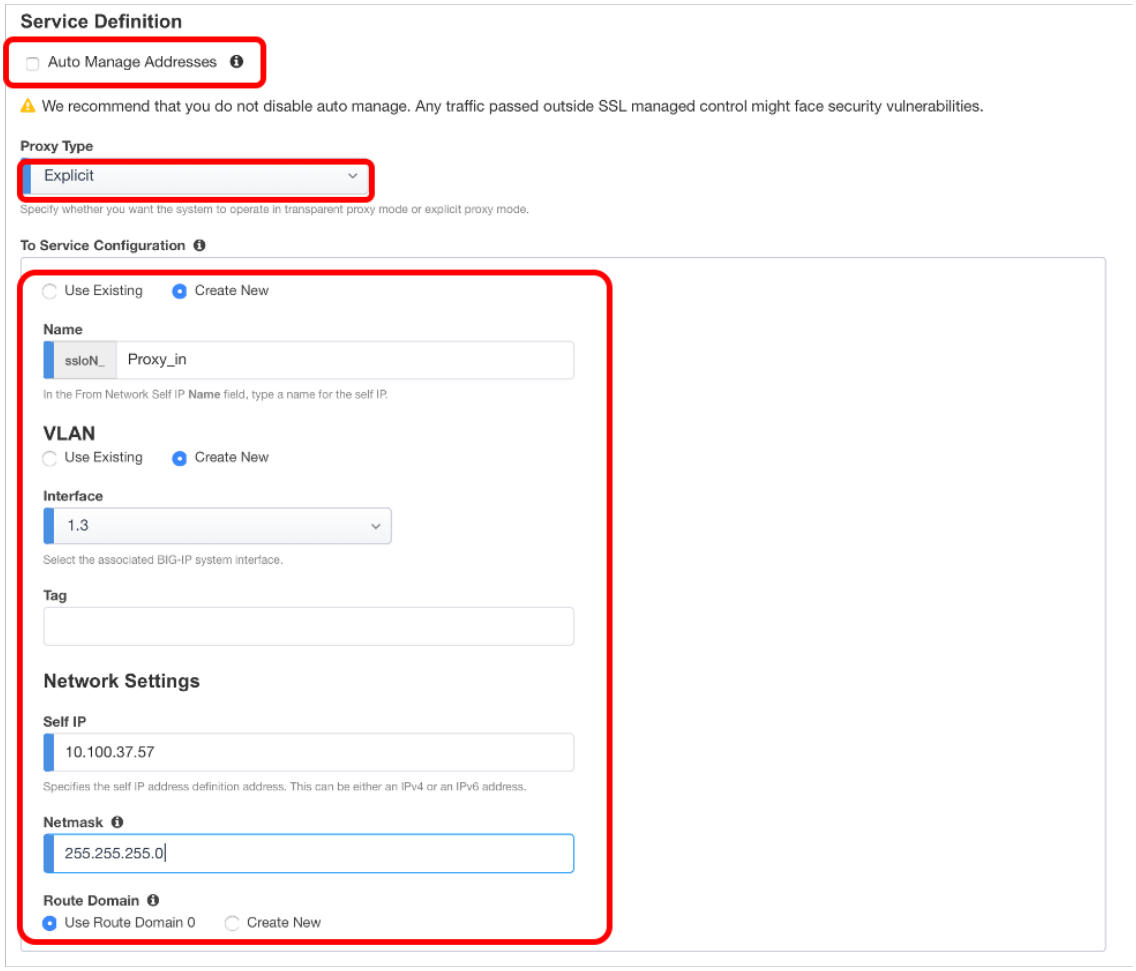
8. **Generic HTTP Service** を選択し、*Add* ボタンを押します。



9. **任意の名前** を設定します。



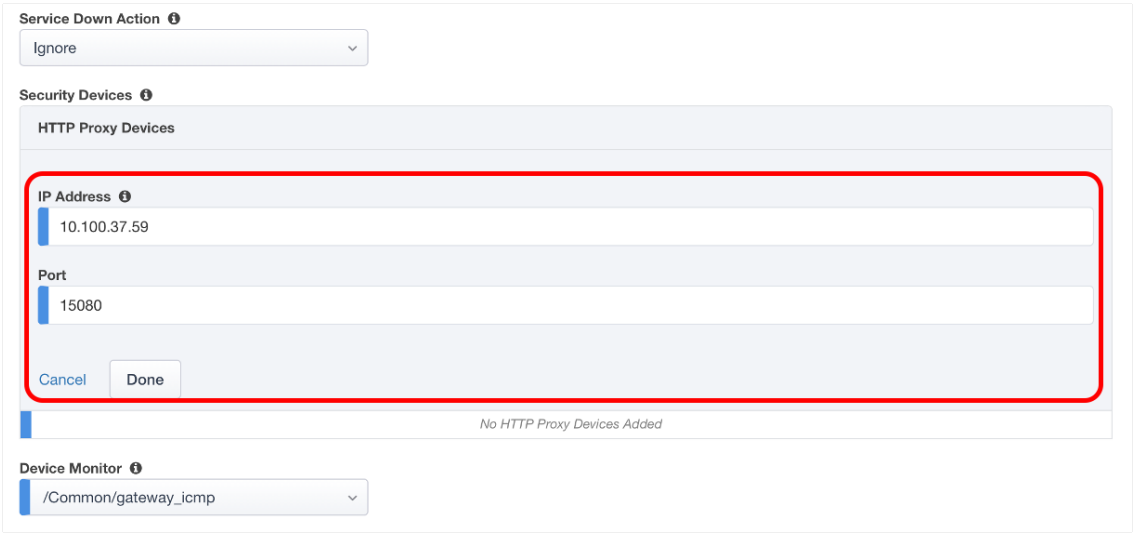
10. **Service Definition** にて、**Auto Manage Addresses** のチェックをはずし、SSLO から i-FILTER Proxy へ HTTP トラフィックを転送する VLAN, Self IP の情報を入力します。（ここでは、検証環境の都合で Auto Manage Addresses は利用しておりません。）



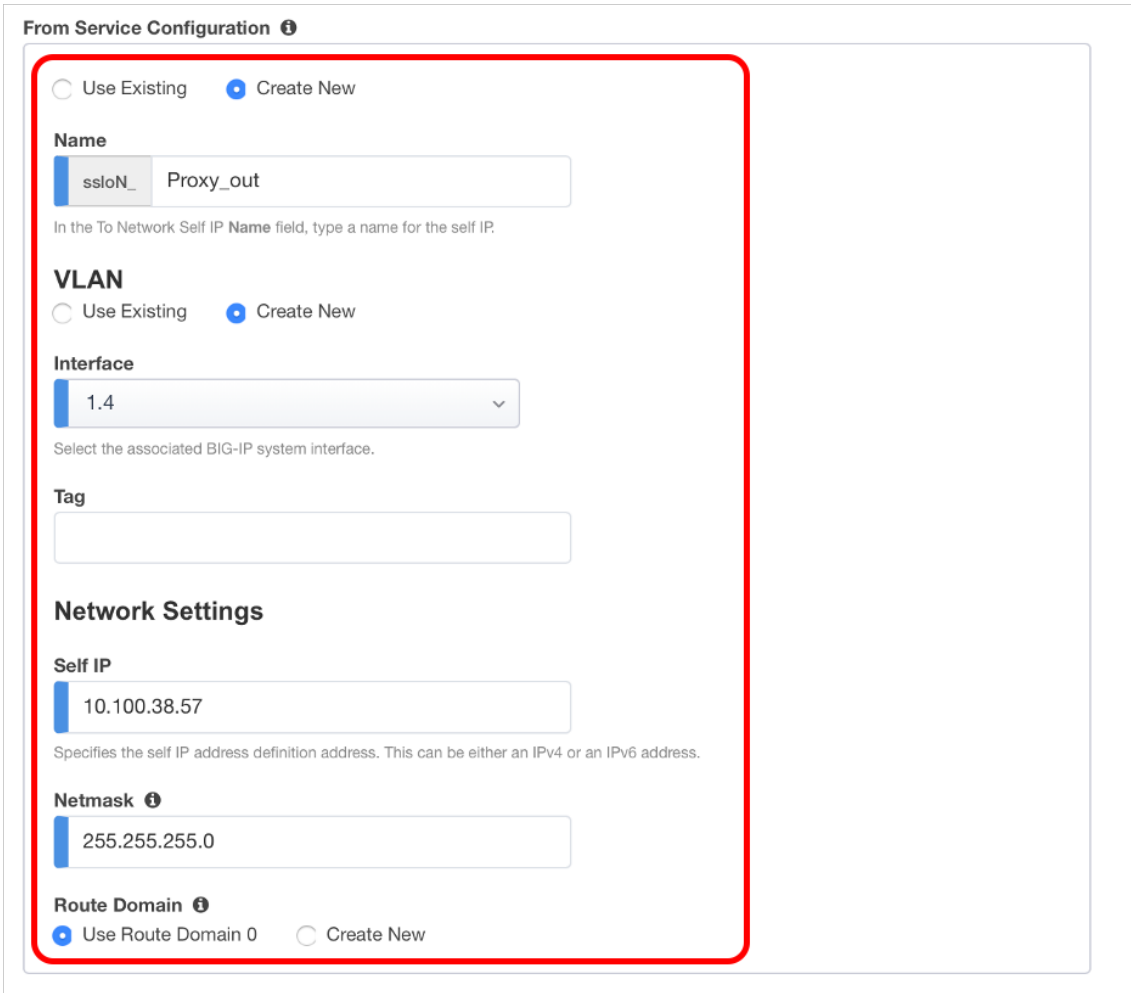
注釈:

- SSL 可視化ゾーン（ここでは i-FILTER Proxy 版）では、HTTP トラフィックは暗号化されておりませんので、第三者からのアクセスを避ける必要があります。そこで、SSLO では、Auto Manage Addresses（RFC2544 に定められているインターナル利用アドレス）をセキュリティ機器に利用頂くことを推奨しております。

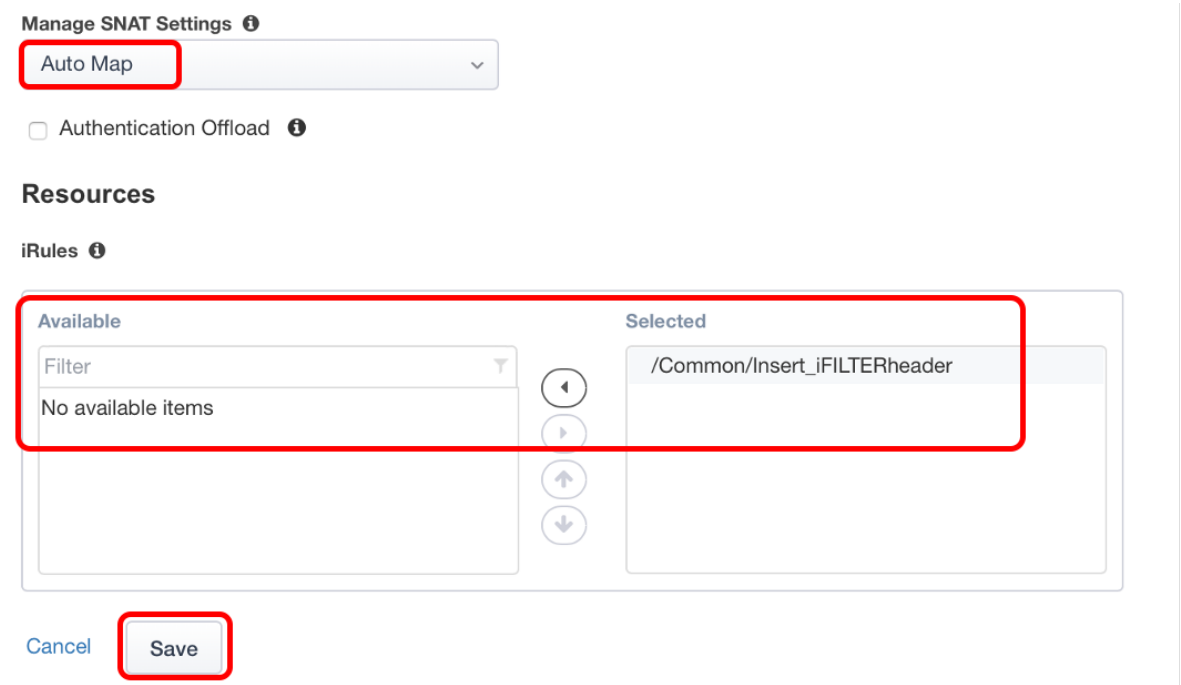
11. i-FILTER Proxy のインライン側の **IP アドレス** と **ポート番号** を入力し、*Done* を押します。



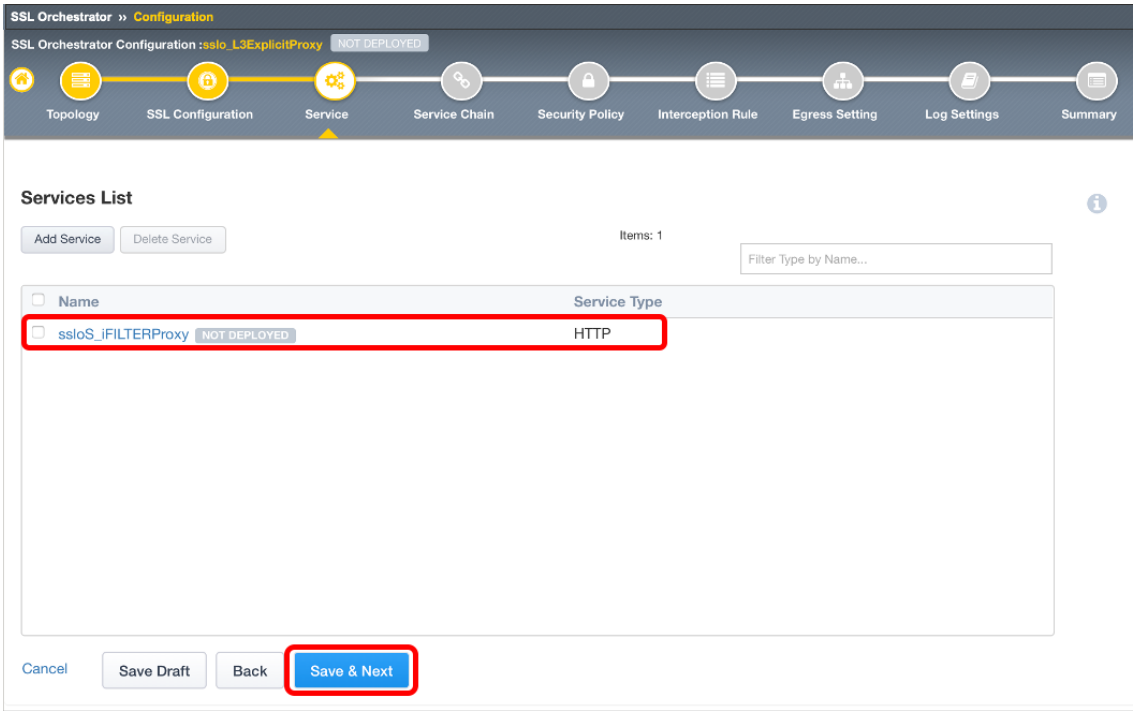
12. i-FILTER Proxy から SSLO へ HTTP トラフィックを転送する VLAN, Self IP の情報を入力します。



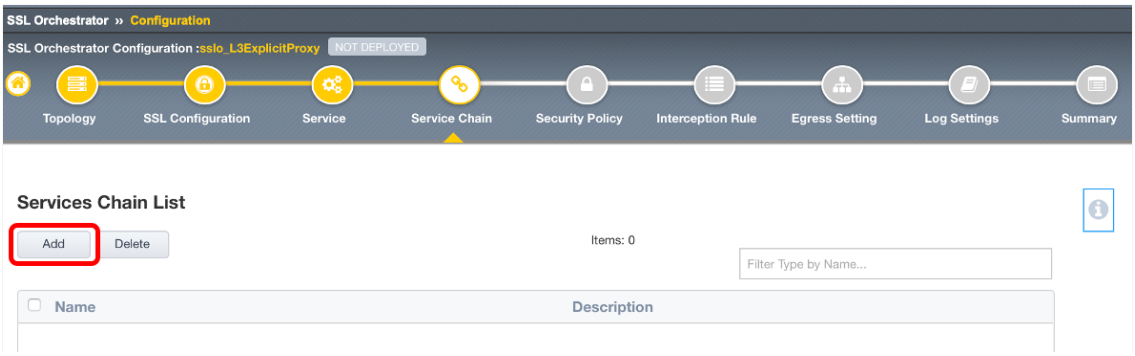
13. 検証環境では **Auto Map** を設定し、作成済みの iRule を右に移動させ、*Save* を押します。



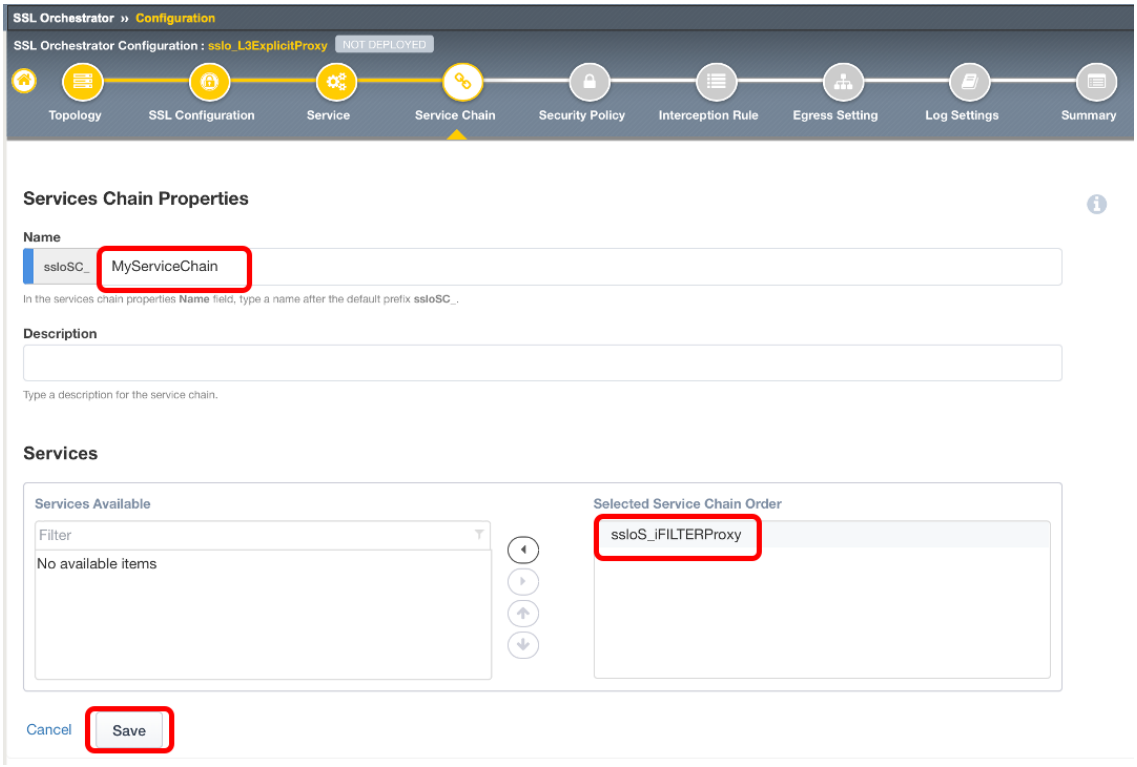
14. Service が追加されるとこのような画面になります。*Save* **&** *Next* を押します。



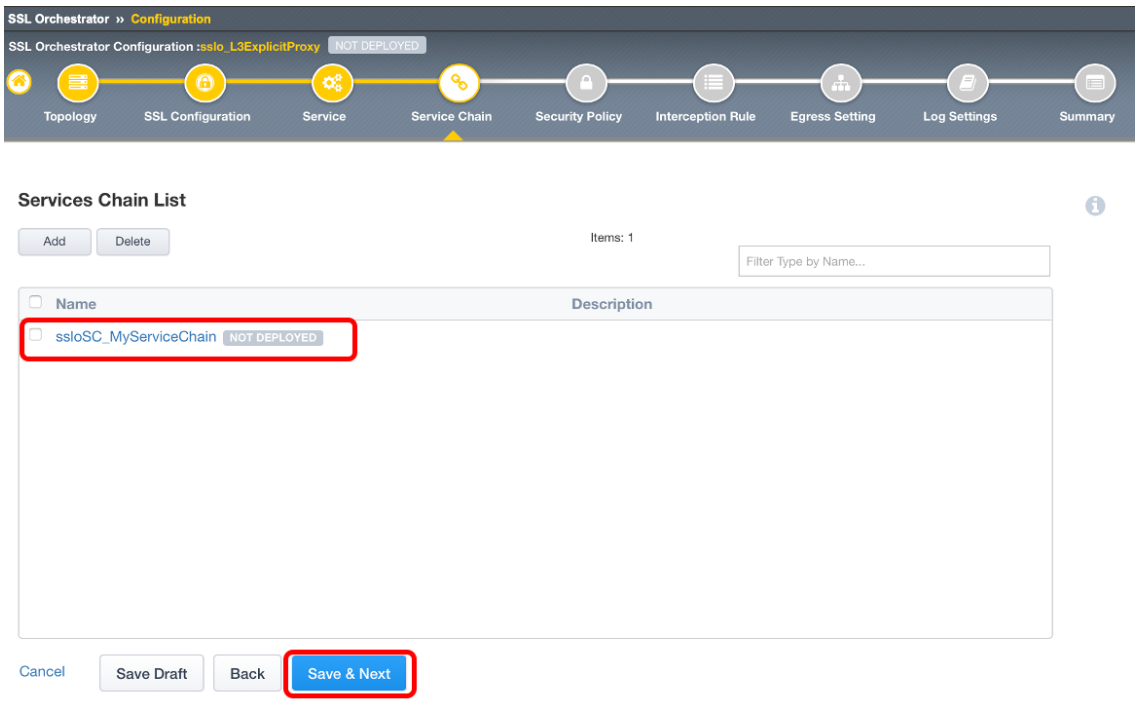
15. *Add* をクリックし、サービスチェーンを作成します。



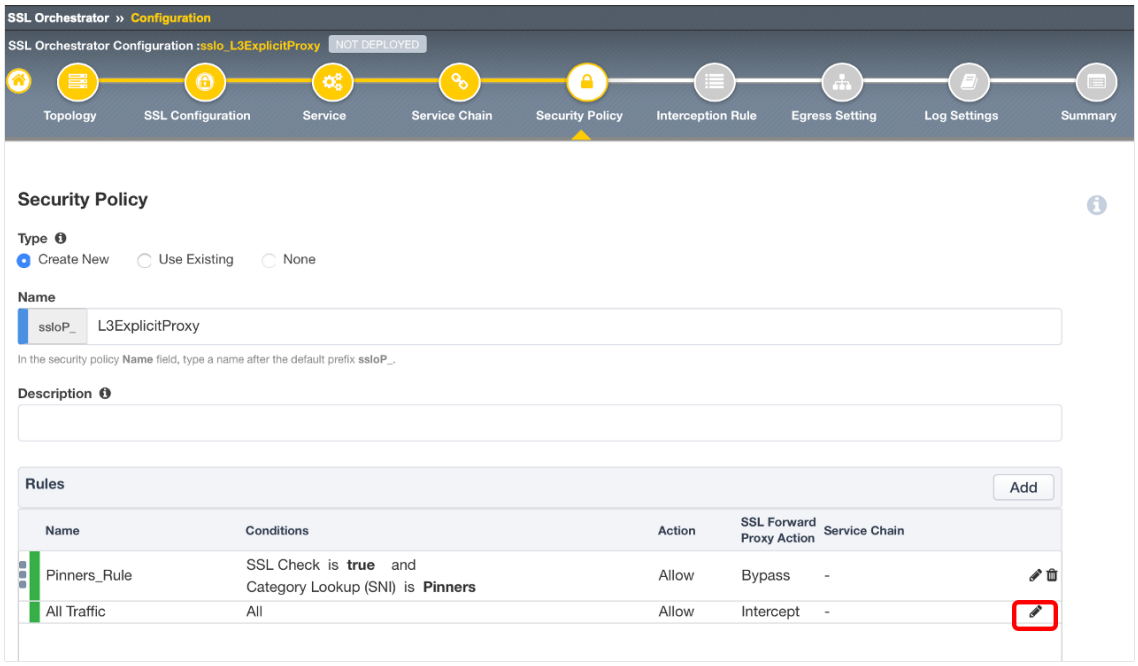
16. **任意の名前** を設定し、作成済みの Service を右に移動させ、*Save* ボタンを押します。



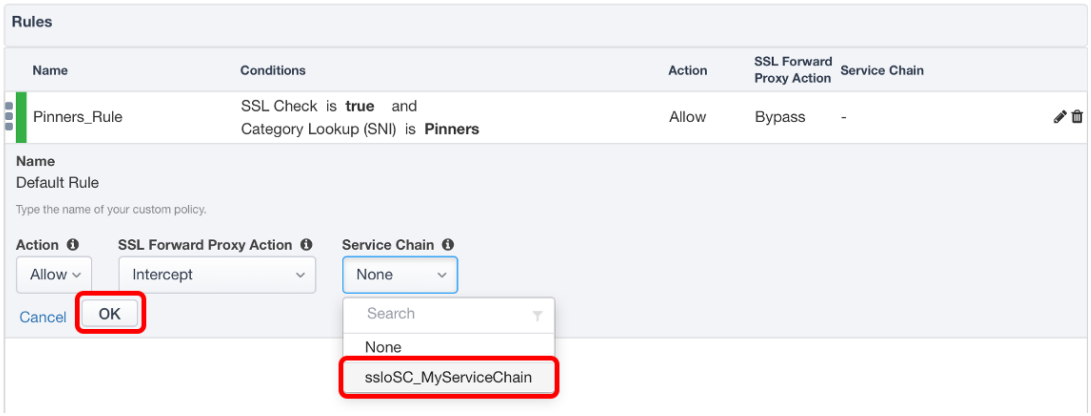
17. サービスチェーンが作成されると、このような画面になります。*Save* **&** *Next* ボタンを押します。



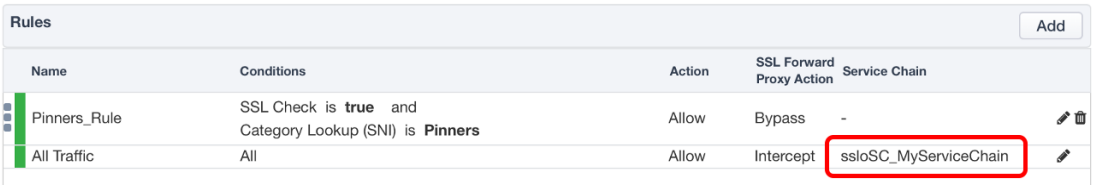
18. All Traffice の **ペンマーク** をクリックします。



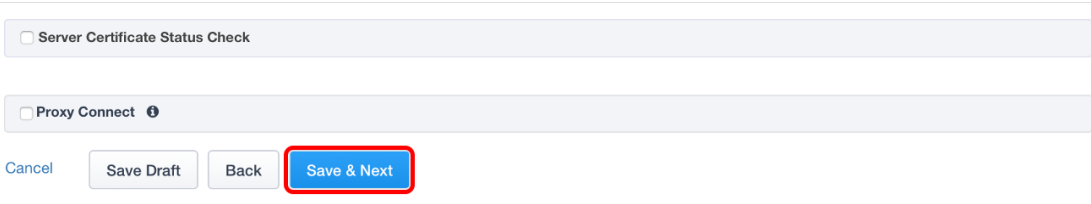
19. 先程作成した Service Chain を選択し、*OK* ボタンを押します。



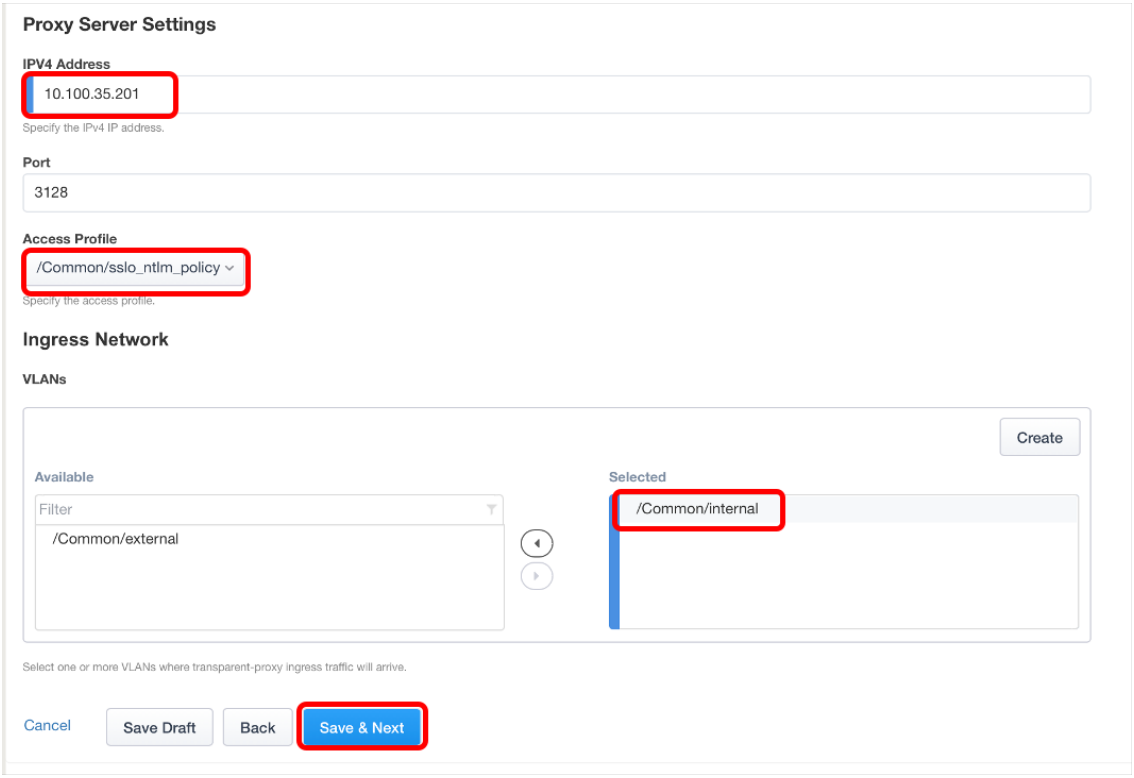
20. All Traffice に Service Chain が追加されると以下のような画面になります。



21. *Save & Next* ボタンを押します。

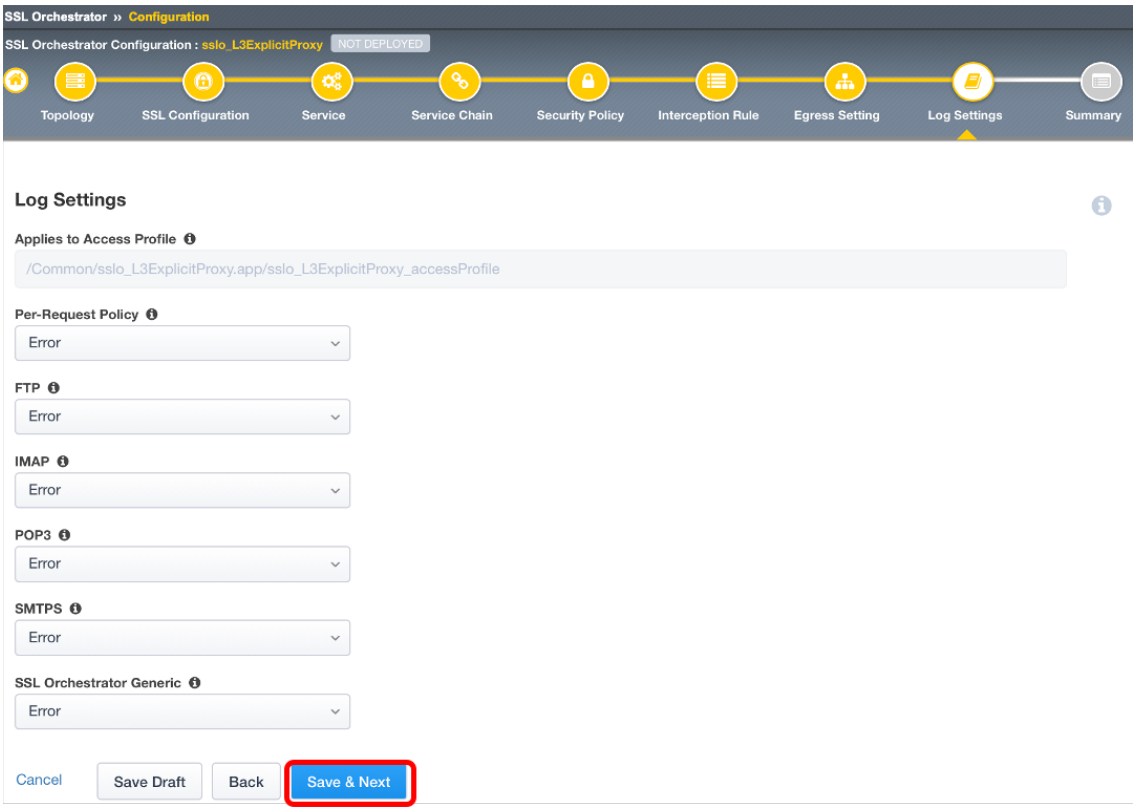


22. Proxy Server Settings にクライアントからプロキシとしてアクセスさせる IP アドレスを入力し、既に作成済みの Access Profile を選択し（プロキシ認証しない場合は不要）、Ingress Network として、クライアントからアクセス可能な VLAN を選択し、*Save* **&** *Next* ボタンを押します。

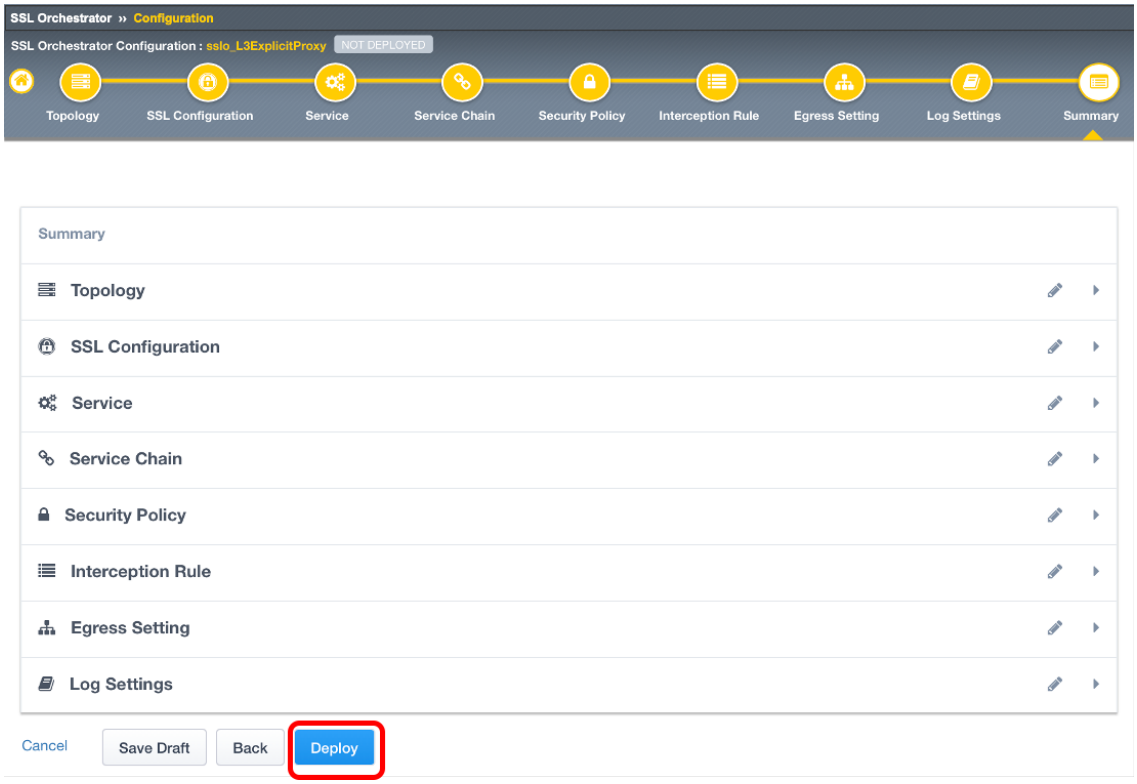


23. 本テスト構成では、Manage SNAT Settings で Auto Map、Gateways で Default Route を選択し、*Save* **&** *Next* ボタンを押します。(設定は検証環境に合わせてください。)

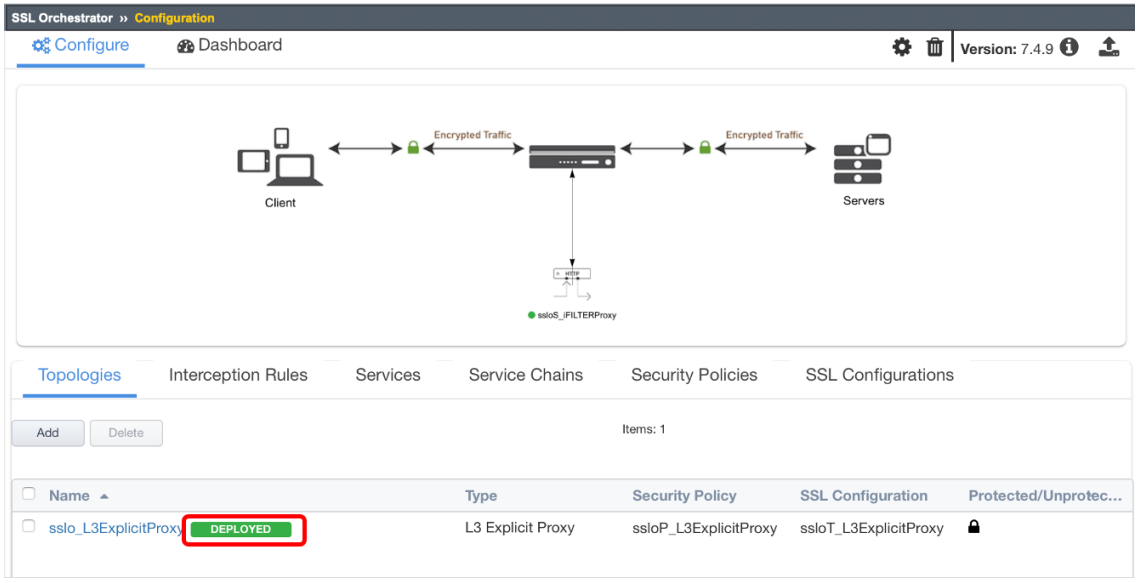
24. *Save* **&** *Next* ボタンを押します。



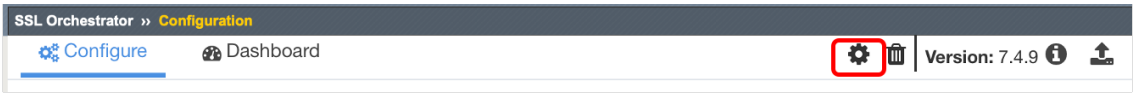
25. 必要に応じて、設定内容を見直し、*Save* **&** *Next* ボタンを押します。



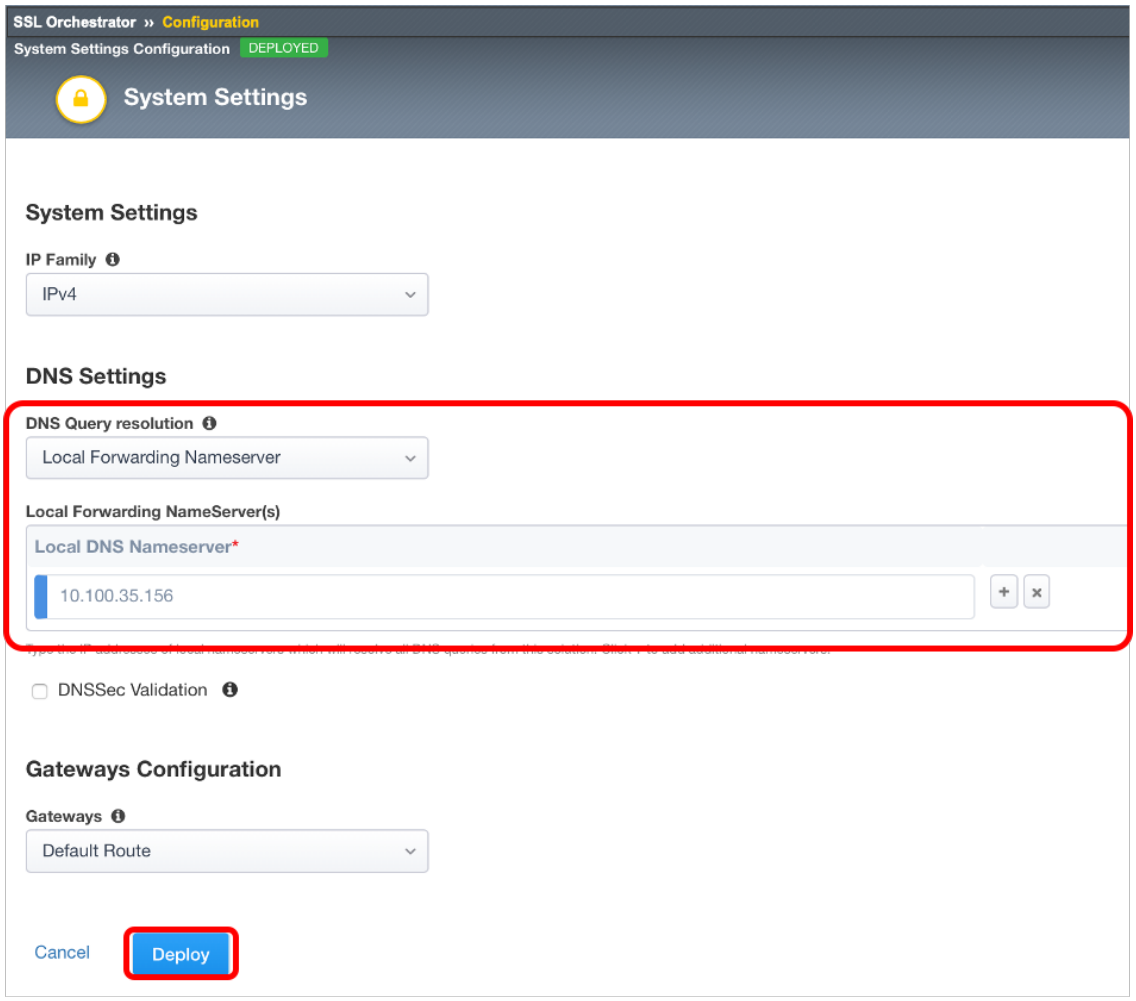
26. *OK* ボタンを押し、Deploy に成功すると以下のような緑色の **DEPLOYED** マークが表示されます。



27. 右上の **System Settings** アイコンを選択します。



28. SSLO が Explicit Proxy として利用する **DNS** を設定し、*Deploy* を押します。



注釈:

- セキュリティデバイスが ICAP サービス、HTTP サービスの場合、SSL 復号していないトラフィックをサービスチェーンに流せません。

1.2.11 i-FILTER Proxy 版（Linux 版）の設定

i-FILTER のインストールは終了しているものとします。

注釈:

- SSL 可視化ゾーン（ここでは i-FILTER Proxy 版）では、HTTP トラフィックは暗号化されておりませんので、第三者からのアクセスを避ける必要があります。そこで、SSLO では、Auto Manage Addresses（RFC2544 に定められているインターナル用アドレス、198.19.x.y/19）をセキュリティ機器に利用頂くことを推奨しております。
- 上記推奨設定を採用できる場合、i-FILTER 用のサーバ構築時にこのアドレス帯を割り当てて下さい。

SSLO 連携の設定

1. conf/ifilter.conf に以下のパラメータを追加します。

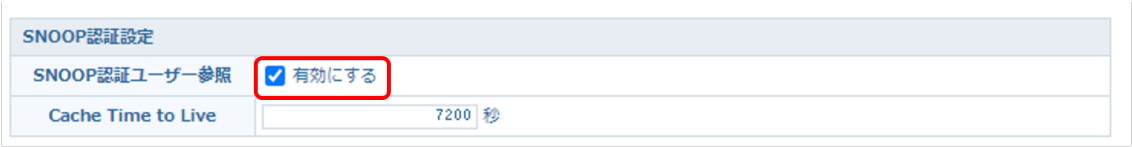
```
proxy_header_rewrite_protocol_enable = on
```

ユーザ認証の設定

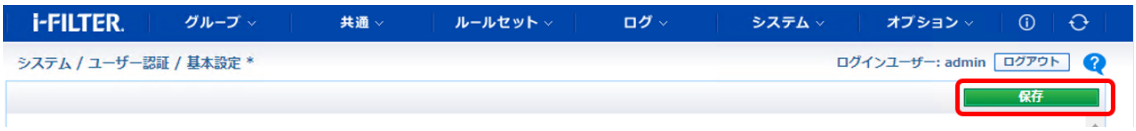
1. **システム／ユーザ認証／基本設定** を選択します。

2. **基本設定** の **ユーザ認証方式** において **SNOOP 認証** を選択します。

3. **ICAP 認証設定** の **SNOOP 認証ユーザ参照** において、**有効にする** をチェックします。



4. 右上の **保存** ボタンを押します。



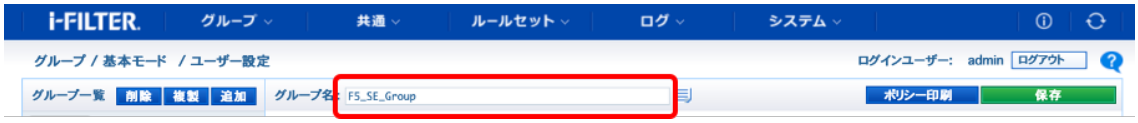
5. 次に、**システム／ユーザ認証／ LDAP サーバ設定** を選択します。本ガイドでは認証ユーザ情報を AD サーバから取り込むため、以下のように設定し、**保存** ボタンを押します。

項目	設定例
アドレス:ポート番号	ご利用の AD 情報を登録
BIND	BIND 有効
BIND DN	ご利用の管理者 BIND DN 情報を登録 l 設定例：cn=Administrator,cn=Users,dc=f5jplab,dc=local
BIND パスワード	ご利用のドメイン管理者のパスワードを登録
Search Base	ご利用のドメイン情報を登録 l 設定例：dc=f5jplab,dc=local
Search Filter	(sAMAccountName=%1)

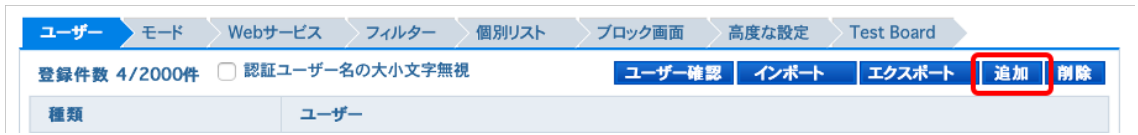
グループポリシーの作成、ユーザ情報の取り込み

1. **グループ／基本モード／モード設定** において、**追加** ボタンを押します。

2. **グループ名** を分かりやすいグループ名に変更します。



3. **ユーザ** タブにおいて、**追加** ボタンをクリックします。



4. **認証ユーザ参照** を選択します。

5. 検索しやすい検索条件を設定し、AD から取り込みたい **ユーザ名** を追加します。

6. **Web サービス** タブを選択し、制御したいサービスを選択し、ユーザに利用不許可とする機能に対し、**ブロック** を選択します。（注：下記イメージはあくまでも一例です。）

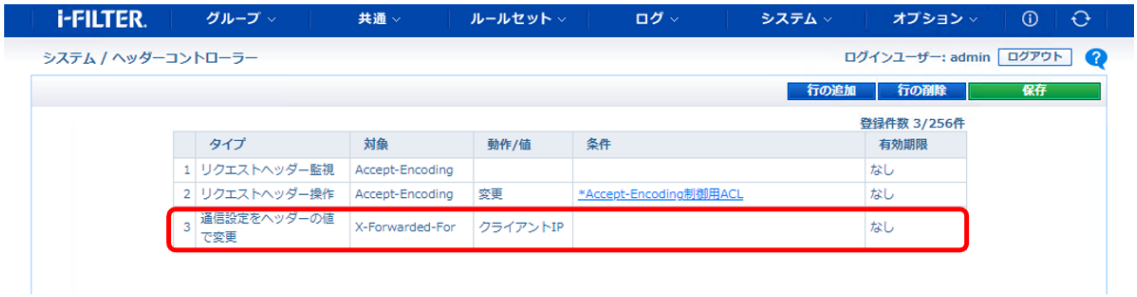
7. **個別リスト** タブを選択し、個別ブロックしたい URL/URI を追加し、右上の **保存** ボタンを押します。（注：下記イメージはあくまでも検証目的で設定した内容です。）

ヘッダーコントローラの設定

1. iRule で挿入した X-Forwarded-For（クライアント IP アドレス）を i-FILTER ログに出力するための設定を行います。

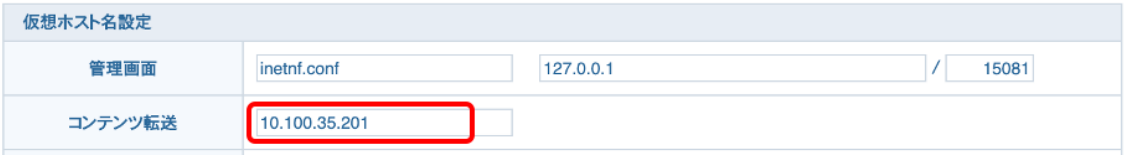
2. ／**システム**／**ヘッダーコントローラ** を選択します。

3. 以下のように、設定します。



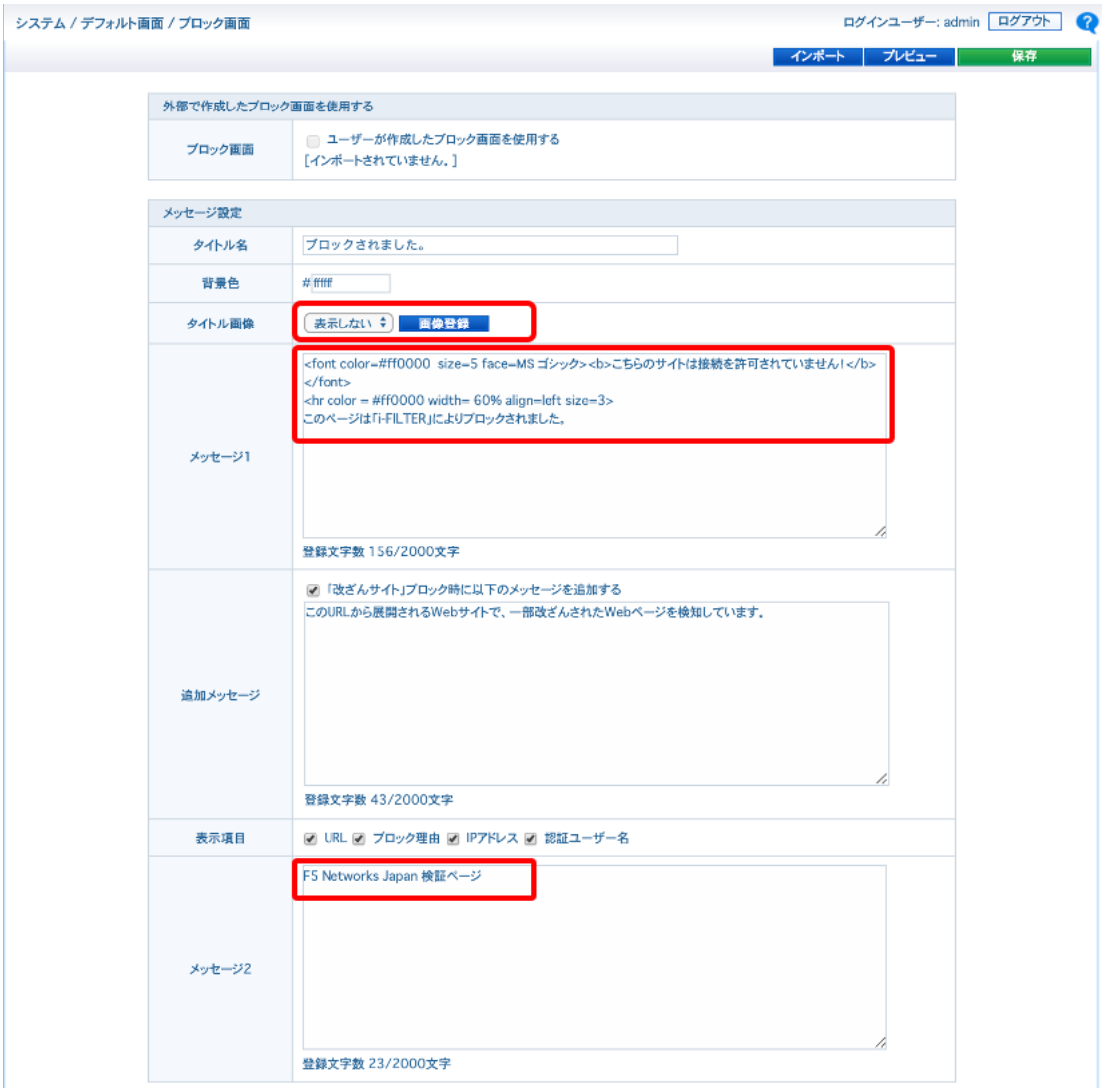
ブロック画面のタイトル画像連携設定

1. **システム／システムパラメーター／動作設定** において、**仮想ホスト名設定** の **コンテンツ転送** の値を、で設定した SSLO の Interception Rule の Destination Address の IP アドレスに変更します。



2. HTTPS 接続の場合、SSLO を通すとブロック画面のタイトル画像が表示されません。回避方法例は 1.13 で説明します。

3. ここでは、簡易的にブロック画面中にタイトル画像を表示せず、**システム／デフォルト画面／ブロック画面** において、タイトル画像を **表示しない** を選択し、**保存** ボタンを押します。必要に応じて、メッセージも変更します。



注釈:

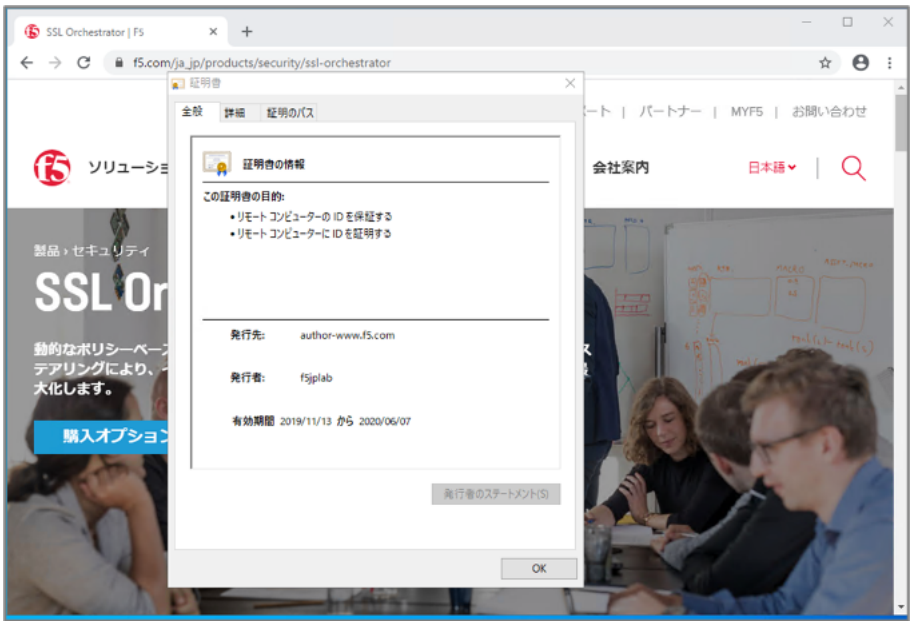
- 必要に応じて、Connection: keep-alive を有効にしてください。
- i-FILTER 製品の詳細に関しては、メーカーにお問い合わせください。

1.2.12 クライアントからの接続テスト

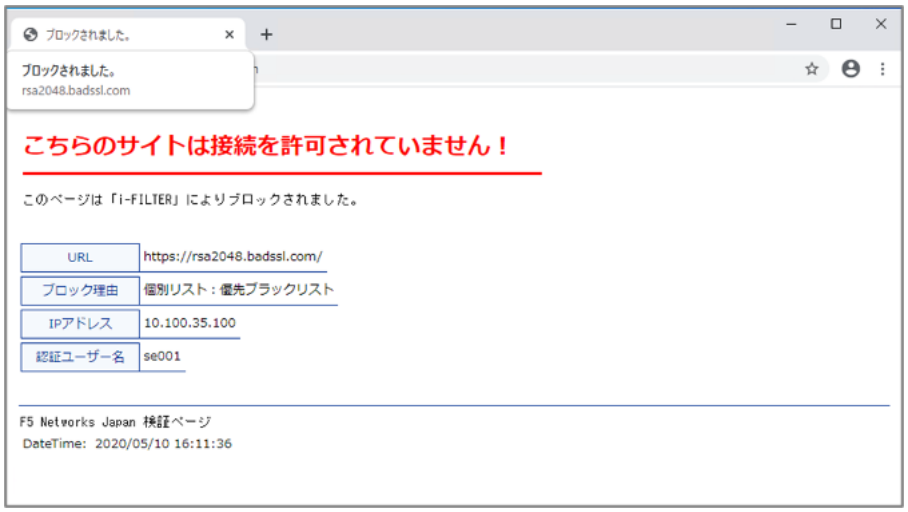
1. クライアントに SSLO の **CA 証明書** をインストールします。
2. コマンドラインで確認する場合は、以下を利用します。

```
curl -vk --proxy 10.100.35.201:3128 -proxy-ntlm -proxy-user 'se001:ilovef5!'␣
↪https://xxxx.xxx
```

3. ブラウザで確認する場合は、プロキシ設定に SSLO を加えてから WEB 接続を確認します。
4. WEB 通信が成功するか確認します。サーバ証明書が SSLO の CA 証明書で **書き換えられている** ことを確認します。



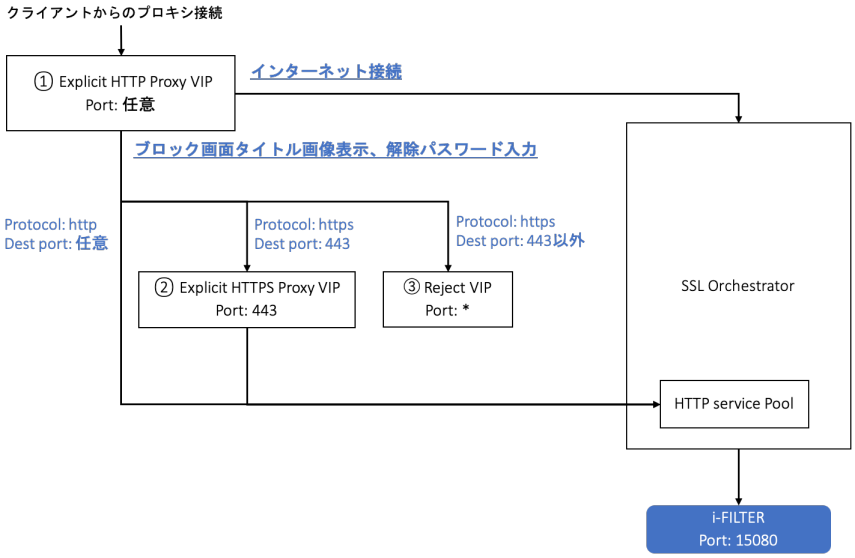
5. ブラックリストの宛先への通信が **ブロック** されることを確認します。



6. SSL 復号除外サイトの通信が成功するか確認します。サーバ証明書が SSLO の CA 証明書で **書き換えられていない** ことを確認します。

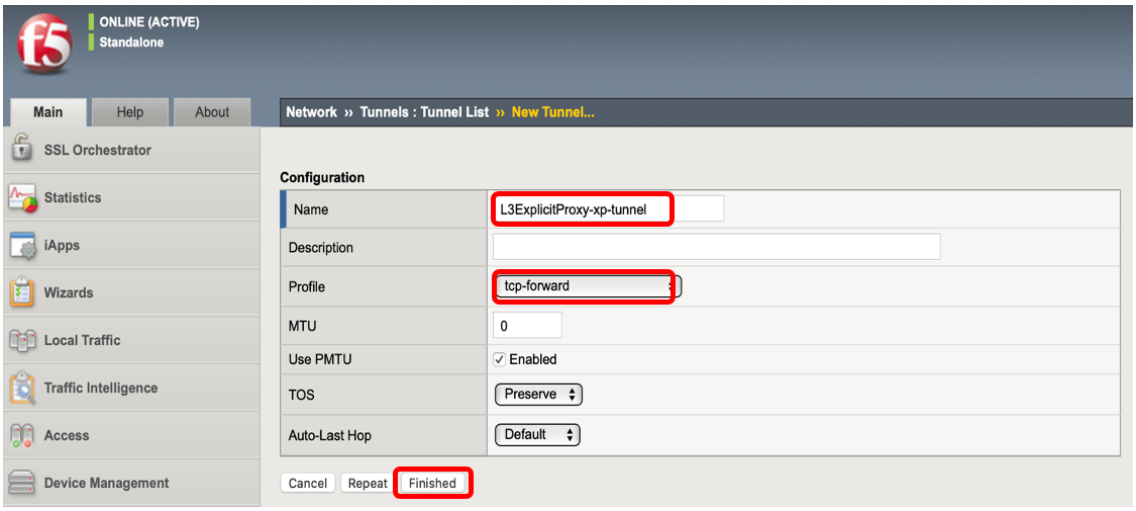
1.2.13 i-FILTER ブロック画面のタイトル画像の表示、パスワードブロック解除の設定

2.11.3 にて、SSLO 連携をするとブロック画面のタイトルが標準では表示されないと記載しましたが、以下の設定を追加することによって、表示させることが可能です。本設定を行うことにより、パスワード入力によるブロック解除も連携が可能となります。上記を実現するためには、下図の 3 つの Virtual Server が追加で必要となります。ここでは、下図の âŚăâŚąâŚć の順で作成していきます。

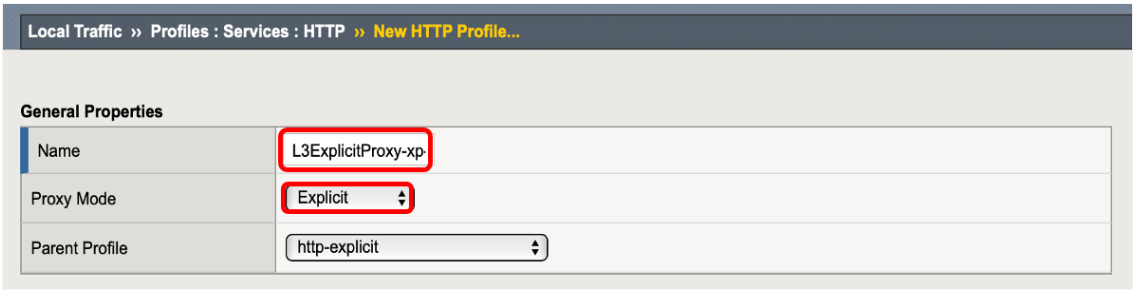


Explicit HTTP Proxy 用 Virtual Server の作成

1. Proxy 用の Tunnel を作成します。**Network >> Tunnel List**Âă にて、*Create*Âă ボタンを押します。任意の名前 Âă を入力し、ProfileÂă にて、tcp-forwardÂă を選択し、*Finished* ボタンを押します。



2. プロキシ用の Explicit Profile を作成します。**Local Traffic >> Profiles >> Services >> HTTP**Âă にて、*Create*Âă ボタンを押します。任意の名前を設定し、**Proxy Mode**Âă にて、**Explicit**Âă を選択します。



3. **DNS Resolver**Âă にて、SSLO Guided Configuration で作成した Âă**DNS Resolver**Âă を選択し、**Tunnel Name**Âă にて、作成済みの Âă**Tunnel**Âă を選択し、*Finished* ボタンを押します。

4. 次に、以下の 2 つの iRule を作成します。

   - インターネット接続用の HTTP/HTTPs トラフィックと i-FILTER ブロックタイトル画面接続トラフィックを分ける iRule
   - 上記後者のトラフィックにおいて、i-FILTER 向けにヘッダーを追加する iRule

   1 つ目の iRule では、ホスト名でインターネット接続トラフィックと判断した場合、SSLO の Explicit 用の Virtual Serer に転送します。2 つ目の iRule では、ホスト名で i-FILTER 宛の通信と判断した場合、ヘッダーを追記して、i-FILTER 用の Pool に転送します。各 iRule 内のホスト名は、環境にあわせた FQDN/IP アドレスに変更して頂く必要があります。

   Local Traffic >> iRulesÂă にて、*Create*Âă ボタンを押します。**任意の名前** Âă を入力して、**Definition**Âă に以下サンプル iRule を入力し、*Finished*Âă ボタンを押します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。また、以下の 2 つの iRule は、1 つのファイルにしていただいても構いません。）

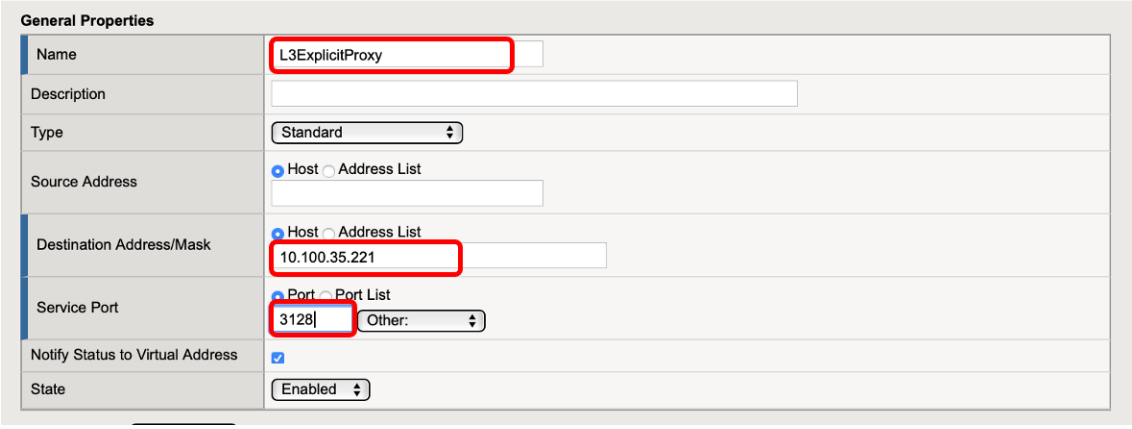
   例）トラフィックを分ける用の iRule

```
### Add this iRule to Explicit HTTP Proxy Virtual Server  ###
when HTTP_PROXY_REQUEST {
    set F5PROXY "bigip.f5jplab.local"    # For block jpg #
    set F5PROXY2 "10.100.35.221”         # For password bypass #
    if { [HTTP::host] contains $F5PROXY || [HTTP::host] contains $F5PROXY2 }
↪{
        HTTP::proxy enable
    } else {
        HTTP::proxy disable
        virtual sslo_L3ExplicitProxy.app/sslo_L3ExplicitProxy-xp-4
        snat automap
    }
}
```

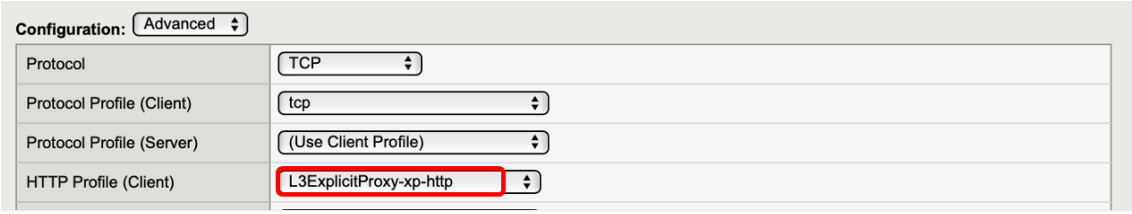
   例）ヘッダー追加用の iRule

```
###  Add this iRule to Explicit HTTP Proxy Virtual Server ###
when HTTP_REQUEST {
    set F5PROXY "bigip.f5jplab.local" # For block jpg #
    set F5PROXY2 "10.100.35.221”       # For password bypass #
    if { [HTTP::host] contains $F5PROXY || [HTTP::host] contains $F5PROXY2 } {
        if { [HTTP::method] ne "CONNECT" } {
            HTTP::header replace "X-Forwarded-Proto" "http"
            pool sslo$_$iFILTERProxy.app/sslo$_$iFILTERProxy
        }
    }
}
```

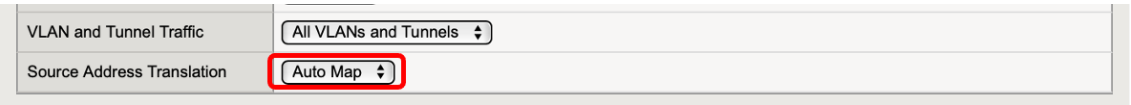
5. Explicit Proxy 用の Virtual Server を作成します。**Local Traffic >> Virtual Servers**Âă にて、*Create*Âă ボタンを押します。**任意の名前** Âă を入力し、**Destination Address/Mask**Âă にて、プロキシ接続用の Âă**IP アドレス** Âă を入力、**Service Port**Âă にて、プロキシとして利用する Âă **ポート番号** Âă を入力します。



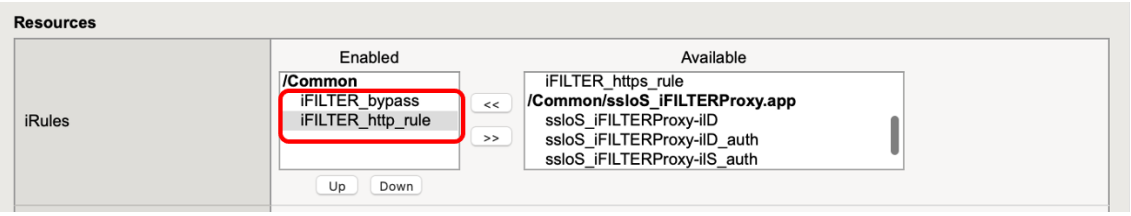
6. **HTTP Profile(Client)**Âă にて、作成済みの Âă**HTTP Explicit Profile**Âă を選択します。



7. **Source Address Translation**Âă にて、**Auto Map**Âă を選択します。



8. **iRules**Âă にて作成済みの 2 つの iRule を選択して、*Finished*Âă ボタンを押します。



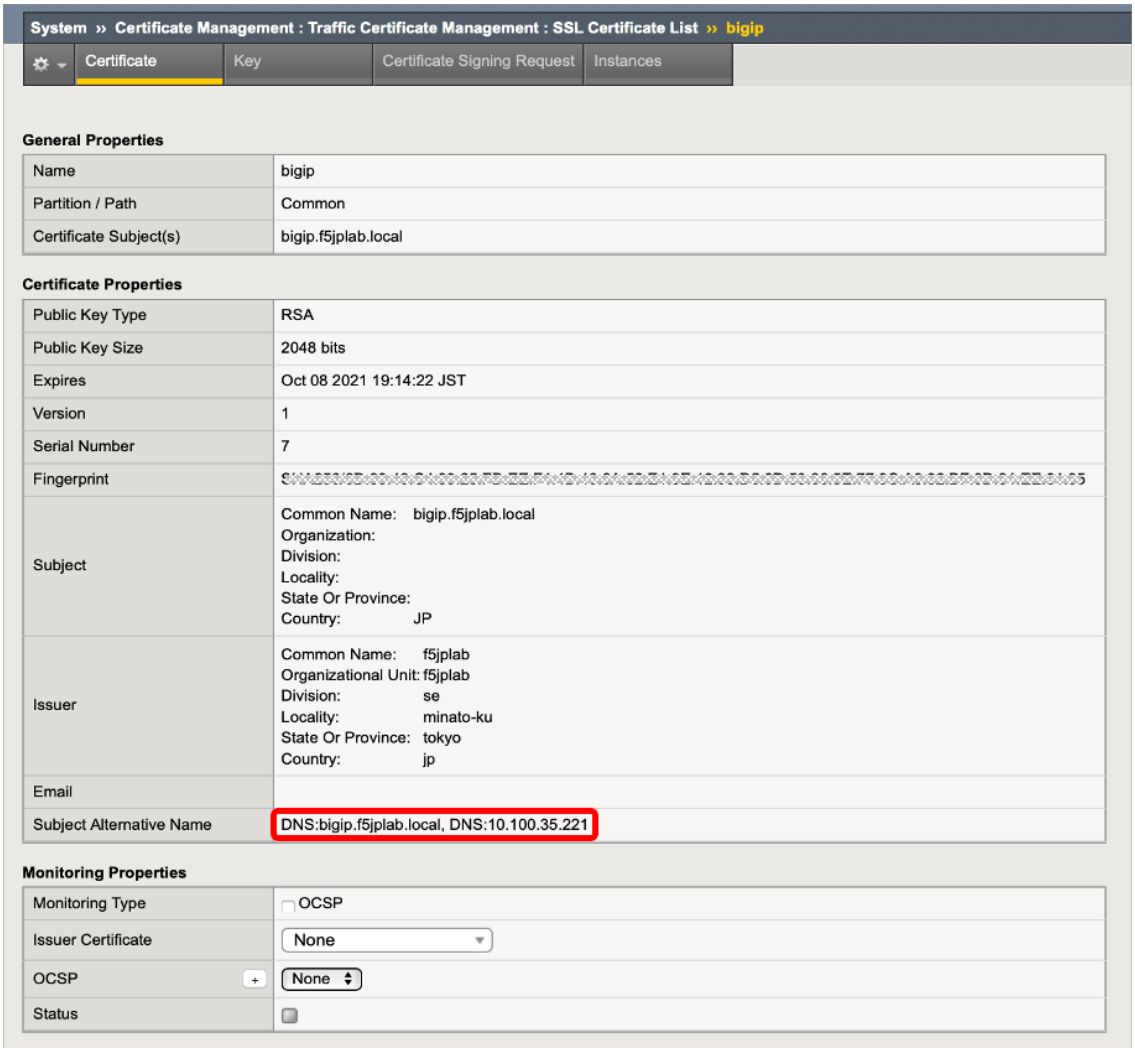
Explicit HTTPS Proxy 用 Virtul Server の作成

1. HTTPS トラフィックにおける i-FILTER 向けヘッダ追加用の iRule を作成します。この iRule では、ヘッダーを追記して、i-FILTER 用の Pool に転送します。 **Local Traffic >> iRules**Âă にて、*Create*Âă ボタンを押します。Âă **任意の名前** Âă を入力して、ÂăDefinitionÂă に以下サンプル **iRule**Âă を入力し、*Finished*Âă ボタンを押します。（以下の iRule はあくまでもサンプルとなります。同じ主旨の内容であれば下記と同じでなくても構いません。）

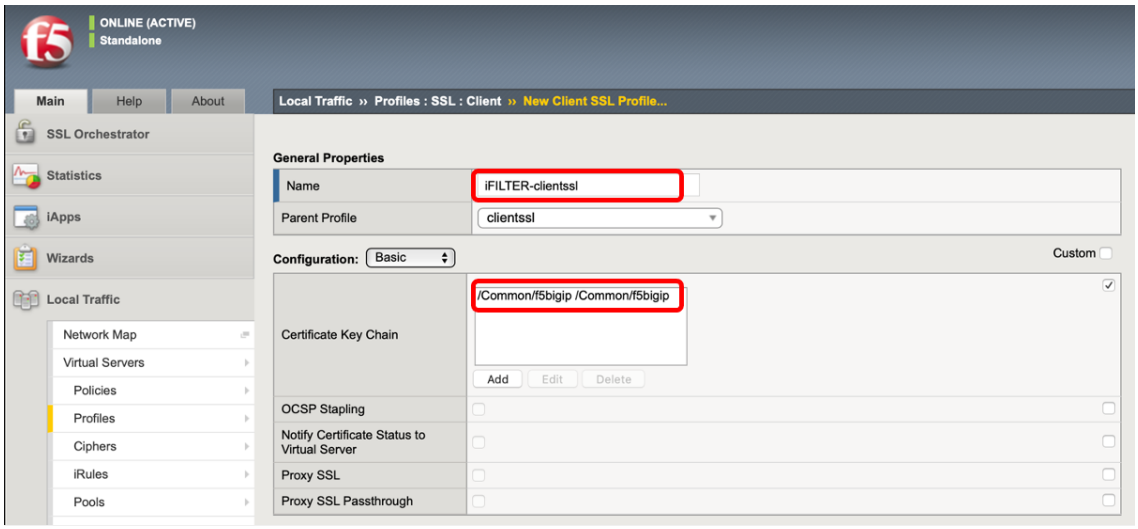
   例）ヘッダー追加用の iRule

```
### Add this iRule to Explicit HTTPs Proxy Virtual Server ###
when HTTP_REQUEST {
    HTTP::header replace "X-Forwarded-Proto" "https"
    pool ssloS_iFILTERProxy.app/ssloS_iFILTERProxy
}
```

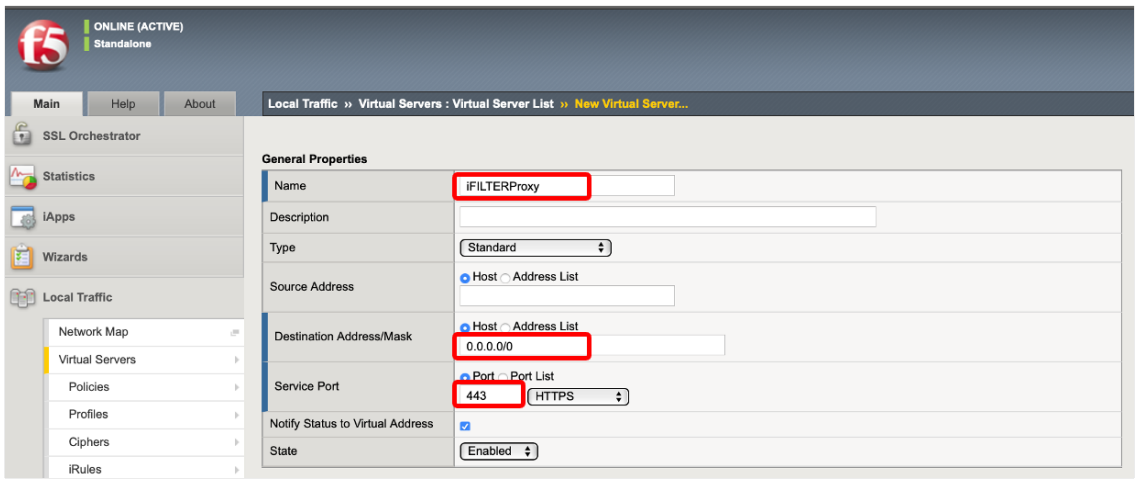
2. ブロック画面内のタイトル画像に SSL 接続するために BIG-IP にて SSL オフロードを行うため、**サーバ証明書** Âă と Âă **秘密鍵** Âă の登録します。 **System >> Certificate Management >> Traffic Certificate Management**Âă にて、**利用するサーバ証明書** Âă と Âă **秘密鍵** Âă の登録します。下記が登録したサーバ証明書のイメージです。ここでは、SAN に FQDN と IP アドレスを登録しています。



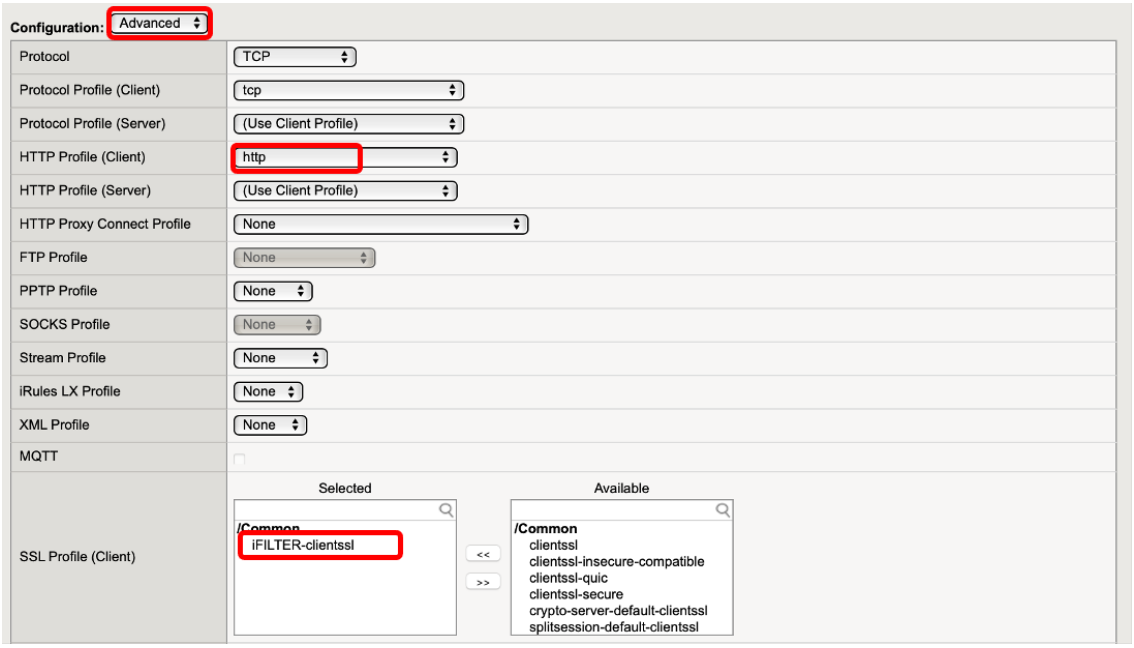
3. 次に、SSL サイトにおけるブロック時に、ブロック画面内のタイトル画像に SSL 接続するための ClientSSL プロファイルを作成します。**Local Traffic >> Profiles >> SSL >> Client**Âă にて、*Create*Âă ボタンを押します。**任意の名前** Âă を入力し、先程登録した **サーバ証明書** Âă と Âă **秘密鍵** Âă を選択し、*Finished*Âă ボタンを押します。



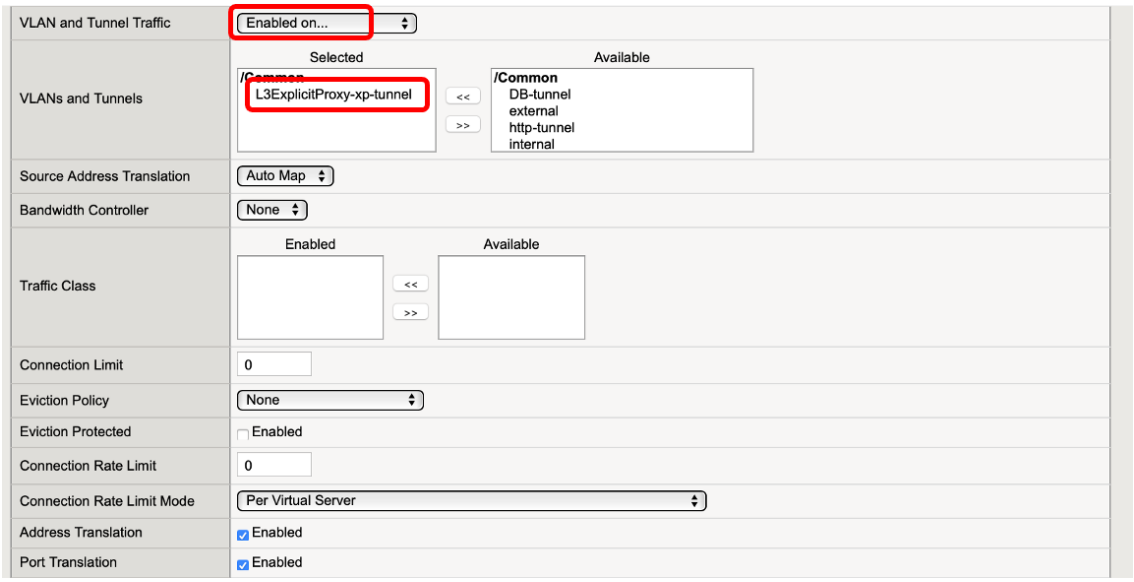
4. HTTPS 用の Explicit Proxy Virtual Server を作成します。**Local Traffic >> Virtual Servers**Âă にて、*Create*Âă ボタンを押します。**任意の名前** Âă を入力し、**Destination Address/Mask**Âă にて、**0.0.0.0/0**Âă を入力、**Service Port**Âă にて、**443**Âă と入力します。



5. **Configuration**Âă にて、**Advanced**Âă を選択し、**HTTP Profile**Âă にて、**http**Âă を選択、**SSL Profile (Client)**Âă にて、作成済みの Âă**clientssl プロファイル** Âă を選択します。



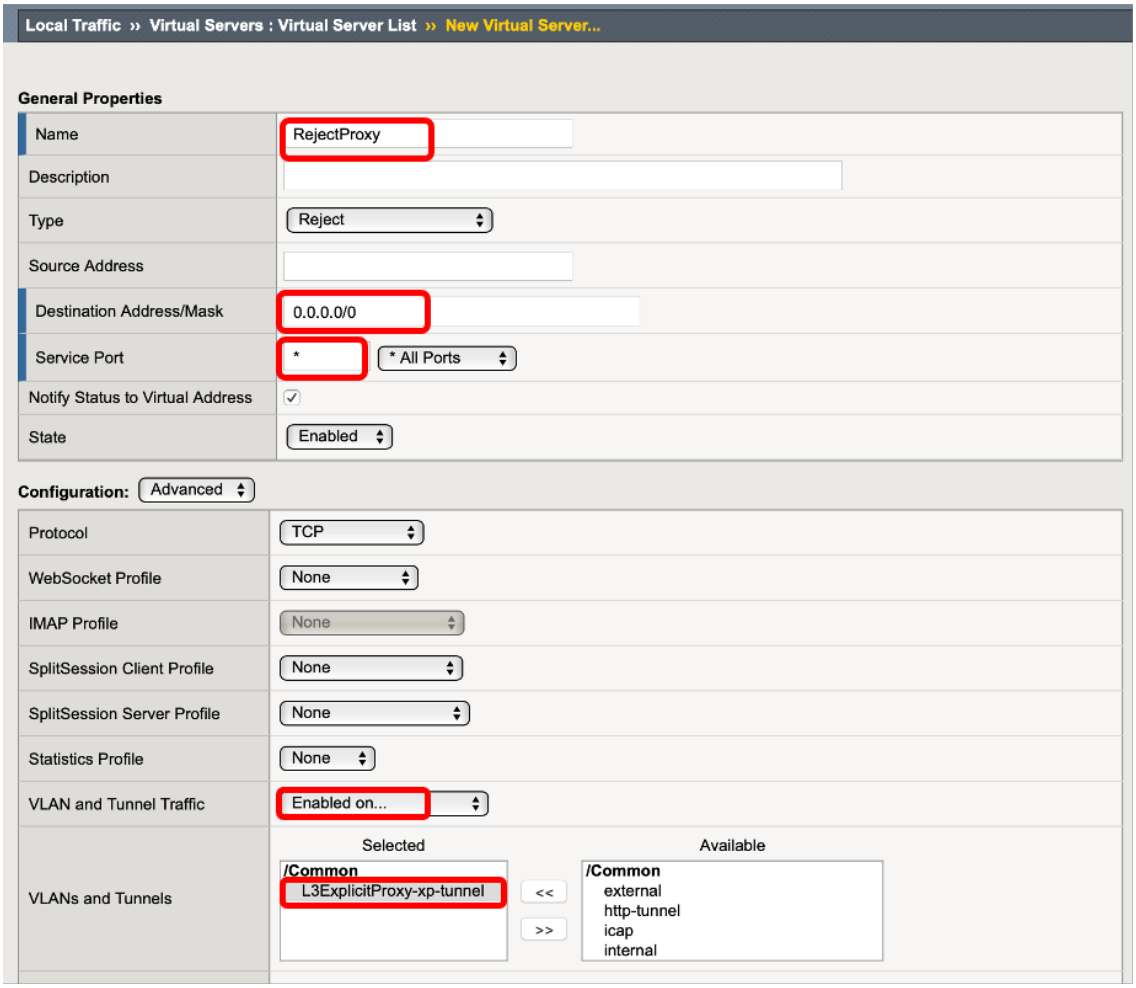
6. **VLAN and Tunnel Traffic**Âă にて、**Enable on...**Âă を選択し、**VLANs and Tunnels**Âă にて作成済みの ÂăTunnelÂă を選択します。



7. **iRulesÂă** にて、作成済みの **ÂăiRuleÂă** を選択し、*Finished* ボタンを押します

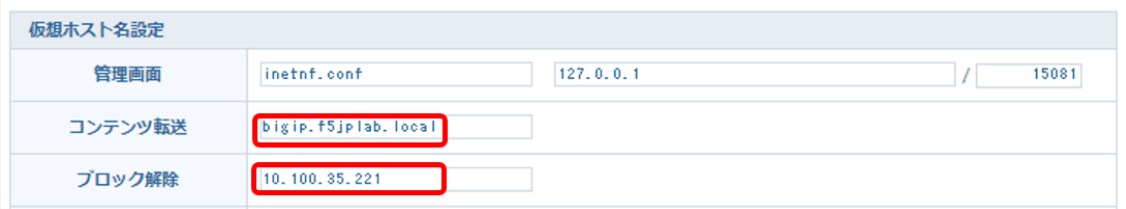
Reject 用 Virtual Server の作成

1. 最後に、443 ポート以外はブロックをする Virtual Server を作成します。（本 Virtual Server がなくても動作はします。）任意の名前を入力し、**Type** にて **Reject** を選択し、**Destination Address/Mask** にて、**0.0.0.0/0** と入力し、**Service Port** にて * と入力します。**VLAN and Tunnel Traffic** にて、**Enabled on...** を選択し、**VLANs and Tunnels** にて作成済みの **Tunnel** を選択し、*Finished* ボタンを押します。



i-FILTER 側の設定

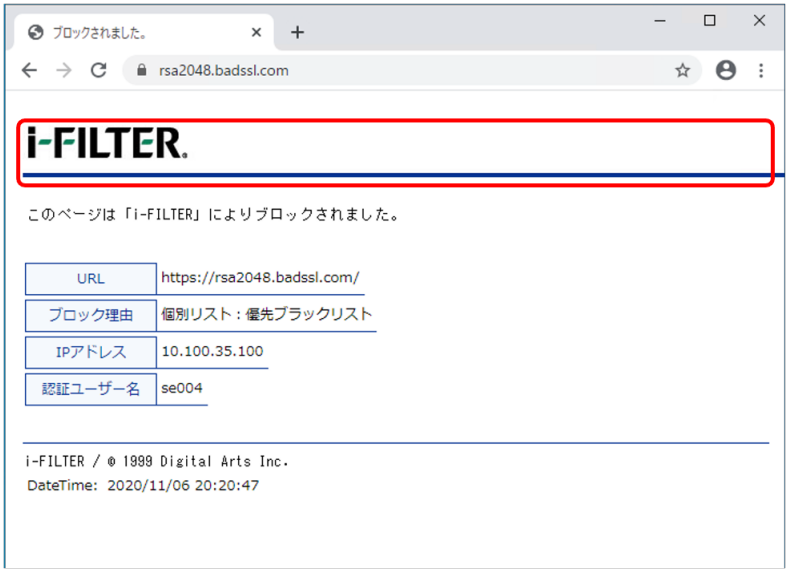
1. **システム／システムパラメーター／動作設定** にて、**コンテンツ転送** 、**ブロック解除** のそれぞれにサーバ証明書に登録した DNS 名を設定します。（ブロック画面タイトル表示時、パスワードブロック解除時の URI の一部として利用されます。）



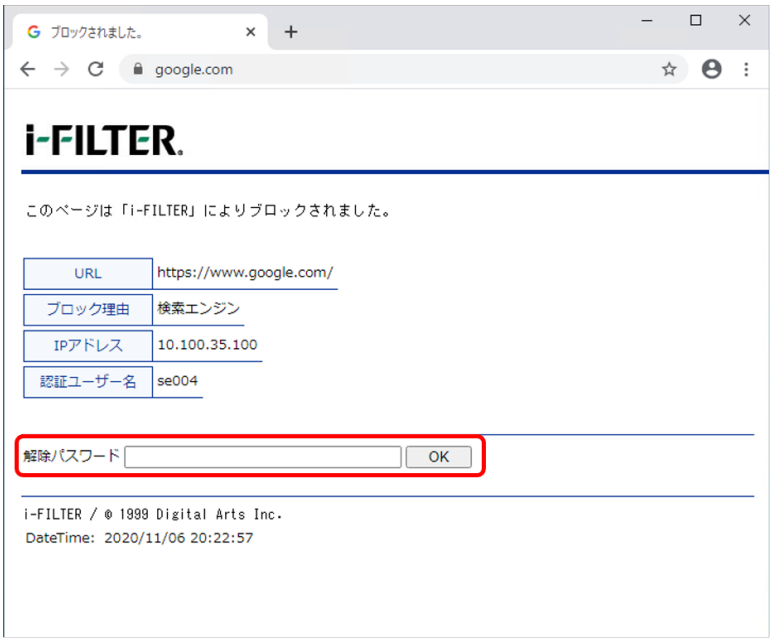
クライアントからの接続テスト

1. ブラウザの Âă プロキシ設定 Âă にて、作成済みの Explicit Proxy の IP アドレスに紐づく ÂăFQDNÂă または、**IP アドレス** に変更します。

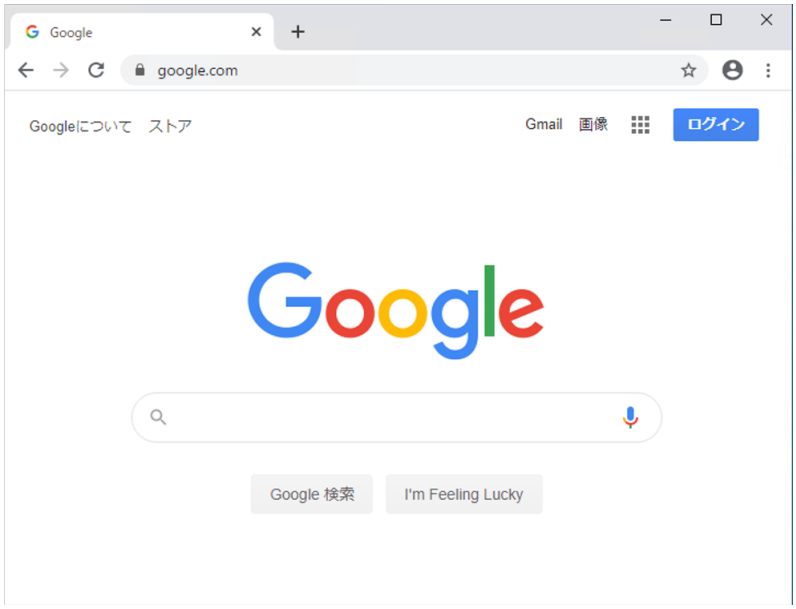
2. ブラックリストの宛先への通信がブロックされ、タイトル画像が表示されることを確認します。



3. パスワードブロック解除が設定されている URL へ接続し、パスワード入力画面が表示されることを確認します。



4. 解除パスワード入力後、無事 WEB 接続ができる事を確認します。



1.2.14 その他

各種ログ

1. **テストの際には、以下の** SSLO **におけるログを参照して下さい。**

    - SSLO **全般ログや** iRule **で** local0 **で指定したログ**　/var/log/ltm
    - APM**（認証）のログ**　/var/log/apm
    - SSL Guided Configuration**（**iAppLX/REST**）に関わるログ**　/var/log/restnoded/restnoded.log
      /var/log/restjava.0.log
    - **その他、**SSLO **に関するトラブルシューティングは以下の** AskF5 **記事をご参照下さい。**
      K26520133: Troubleshooting SSL Orchestrator(SSLO) https://support.f5.com/csp/article/K26520133

2. **テストの際には、以下の** i-FILTER **におけるログを参照して下さい。**

    - **アクセスログ**　/usr/local/ifilter10/logs/access.log0000

フィルター除外設定（必須ではございません）

1. 本テストでは、httpbin.org を接続先ホストとして利用させて頂いておりますが、現状デフォルトではブロックされるので、**システム／システムフィルター／推奨フィルター設定 – 除外設定** にて、テストで利用しているホスト名を **除外設定** に追加します。

i-FILTER 通信について

1. i-FILTER の DB アップデートやシステムアップデートに必要な通信に関しては、i-FILTER 用サーバにて管理用 VLAN を通して、アップデートできるように設定して下さい。