Cisco Secure Firewall Management Center(FMC)でのプロキシのトラブルシューティング

内容

はじめに

このドキュメントでは、ユーザが中間サーバを介してインターネットに接続できるようにFMC上でプロキシを設定して、セキュリティを強化し、場合によってはパフォーマンスを向上させる方法について説明します。この記事では、FMCでプロキシを設定する手順を説明し、一般的な問題のトラブルシューティングのヒントを提供します。

要件

次の項目に関する知識があることが推奨されます。

- Cisco Secure Firewall Management Center(FMC)

- プロキシ

使用するコンポーネント

このドキュメントの情報は、次のソフトウェアとハードウェアのバージョンに基づいています。

- FMC 7.4.x

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな（デフォルト）設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

コンフィギュレーション

FMC GUIでネットワークhttp-proxyを設定します。

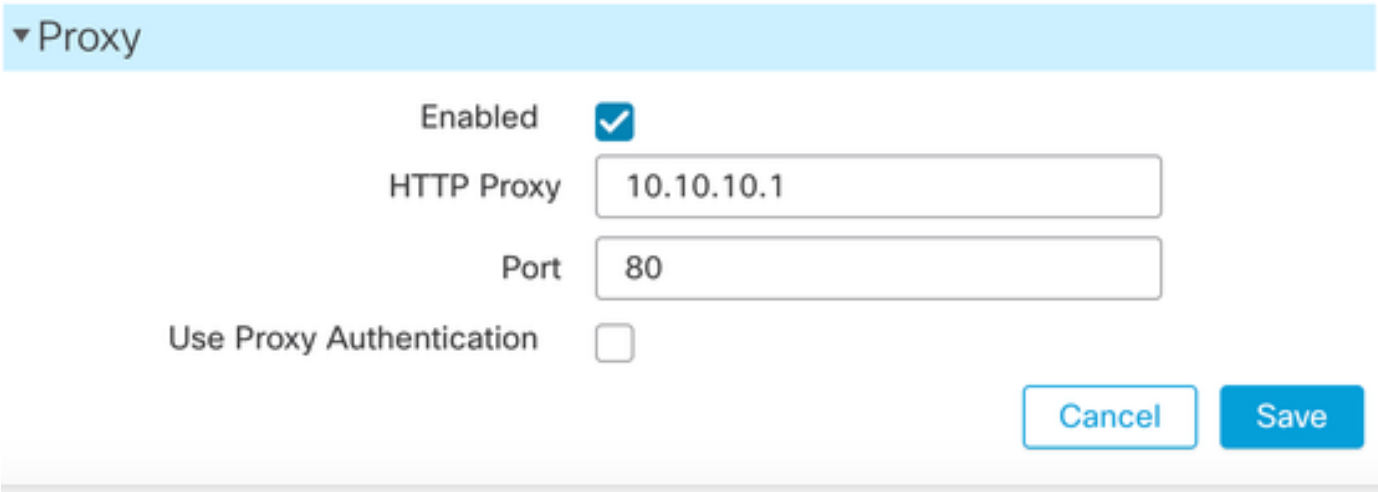
FMC GUIにログインし、System > Configurationの順に選択してから、Management Interfacesを選択します。

注:NT LAN Manager(NTLM)認証を使用するプロキシはサポートされません。スマートライセンスを使用する場合、プロキシFQDNは64文字以下にする必要があります。

Proxy領域で、HTTPプロキシを設定します。

Management Centerは、ポートTCP/443(HTTPS)およびTCP/80(HTTP)でインターネットに直接接続するように設定されています。 プロキシサーバを使用して、HTTPダイジェストで認証することができます。

- Enabledcheckボックスにチェックマークを入れます。
- HTTPプロキシフィールドに、プロキシサーバのIPアドレスまたは完全修飾ドメイン名を入力します。
- ポートフィールドにポート番号を入力します。
- Use Proxy Authenticationを選択して認証資格情報を指定し、aUser NameandPasswordを指定します。
- [Save] をクリックします。



注：プロキシパスワードには、A ~ Z、a ~ z、0 ~ 9および特殊文字を使用できます。

トラブルシュート

FMC CLIとエキスパートモードにアクセスし、iprep_proxy.confを確認してプロキシ設定が正しいことを確認します。

<#root>

admin@fmc:~$

cat /etc/sf/iprep_proxy.conf

iprep_proxy {
PROXY_HOST 10.10.10.1;
PROXY_PORT 80;
}

アクティブなプロキシ接続を確認するために、アクティブな接続を確認します。

<#root>

admin@fmc:~$

netstat -na | grep 10.10.10.1

tcp 0 0 10.40.40.1:40220 10.10.10.1:80

ESTABLISHED

curlコマンドを使用して、要求の詳細とプロキシからの応答の両方を確認します。「HTTP/1.1 200 Connection established」という応答が表示された場合は、FMCがプロキシ経由でトラフィックを正常に送受信していることを示しています。

<#root>

admin@fmc:~$

curl -x http://10.10.10.1:80 -I https://tools.cisco.com

HTTP/1.1 200 Connection established

プロキシのユーザ名とパスワードを設定した場合は、認証とプロキシ応答を確認します。

```
curl -u proxyuser:proxypass --proxy http://proxy.example.com:80 https://example.com
```

検証

プロキシを介した接続の確立の成功

curl -x http://proxy:80 -I https://tools.cisco.comなどのプロキシを使用してcurlコマンドを実行すると、予期される一連のネットワークの相互作用が発生します。これは、パケットキャプチャ(tcpdump)を使用して確認できます。 次に、プロセスの概要を実際のtcpdump出力と統合して示します。

TCPハンドシェイクの開始：
クライアント(FMC)は、SYNパケットを送信して、ポート80でプロキシサーバへのTCP接続を開始します。プロキシはSYN-ACKで応答し、クライアントはACKでハンドシェイクを完了します。これにより、HTTP通信が行われるTCPセッションが確立されます。

tcpdumpの出力例：

```
10:20:58.987654 IP client.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > client.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], le
10:20:58.987734 IP client.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT要求：
TCP接続が確立されると、クライアントはHTTP CONNECT要求をプロキシに送信し、ターゲットのHTTPSサーバ(tools.cisco.com:443)へのトンネルを作成するように指示します。 この要求に

より、クライアントはプロキシを介してエンドツーエンドのTLSセッションをネゴシエートできます。

例tcpdump (デコードされたHTTP) :

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: curl/8.5.0
Proxy-Connection: Keep-Alive
```

接続確立確認応答：
プロキシは、ターゲットサーバへのトンネルが正常に作成されたことを示すHTTP/1.1 200 Connection established応答で応答します。これは、プロキシがリレーとして機能し、クライアントとtools.cisco.comの間で暗号化されたトラフィックを転送していることを意味します。

例tcpdump:

<#root>

```
HTTP/1.1
```

200

```
 Connection established
```

トンネル経由のHTTPS通信：
正常なCONNECT応答に続いて、クライアントは、確立されたトンネルを介してtools.cisco.comとのSSL/TLSハンドシェイクを直接開始します。このトラフィックは暗号化されるため、内容はtcpdumpには表示されませんが、TLS Client HelloパケットやServer Helloパケットを含む、パケットの長さとタイミングを確認できます。

例tcpdump:

```
10:20:59.123456 IP client.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > client.54321: Flags [P.], length 1514 (Server Hello)
```

HTTPリダイレクトの処理 (302が見つかりました) :
HTTPS通信の一部として、クライアントはtools.cisco.comにリソースを要求します。サーバはHTTP/1.1 302 Foundで別のURL(https://tools.cisco.com/healthcheck)へのリダイレクトに応答します。クライアントはこのリダイレクトに従うことができ、curlパラメータと要求の目的に応じて

異なります。このリダイレクトは暗号化されたTLSセッション内で発生し、直接認識されませんが、予期される動作であり、TLSトラフィックが復号化された場合に確認できます。

暗号化されたリダイレクトトラフィックは次のようになります。

```
10:21:00.123000 IP client.54321 > proxy.80: Flags [P.], length 517 (Encrypted Application Data)
10:21:00.123045 IP proxy.80 > client.54321: Flags [P.], length 317 (Encrypted Application Data)
```

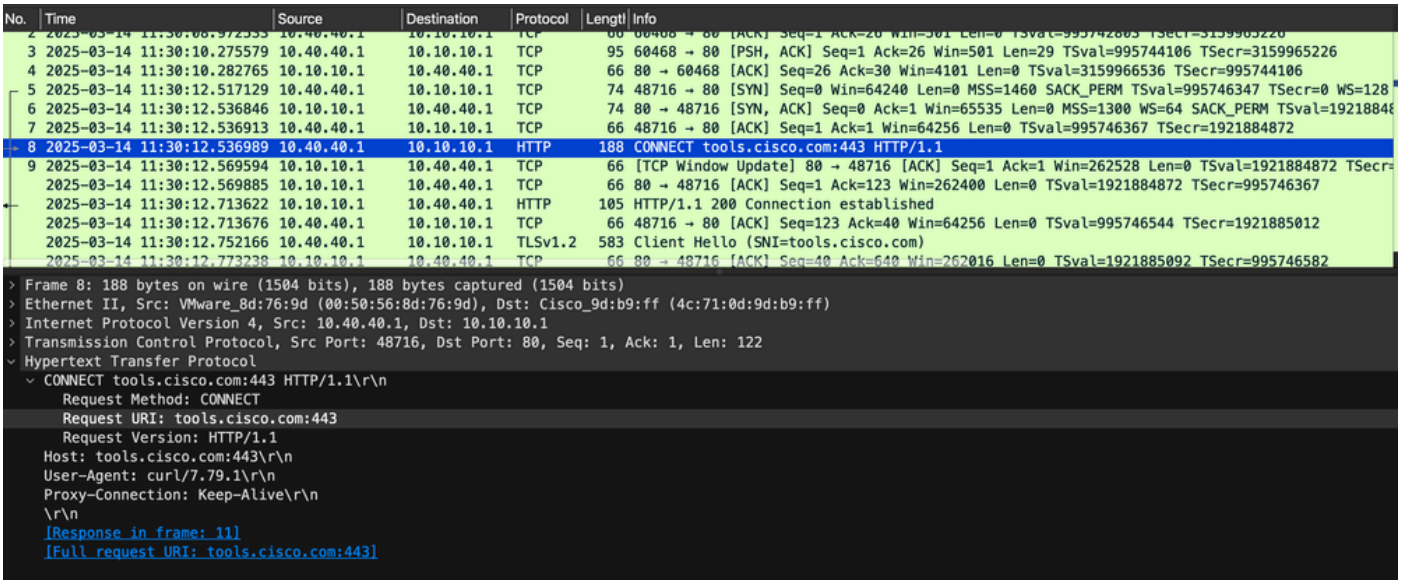
接続のティアダウン:
交換が完了すると、クライアントとプロキシの両方がFINパケットとACKパケットを交換してTCP接続を正常に閉じ、適切なセッションの終了を保証します。

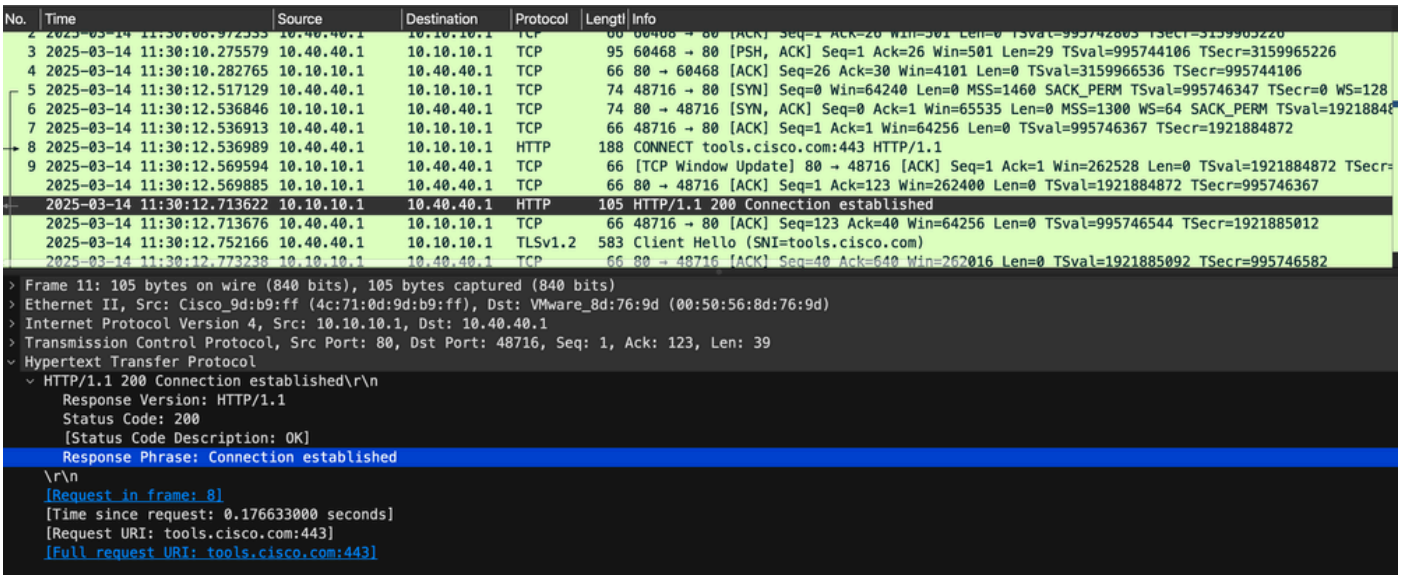
tcpdumpの出力例：

```
10:21:05.000111 IP client.54321 > proxy.80: Flags [F.], seq 1234, ack 5678, length 0
10:21:05.000120 IP proxy.80 > client.54321: Flags [F.], seq 5678, ack 1235, length 0
10:21:05.000125 IP client.54321 > proxy.80: Flags [.], ack 5679, length 0
```

ヒント:tcpdumpの出力を分析することにより、明示的なプロキシによるHTTPS要求が、TCPハンドシェイク、CONNECT要求、トンネル確立、TLSハンドシェイク、暗号化された通信（リダイレクトの可能性を含む）、および正常な接続の終了という期待されるフローに従っていることを確認できます。これにより、プロキシとクライアントのインタラクションが設計どおりに動作していることが確認され、トンネリングやSSLネゴシエーションの失敗など、フロー内の問題の特定に役立ちます。

FMC(10.40.40.1)は、ポート80でプロキシ(10.10.10.1)との正常なTCPハンドシェイクを確立し、その後、ポート443でサーバ(72.163.4.161)へのHTTP CONNECTを確立します。サーバはHTTP 200 Connection establishedメッセージで応答します。TLSハンドシェイクが完了し、データフローが適切に行われます。最後に、TCP接続は正常に終了します(FIN)。





既知の問題

プロキシACLの制限

権限の問題（プロキシ上のアクセスリストなど）がある場合は、パケットキャプチャ(tcpdump)を使用してそれを確認できます。 次に、障害シナリオの概要と、tcpdump出力例を示します。

TCPハンドシェイクの開始：
クライアント(Firepower)は、ポート80でプロキシへのTCP接続を確立することで起動します。TCPハンドシェイク(SYN、SYN-ACK、ACK)が正常に完了します。これは、プロキシが到達可能であることを意味します。

tcpdumpの出力例：

```
10:20:58.987654 IP client.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
```

```
10:20:58.987700 IP proxy.80 > client.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], le
10:20:58.987734 IP client.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT要求：
接続されると、クライアントはプロキシにHTTP CONNECT要求を送信し、tools.cisco.com:443へのトンネルの作成を要求します。

例tcpdump (デコードされたHTTP) :

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: curl/8.5.0
Proxy-Connection: Keep-Alive
```

プロキシからのエラー応答：

トンネルを許可する代わりに、おそらくアクセスリスト(ACL)がこのトラフィックを許可していないことが原因で、プロキシは要求を拒否します。プロキシは403 Forbiddenまたは502 Bad Gatewayのようなエラーで応答します。

エラーを示すtcpdump出力例：

<#root>

```
HTTP/1.1
```

403

```
 Forbidden
Content-Type: text/html
Content-Length: 123
Connection: close
```

接続のティアダウン:
エラーメッセージの送信後、プロキシは接続を閉じ、両側がFIN/ACKパケットを交換します。

tcpdumpの出力例：

```
10:21:05.000111 IP client.54321 > proxy.80: Flags [F.], seq 1234, ack 5678, length 0
10:21:05.000120 IP proxy.80 > client.54321: Flags [F.], seq 5678, ack 1235, length 0
10:21:05.000125 IP client.54321 > proxy.80: Flags [.], ack 5679, length 0
```

ヒント:tcpdumpを見ると、TCP接続およびHTTP CONNECT要求は成功しましたが、プロキシがトンネル設定を拒否していることがわかります。これは通常、トラフィックの通過を妨げるACLまたは権限の制限がプロキシにあることを示します。

プロキシがダウンロードに失敗する（タイムアウト/未完了の転送）

このシナリオでは、FMCはプロキシに正常に接続し、ファイルのダウンロードを開始しますが、転送がタイムアウトするか、完了しません。これは通常、プロキシのインスペクション、タイムアウト、またはプロキシのファイルサイズ制限が原因で発生します。

TCPハンドシェイクの開始
FMCがポート80でプロキシへのTCP接続を開始し、ハンドシェイクが正常に完了します。
tcpdumpの出力例：

```
10:20:58.987654 IP fmc.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > fmc.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], lengt
10:20:58.987734 IP fmc.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT要求
FMCは、外部ターゲットに到達するためにプロキシにHTTP CONNECT要求を送信します。
例tcpdump（デコードされたHTTP）：

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: FMC-Agent
Proxy-Connection: Keep-Alive
```

トンネルの確立とTLSハンドシェイク
プロキシはHTTP/1.1 200 Connection establishedで応答し、TLSハンドシェイクの開始を許可します。

tcpdumpの出力例：

```
<#root>

HTTP/1.1

200

 Connection established
```

```
10:20:59.123456 IP fmc.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > fmc.54321: Flags [P.], length 1514 (Server Hello)
```

タイムアウトまたは不完全なダウンロード
ファイル転送を開始した後、ダウンロードが停止するか、完了しないため、タイムアウトが発生します。接続はアイドル状態のままです。
考えられる理由は次のとおりです。

- プロキシ検査の遅延またはフィルタリング。
- 長時間転送のプロキシタイムアウト。
- プロキシによって課されたファイルサイズの制限。

非アクティブを示すtcpdumpの例：

<#root>

```
10:21:00.456000 IP fmc.54321 > proxy.80: Flags [P.], length 1440
```

```
# FMC sending data
```

```
# No response from proxy, connection goes idle...
```

```
# After a while, FMC may close the connection or retry.
```

ヒント: FMCがダウンロードを開始しますが、タイムアウトまたは不完全な転送が原因で完了しません。これは多くの場合、プロキシフィルタリングやファイルサイズの制限が原因です。

プロキシがファイルのダウンロードに失敗する（MTUの問題）

この場合、FMCはプロキシに接続し、ファイルのダウンロードを開始しますが、MTUの問題が原因でセッションが失敗します。これらの問題は、特に大きなファイルやSSL/TLSハンドシェイクでは、パケットのフラグメント化やパケットのドロップの原因となります。

TCPハンドシェイクの開始
FMCがプロキシとのTCPハンドシェイクを開始し、これが成功します。
tcpdumpの出力例：

```
10:20:58.987654 IP fmc.54321 > proxy.80: Flags [S], seq 0, win 64240, options [mss 1460], length 0
10:20:58.987700 IP proxy.80 > fmc.54321: Flags [S.], seq 0, ack 1, win 65160, options [mss 1460], lengt
```

```
10:20:58.987734 IP fmc.54321 > proxy.80: Flags [.], ack 1, win 64240, length 0
```

HTTP CONNECT要求とトンネル確立
FMCがHTTP CONNECT要求を送信し、プロキシが応答してトンネルの確立を許可します。
例tcpdump（デコードされたHTTP）：

```
CONNECT tools.cisco.com:443 HTTP/1.1
Host: tools.cisco.com:443
User-Agent: FMC-Agent
Proxy-Connection: Keep-Alive
```

TLSハンドシェイクの開始
FMCとtools.cisco.comがSSL/TLSのネゴシエーションを開始し、初期パケットが交換されます。
tcpdumpの出力例：

```
<#root>

HTTP/1.1

200

 Connection established
10:20:59.123456 IP fmc.54321 > proxy.80: Flags [P.], length 517 (Client Hello)
10:20:59.123789 IP proxy.80 > fmc.54321: Flags [P.], length 1514 (Server Hello)
```

MTUによるパケットフラグメンテーションまたはドロップ
FMCまたはサーバが大きなパケットを送信しようとすると、MTUの問題によりパケットのフラグメント化またはドロップが発生し、その結果ファイル転送またはTLSネゴシエーションが失敗します。
これは通常、FMCとプロキシ間（またはプロキシとインターネット間）のMTUが誤って設定されているか、または小さすぎる場合に発生します。
フラグメンテーションの試みを示すtcpdumpの例：

```
<#root>

10:21:00.456000 IP fmc.54321 > proxy.80: Flags [P.], length 1440

# Large packet

10:21:00.456123 IP proxy.80 > fmc.54321: Flags [R], seq X, win 0, length 0

# Proxy resets connection due to MTU issue
```

ヒント:MTUの問題により、パケットが廃棄またはフラグメント化され、TLSハンドシェイクが中断するか、ファイルのダウンロードが失敗する。これは、MTUの設定が誤っているためにSSLインスペクションまたはパケットフラグメンテーションが発生する場合によく見られます。

障害シナリオでは、おそらくMTUの問題かプロキシ/アップストリームの問題が原因で、再送信とFINによりTLS/データ交換が行われないことが確認された状態で、FMCがHTTP 200なしでCONNECTを取得します。

curlを使用すると、サーバ側の問題や認証エラーを示すさまざまなHTTP応答コードが発生する場合があります。次に、最も一般的なエラーコードとその意味のリストを示します。

HTTPコード	意味	原因
400	不正な要求	要求の構文が正しくありません
401	Unauthorized	クレデンシャルが欠落しているか正しくない
403	禁止	アクセス拒否
404	見つかりません	リソースが見つかりません
500	Internal Error	サーバ エラー
502	不正なゲートウェイ	サーバの通信ミス
503	利用不能なサービス	サーバの過負荷またはメンテナンス
504	ゲートウェイタイムアウト	サーバ間のタイムアウト

参考資料

Cisco Secure Firewall Threat Defenseリリースノート、バージョン7.4.x

翻訳について
シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。