CHAPTER 16

フィルタリング サービスの適用

この章では、Web トラフィックをフィルタリングして、セキュリティ リスクを低減し、不適切な使用を回避する方法について説明します。この章で説明する内容は、次のとおりです。

- フィルタリングの概要（p.16-1）
- ActiveX オブジェクトのフィルタリング（p.16-2）
- Java アプレットのフィルタリング（p.16-4）
- 外部サーバによる URL および FTP 要求のフィルタリング（p.16-5）
- フィルタリングの統計情報とフィルタリング設定の表示（p.16-11）

フィルタリングの概要

ここでは、フィルタリングが FWSM を通過するトラフィックをより制御できる方法について説明します。フィルタリングは次の 2 つの方法に使用できます。

- ActiveX オブジェクトまたは Java アプレットのフィルタリング
- 外部フィルタリング サーバを使用した URL のフィルタリング

アクセスを一緒にブロックしないで、特定の不適切なオブジェクト（ActiveX オブジェクトまたは Java アプレットなど）を HTTP トラフィックから削除できます。これは、所定の状況でセキュリティ リスクが発生させることがあります。

URL フィルタリングを使用して、特定のトラフィックを N2H2 Sentian または Websense フィルタリング サーバなどの外部フィルタリング サーバへ転送できます。フィルタリング サーバは、セキュリティ ポリシーで指定された特定のサイトへのトラフィック、またはサイト タイプへのトラフィックをブロックできます。

URL フィルタリングは CPU 中心なので、外部フィルタリング サーバを使用すると、他のトラフィックのスループットは影響を受けません。ただし、ネットワーク速度と URL フィルタリング サーバのキャパシティに応じて、外部フィルタリング サーバを使用してトラフィックをフィルタリングするときに、初期接続に必要な時間は著しく遅くなります。

ActiveX オブジェクトのフィルタリング

ここでは、フィルタリングを適用して、ファイアウォールを通過する HTTP トラフィックから ActiveX オブジェクトを削除する手順について説明します。次の内容について説明します。

- ActiveX フィルタリングの概要（p.16-2）
- ActiveX フィルタリングのイネーブル化（p.16-2）

ActiveX フィルタリングの概要

ActiveX オブジェクトには保護されたネットワーク上のホストやサーバを攻撃する目的のコードが含まれているので、セキュリティ リスクを発生させることがあります。ActiveX フィルタリングを使用して ActiveX オブジェクトをディセーブルにできます。

ActiveX コントロール（従来の OLE コントロールまたは OCX コントロール）は、Web ページまたはその他のアプリケーションに挿入できるコンポーネントです。これらのコントロールには、カスタム フォーム、カレンダー、または情報の収集と表示に使用するサード パーティ製の広範なフォームが含まれています。ActiveX は、技術的に、ネットワーク クライアントに対して多くの問題を発生させる可能性があります。たとえば、ワークステーションの障害の原因となる、ネットワーク セキュリティ問題を引き起こす、サーバへの攻撃に利用されるなどの恐れがあります。

filter activex コマンドは、HTML <object> コマンドを HTML Web ページ内でコメントアウトすることでブロックします。HTML ファイルの ActiveX のフィルタリングは、<APPLET> タグ、</APPLET> タグ、<OBJECT CLASSID> タグ、</OBJECT> タグを選別し、コメントで置き換えることによって実行されます。ネストされたタグのフィルタリングは、最上位タグをコメントに変換することによってサポートされます。

注意　このコマンドは、オブジェクト タグに組み込まれた Java アプレット、イメージ ファイル、またはマルチメディア オブジェクトもブロックします。

<object> または </object> HTML タグがネットワーク パケットに分割されている場合、またはタグ内のコードが Maximum Transmission Unit（MTU; 最大伝送ユニット）のバイト数より長い場合、FWSM はそのタグをブロックできません。

ActiveX ブロックは、ユーザが ***alias*** コマンドによって参照される IP アドレスにアクセスしている場合は実行されません。

ActiveX フィルタリングのイネーブル化

ここでは、FWSM を通過する HTTP トラフィック内の ActiveX オブジェクトを削除する方法について説明します。ActiveX オブジェクトを削除するには、グローバル コンフィギュレーション モードで次のコマンドを入力します。

```
hostname(config)# filter activex {port[-port] | except} local_ip local_mask foreign_ip
foreign_mask
```

このコマンドを使用するには、フィルタリングを適用する TCP ポートに *port* を置き換えます。一般的には、これはポート 80 ですが、他の値は受け入れられます。**http** または **url** の文字列はポート 80 に使用できます。開始ポート番号と終了ポート番号の間にハイフンを使用することでポート範囲を指定できます。

以前のフィルタリング条件の例外を作成するには、キーワード **except** を指定します。

ローカル IP アドレスとマスクは、フィルタリングされるトラフィックの送信元である 1 つまたは複数の内部ホストを識別します。外部アドレスとマスクは、フィルタリングされるトラフィックの外部宛先を指定します。

すべてのホストを指定するには、いずれかのアドレスを **0.0.0.0**（または短縮形 **0**）に設定します。すべてのホストを指定するには、いずれかのマスクとして **0.0.0.0**（または短縮形 **0**）を使用します。

次に、すべての発信接続で ActiveX オブジェクトをブロックする例を示します。

```
hostname(config)# filter activex 80 0 0 0 0
```

このコマンドは、ActiveX オブジェクト ブロックが、あらゆるローカル ホストからあらゆる外部ホストへのポート 80 の Web トラフィックに対して、適用されることを指定しています。

この設定を削除するには、次の例のように、このコマンドの **no** 形式を使用します。

```
hostname(config)# no filter activex 80 0 0 0 0
```

Java アプレットのフィルタリング

ここでは、フィルタリングを適用して、ファイアウォールを通過する HTTP トラフィックから Java アプレットを削除する手順について説明します。Java アプレットには保護されたネットワーク上のホストやサーバを攻撃する目的のコードが含まれているので、セキュリティ リスクを発生させることがあります。**filter java** コマンドを使用して、Java アプレットを削除できます。

filter java コマンドは、発信接続から FWSM へ戻る Java アプレットをフィルタリングします。それでもユーザは HTML ページを受信できますが、Java アプレットの Web ページの発信元がコメントアウトされるため、アプレットは実行できなくなります。

（注）　<object> タグに組み込まれた Java アプレットを削除するには、**filter activex** コマンドを使用します。

FWSM を通過する HTTP トラフィック内の Java アプレットを削除するには、グローバル コンフィギュレーション モードで次のコマンドを入力します。

```
hostname(config)# filter java {port[-port] | except} local_ip local_mask foreign_ip
foreign_mask
```

このコマンドを使用するには、フィルタリングを適用する TCP ポートに *port* を置き換えます。一般的には、これはポート 80 ですが、他の値は受け入れられます。**http** または **url** の文字列はポート 80 に使用できます。開始ポート番号と終了ポート番号の間にハイフンを使用することでポート範囲を指定できます。

以前のフィルタリング条件の例外を作成するには、キーワード **except** を指定します。

ローカル IP アドレスとマスクは、フィルタリングされるトラフィックの送信元である 1 つまたは複数の内部ホストを識別します。外部アドレスとマスクは、フィルタリングされるトラフィックの外部宛先を指定します。

すべてのホストを指定するには、いずれかのアドレスを **0.0.0.0**（または短縮形 **0**）に設定します。すべてのホストを指定するには、いずれかのマスクとして **0.0.0.0**（または短縮形 **0**）を使用します。

すべてのホストを指定するには、アドレスを **0.0.0.0**（または短縮形 **0**）に設定します。すべてのホストを指定するには、マスクとして **0.0.0.0**（または短縮形 **0**）を使用します。

次に、すべての発信接続で Java アプレットをブロックする例を示します。

```
hostname(config)# filter java 80 0 0 0 0
```

このコマンドは、Java アプレット ブロックが、あらゆるローカル ホストからあらゆる外部ホストへのポート 80 の Web トラフィックに対して、適用されることを指定しています。

次に、保護されたネットワーク上のホストに Java アプレットがダウンロードされないようにする例を示します。

```
hostname(config)# filter java http 192.168.3.3 255.255.255.255 0 0
```

このコマンドは、ホスト 192.168.3.3 による Java アプレットのダウンロードをブロックします。

この設定を削除するには、次の例のように、このコマンドの **no** 形式を使用します。

```
hostname(config)# no filter java http 192.168.3.3 255.255.255.255 0 0
```

外部サーバによる URL および FTP 要求のフィルタリング

ここでは、外部サーバによる URL および FTP 要求をフィルタリングする手順について説明します。次の内容について説明します。

- URL フィルタリングの概要（p.16-5）
- フィルタリング サーバの指定（p.16-5）
- コンテンツ サーバの応答のバッファリング（p.16-6）
- サーバ アドレスのキャッシング（p.16-7）
- HTTP URL のフィルタリング（p.16-8）
- HTTPS URL のフィルタリング（p.16-9）
- FTP 要求のフィルタリング（p.16-10）

URL フィルタリングの概要

安全性の高いネットワークから安全性の低いネットワークへの接続要求にフィルタリングを適用できます。アクセス リストを使用して、特定のコンテンツ サーバへの発信アクセスを阻止できますが、インターネットの規模およびダイナミック特性を考慮すると、この方法での使用の管理は困難です。次のいずれかのインターネット フィルタリング製品で稼働する別途サーバを使用することで、設定を簡素化し、FWSM のパフォーマンスを向上できます。

- HTTP、HTTPS、FTP のフィルタリング用 Websense Enterprise
- HTTP のフィルタリング専用の N2H2 による Sentian（一部の Sentian バージョンは HTTPS をサポートしていますが、FWSM がサポートしているのは Sentian の HTTP フィルタリングだけです）。

外部サーバを使用するときは FWSM のパフォーマンスはほとんど影響を受けませんが、フィルタリング サーバが FWSM から離れた場所にある場合には、Web サイトまたは FTP サーバへのアクセス時間が大幅に長くなることがあります。

フィルタリングがイネーブルで、接続要求を FWSM 経由で転送すると、その要求はコンテンツ サーバとフィルタリング サーバに同時に送信されます。フィルタリング サーバによって接続が許可されると、FWSM はコンテンツ サーバからの応答を、発信元のクライアントに転送します。フィルタリング サーバが接続を拒否した場合、FWSM は応答を廃棄し、接続が成功しなかったことを示すメッセージまたはリターン コードを送信します。

認証が FWSM 上でイネーブルの場合、FWSM はまたユーザ名をフィルタリング サーバに送信します。フィルタリング サーバで、ユーザ名のフィルタリング設定を使用するか、使用に関する拡張レポート機能を提供できます。

フィルタリング サーバの指定

各コンテキストに最大 4 つのフィルタリング サーバを指定できます。FWSM は、サーバから応答が得られるまで、各サーバを順番に使用します。コンフィギュレーションに指定できるサーバは、1 つのタイプ（Websense または N2H2）だけです。

（注）　**filter** コマンドで HTTP または HTTPS のフィルタリングを設定するには、事前にフィルタリング サーバを追加する必要があります。コンフィギュレーションからフィルタリング サーバを削除すると、すべての **filter** コマンドも一緒に削除されます。

url-server コマンドを使用してフィルタリング サーバのアドレスを指定します。

Websense の場合は次のとおりです。

```
hostname(config)# url-server (if_name) host local_ip [timeout seconds] [protocol TCP
connections number| UDP version 1|4]
```

N2H2 の場合は次のとおりです。

```
hostname(config)# url-server (if_name) vendor n2h2 host local_ip[:port number]
[timeout seconds] [protocol TCP connections number | UDP]
```

if_name には、フィルタリング サーバに接続される FWSM インターフェイスの名前を指定します。*local_ip* には、フィルタリング サーバの IP アドレスを指定します。*seconds* には、FWSM がフィルタリング サーバへの接続を試行する秒数を指定します。

（注）　デフォルト ポートは 4005 です。これは、TCP または UDP 経由で FWSM と通信するのに N2H2 サーバが使用するデフォルト ポートです。デフォルト ポートの変更の詳細については、『*Filtering by N2H2 Administrator's Guide*』を参照してください。

たとえば、1 つの Websense フィルタリング サーバを指定するには、次のコマンドを入力します。

```
hostname(config)# url-server (perimeter) host 10.0.1.1 protocol TCP version 4
```

このコマンドは、Websense フィルタリング サーバに FWSM の境界インターフェイス上の IP アドレス 10.0.1.1 を指定します。バージョン 4 ではキャッシングがサポートされているので、この例では Websense はイネーブルであるバージョン 4 を推奨します。

冗長 N2H2 Sentian サーバを指定するには、次のコマンドを入力します。

```
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.1
hostname(config)# url-server (perimeter) vendor n2h2 host 10.0.1.2
```

このコマンドは、FWSM の境界インターフェイス上で 2 つの Sentian フィルタリング サーバを指定します。

コンテンツ サーバの応答のバッファリング

ユーザが コンテンツ サーバへの接続要求を発行すると、FWSM はその要求を コンテンツ サーバとフィルタリング サーバに同時に送信します。フィルタリング サーバがコンテンツ サーバより先に応答しなかった場合、サーバからの応答は廃棄されます。これにより、Web クライアントは要求を再発行する必要があるので、Web クライアントからの Web サーバの応答が遅れます。

HTTP 応答バッファをイネーブルにすることにより、Web コンテンツ サーバからの応答はバッファリングされ、フィルタリング サーバによって接続が許可された場合に、要求クライアントに転送されます。これにより、他の遅延の発生を回避します。

HTTP または FTP 要求への応答のバッファリングを設定する手順は、次のとおりです。

ステップ 1　フィルタリング サーバからの応答が保留中である HTTP または FTP 要求に対する応答のバッファリングをイネーブルにするには、次のコマンドを入力します。

```
hostname(config)# url-block block block-buffer-limit
```

block-buffer-limit に、バッファリングするブロックの最大数を指定します。

（注）　1159 バイトより長い URL のバッファリングは、Websense フィルタリング サーバでのみサポートされています。

ステップ 2　保留中の URL のバッファリング（および Websense による長い URL のバッファリング）に使用できる最大メモリを設定するには、次のコマンドを入力します。

```
hostname(config)# url-block url-mempool memory-pool-size
```

memory-pool-size に、最大メモリ割り当ての 2 KB ～ 10 MB に相当する 2 ～ 10240 の値を指定します。

サーバ アドレスのキャッシング

ユーザがサイトにアクセスしたあと、宛先アドレスでホスティングされている全サイトが常時許可されるカテゴリに含まれていれば、フィルタリング サーバは一定時間、FWSM にサーバ アドレスのキャッシュを許可できます。これにより、ユーザが同じサーバに再アクセスしたり、別のユーザが同じサーバにアクセスした場合、FWSM からフィルタリング サーバに再度、問い合わせる必要がありません。

（注）　キャッシュされている IP アドレスへの要求は、フィルタリング サーバに転送されないのでロギングされません。その結果、これらの動作は記録されません。***url-cache*** コマンドを使用する前に、Websense 実行ログを蓄積できます。

スループットを高める必要がある場合は、次のように **url-cache** コマンドを使用します。

```
hostname(config)# url-cache {dst | src_dst} size
```

size に、1 ～ 128 KB 範囲のキャッシュ サイズの値を指定します。

URL 宛先アドレスに基づいて、エントリをキャッシュするには、***dst*** キーワードを使用します。すべてのユーザが Websense サーバ上で同一の URL フィルタリング ポリシーを共有している場合に、このモードを選択します。

URL 要求を開始した発信元アドレスと URL 宛先アドレスの両方に基づいて、エントリをキャッシュするには、***src_dst*** キーワードを使用します。ユーザが Websense サーバ上で同一の URL フィルタリング ポリシーを共有していない場合に、このモードを選択します。

HTTP URL のフィルタリング

ここでは、外部フィルタリング サーバを使用して HTTP フィルタリングを設定する手順について説明します。次の内容について説明します。

- HTTP フィルタリングの設定（p.16-8）
- 長い HTTP URL のフィルタリングのイネーブル化（p.16-8）
- 長い HTTP URL の短縮（p.16-9）
- フィルタリングから除外するトラフィックを指定（p.16-9）

HTTP フィルタリングの設定

HTTP フィルタリングをイネーブルにする前に、URL フィルタリング サーバを指定およびイネーブルにする必要があります。

フィルタリング サーバが HTTP 接続要求を許可すると、FWSM は Web サーバからの応答を発信元のクライアントに到達させます。フィルタリング サーバによって応答が拒否された場合、FWSM はアクセスが拒否されたことを示すブロック ページにユーザをリダイレクトします。

HTTP フィルタリングをイネーブルにするには、次のコマンドを入力します。

```
hostname(config)# filter url [http | port[-port] | except} local_ip local_mask
foreign_ip foreign_mask] [allow] [proxy-block]
```

HTTP（80）のデフォルト ポートとは異なるポートが使用されている場合、*port* に、1 つまたは複数のポート番号を指定します。*local_ip* と *local_mask* には、要求を行うユーザまたはサブネットワークの IP アドレスとサブネット マスクを指定します。*foreign_ip* と *foreign_mask* には、要求に応答するサーバまたはサブネットワークの IP アドレスとサブネット マスクを指定します。

以前のフィルタリング条件の例外を作成するには、キーワード **except** を指定します。

allow オプションは、プライマリ フィルタリング サーバが利用できないときに、FWSM がフィルタリングせずに HTTP トラフィックを転送するようにします。プロキシ サーバへの要求をすべて廃棄するには、***proxy-block*** コマンドを使用します。

長い HTTP URL のフィルタリングのイネーブル化

デフォルトでは、FWSM は、1159 文字を超える長さの HTTP URL を長い URL であるとみなします。Websense サーバの場合、最大長を増加できます。

（Websense のみ）次のコマンドを入力して、1 つの URL の最大サイズを設定します。

```
hostname(config)# url-block url-size long_url_size
```

long_url_size に、最大 URL サイズの 2 ～ 4 KB に相当する 2 ～ 4 の値を指定します。デフォルト値は 2 です。

（Websense のみ）次のコマンドを入力して、URL バッファ メモリ プールの最大サイズを設定することもできます。

```
hostname(config)# url-block url-mempool memory_pool_size
```

memory_pool_size に、URL バッファ メモリ プール サイズの 2 ～ 10,240 KB に相当する 2 ～ 10240 の値を指定します。

長い HTTP URL の短縮

最大許可サイズを超える URL は、デフォルトでは廃棄されます。これを回避するには、次のコマンドを入力して、長い URL を短縮するように FWSM を設定できます。

```
hostname(config)# filter url [longurl-truncate | longurl-deny | cgi-truncate]
```

URL が最大許可長を超えている場合、**longurl-truncate** オプションを指定すると、フィルタリング サーバで評価するために、FWSM は URL のホスト名または IP アドレスの部分だけを送信します。URL が最大許可長を超えている場合、**longurl-deny** オプションを指定すると、発信 URL トラフィックは拒否されます。

cgi-truncate オプションを指定すると、CGI URL を、CGI スクリプトの場所およびスクリプト名（パラメータは含まない）だけになるように短縮します。長い HTTP 要求の多くは、CGI 要求です。パラメータ リストが非常に長い場合、パラメータ リストを含む完全な CGI 要求を待機および送信すると、メモリ リソースが浪費され、ファイアウォールのパフォーマンスに影響します。

フィルタリングから除外するトラフィックを指定

フィルタリングから除外する特定のトラフィックを指定するには、次のコマンドを入力します。

```
hostname(config)# filter url except source_ip source_mask dest_ip dest_mask
```

たとえば、次のコマンドを使用すると、10.0.2.54 からの要求を除く、すべての HTTP 要求をフィルタリング サーバに転送します。

```
hostname(config)# filter url http 0 0 0 0
hostname(config)# filter url except 10.0.2.54 255.255.255.255 0 0
```

HTTPS URL のフィルタリング

HTTPS フィルタリングをイネーブルにする前に、URL フィルタリング サーバを指定およびイネーブルにする必要があります。

（注）　HTTPS URL のフィルタリングは、Websense フィルタリング サーバでのみサポートされています。

HTTPS コンテンツは暗号化されているため、FWSM からフィルタリング サーバに送信される URL 検索には、ディレクトリおよびファイル名の情報は含まれません。フィルタリング サーバが HTTPS 接続要求を許可すると、FWSM は SSL 接続ネゴシエーションを完了させ、Web サーバからの応答を発信元のクライアントに到達させます。フィルタリング サーバが要求を拒否すると、FWSM は SSL 接続ネゴシエーションの完了を阻止します。ブラウザには、「The Page or the content cannot be displayed.（ページまたはコンテンツを表示できません）」などのエラー メッセージが表示されます。

（注）　FWSM は HTTPS 用に認証プロンプトを提供しないので、HTTPS サーバにアクセスする前に、HTTP または FTP を使用して FWSM でまずユーザを認証する必要があります。

HTTPS フィルタリングをイネーブルにするには、次のコマンドを入力します。

```
hostname(config)# filter https port localIP local_mask foreign_IP foreign_mask [allow]
```

HTTPS（443）のデフォルト ポートとは異なるポートが使用されている場合、*port* に、ポート番号を指定します。

（注）　HTTPS と HTTP トラフィックの両方に同じ GET 要求がある場合、HTTPS プロトコル インスペクタも指定したポート番号上の HTTP トラフィックをフィルタリングします。

local_ip と *local_mask* には、要求を行うユーザまたはサブネットワークの IP アドレスとサブネット マスクを指定します。*foreign_ip* と *foreign_mask* には、要求に応答するサーバまたはサブネットワークの IP アドレスとサブネット マスクを指定します。

allow オプションは、プライマリ フィルタリング サーバが利用できないときに、FWSM がフィルタリングせずに HTTPS トラフィックを転送するようにします。

FTP 要求のフィルタリング

FTP フィルタリングをイネーブルにする前に、URL フィルタリング サーバを指定およびイネーブルにする必要があります。

（注）　FTP URL のフィルタリングは、Websense フィルタリング サーバでのみサポートされています。

フィルタリング サーバが FTP 接続要求を許可すると、FWSM は成功を示す FTP リターン コードを発信元のクライアントに到達させます。たとえば、「250: CWD command successful.」は成功したリターン コードです。フィルタリング サーバが要求を拒否した場合、接続が拒否されたことを示すため FTP リターン コードを変更します。たとえば、FWSM は、コード 250 を「550 Requested file is prohibited by URL filtering policy.」に変更します。

FTP フィルタリングをイネーブルにするには、次のコマンドを入力します。

```
hostname(config)# filter ftp {port[-port] | except} localIP local_mask foreign_IP
foreign_mask [allow] [interact-block]
```

FTP（21）のデフォルト ポートとは異なるポートが使用されている場合、*port* に、ポート番号を指定します。*local_ip* と *local_mask* には、要求を行うユーザまたはサブネットワークの IP アドレスとサブネット マスクを指定します。*foreign_ip* と *foreign_mask* には、要求に応答するサーバまたはサブネットワークの IP アドレスとサブネット マスクを指定します。

以前のフィルタリング条件の例外を作成するには、キーワード **except** を指定します。

allow オプションは、プライマリ フィルタリング サーバが利用できないときに、FWSM がフィルタリングせずに FTP トラフィックを転送するようにします。

interact-block オプションを指定すると、完全なディレクトリ パスが提供されないインタラクティブ FTP セッションが阻止されます。インタラクティブ FTP クライアントにより、完全なパスを入力しなくてもディレクトリを変更できます。たとえば、**cd /public/files** ではなく、**cd ./files** を入力します。

フィルタリングの統計情報とフィルタリング設定の表示

ここでは、フィルタリングの統計情報をモニタする手順について説明します。次の内容について説明します。

- フィルタリング サーバの統計情報の表示（p.16-11）
- バッファ設定とバッファ統計情報の表示（p.16-11）
- キャッシングの統計情報の表示（p.16-12）
- フィルタリング パフォーマンスの統計情報の表示（p.16-12）
- フィルタリング設定の表示（p.16-12）

フィルタリング サーバの統計情報の表示

フィルタリング サーバに関する情報を表示するには、次のコマンドを入力します。

```
hostname# show running-config url-server
```

次に、**show url-server** コマンドの出力例を示します。

```
hostname# show running-config url-server
url-server (outside) vendor n2h2 host 128.107.254.202 port 4005 timeout 5 protocol TCP
```

フィルタリング サーバの情報または統計情報を表示するには、次のコマンドを入力します。

次に、フィルタリングの統計情報を表示する **show url-server** コマンドの出力例を示します。

```
hostname# show url-server
URL Server Statistics:
----------------------
Vendor                           websense
URLs total/allowed/denied        50/35/15
HTTPSs total/allowed/denied      1/1/0
FTPs total/allowed/denied        3/1/2

URL Server Status:
------------------
10.130.28.18            UP

URL Packets Sent and Received Stats:
------------------------------------
Message                 Sent    Received
STATUS_REQUEST          65155   34773
LOOKUP_REQUEST          0       0
LOG_REQUEST             0       NA
------------------------------------
```

バッファ設定とバッファ統計情報の表示

show running-config url-block コマンドは、url-block バッファで保持されるパケット数と、バッファ制限を超えた場合または再送信が発生した場合に廃棄される数（存在する場合）を示します。

次に、**show running-config url-block** コマンドの出力例を示します。

```
hostname# show running-config url-block
    url-block url-mempool 128
    url-block url-size 4
    url-block block 128
```

この出力では、URL ブロック バッファの設定が表示されています。

次に、**show url-block** コマンドの出力例を示します。

```
hostname# show url-block

URL Pending Packet Buffer Stats with max block  128
-----------------------------------------------------
Cumulative number of packets held:              896
Maximum number of packets held (per URL):       3
Current number of packets held (global):        38
Packets dropped due to
       exceeding url-block buffer limit:        7546
       HTTP server retransmission:              10
Number of packets released back to client:      0
```

この出力では、URL ブロックの統計情報が表示されています。

キャッシングの統計情報の表示

次に、**show url-cache** コマンドの出力例を示します。

```
hostname# show url-cache
URL Filter Cache Stats
----------------------
    Size :     128KB
 Entries :      1724
  In Use :       456
 Lookups :        45
    Hits :         8
```

この出力では、キャッシュがどのように使用されているかが表示されています。

フィルタリング パフォーマンスの統計情報の表示

次に、**show perfmon** コマンドの出力例を示します。

```
hostname# show perfmon
PERFMON STATS:    Current      Average
Xlates               0/s          0/s
Connections          0/s          2/s
TCP Conns            0/s          2/s
UDP Conns            0/s          0/s
URL Access           0/s          2/s
URL Server Req       0/s          3/s
TCP Fixup            0/s          0/s
TCPIntercept         0/s          0/s
HTTP Fixup           0/s          3/s
FTP Fixup            0/s          0/s
AAA Authen           0/s          0/s
AAA Author           0/s          0/s
AAA Account          0/s          0/s
```

この出力では、URL フィルタリング パフォーマンスの統計情報、および他のパフォーマンスの統計情報が表示されています。フィルタリング統計情報は、URL Access および URL Server Req の行に表示されています。

フィルタリング設定の表示

次に、**show running-config filter** コマンドの出力例を示します。

```
hostname# show running-config filter
filter url http 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0
```