CHAPTER 18

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。この章では、次の内容について説明します。

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

アクセスを全面的にブロックする代わりに、ActiveX オブジェクトや Java アプレットなど、特定の状況でセキュリティ上の脅威をもたらす可能性のある特定の不適切なオブジェクトを HTTP トラフィックから取り除くことができます。

URL フィルタリングを使用して、Secure Computing SmartFilter（従来の N2H2）や Websense などの外部フィルタリング サーバに特定のトラフィックを誘導することもできます。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ている特定のサイトまたは特定のタイプのサイトに向かう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 に大きな負荷がかかるため、外部フィルタリング サーバを使用することにより、他のトラフィックのスループットに影響を与えることがなくなります。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。ここでは、次の内容について説明します。

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 オブジェクトは、ActiveX フィルタリングでディセーブルにできます。

ActiveX コントロールは、以前は OLE コントロールまたは OCX コントロールと呼ばれていたもので、Web ページや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。ActiveX コントロールには、カスタム フォーム、カレンダ、その他さまざまな種類のサードパーティ フォームがあり、情報の収集や表示に使用されます。ActiveX テクノロジーは、ネットワーク クライアントにおいて、ワークステーションで障害が発生する原因となったり、ネットワーク セキュリティの問題を引き起こしたり、サーバの攻撃に使用されたりするなど、数多くの問題を引き起こす可能性があります。

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。<APPLET> ～ </APPLET> タグおよび <OBJECT CLASSID> ～ </OBJECT> タグを選択的にコメントに置換することによって、HTML ファイルの ActiveX フィルタリングが実行されます。最上位のタグをコメントに変換することによって、ネストされたタグのフィルタリングもサポートされます。

注意　このコマンドは、オブジェクト タグに埋め込まれている Java アプレット、イメージ ファイル、またはマルチメディア オブジェクトもすべてブロックします。

<object> または </object> という HTML タグが複数のネットワーク 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 MTU 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 ポートで *port* を置き換えます。一般的に、これはポート 80 ですが、他の値も受け入れられます。ポート 80 には、**http** または **url** リテラルを使用できます。ポート範囲を指定するには、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。

（注）　フィルタリングの例外ルールは、デフォルト ポートを使用している場合にだけ有効です。

ローカル IP アドレスおよびマスクによって、フィルタリングされるトラフィックの発信元である 1 台以上の内部ホストを指定します。外部アドレスおよびマスクは、フィルタリングされるトラフィックの外部の宛先を指定します。

これらのアドレスに **0.0.0.0**（短縮形は **0**）を設定して、すべてのホストを指定できます。これらのマスクに **0.0.0.0**（短縮形は **0**）を使用して、すべてのホストを指定できます。

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、任意のローカル ホストから任意の外部ホストへの接続において、ポート 80 で Web トラフィックに対して ActiveX オブジェクト ブロッキングを適用することを指定します。

このコンフィギュレーションを削除するには、次の例で示すように、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、保護されているネットワーク上のホストやサーバを攻撃することを目的とするコードが含まれている場合があるため、セキュリティのリスクが発生する可能性があります。Java アプレットは、**filter java** コマンドを使用して削除できます。

filter java コマンドは、発信接続から FWSM に返される Java アプレットをフィルタリングします。フィルタリングされてもユーザは HTML ページを受信できますが、アプレットの Web ページ ソースはコメントアウトされているため、アプレットは実行できません。

（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 ポートで *port* を置き換えます。一般的に、これはポート 80 ですが、他の値も受け入れられます。ポート 80 には、**http** または **url** リテラルを使用できます。ポート範囲を指定するには、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。

（注）　フィルタリングの例外ルールは、デフォルト ポートを使用している場合にだけ有効です。

ローカル IP アドレスおよびマスクによって、フィルタリングされるトラフィックの発信元である 1 台以上の内部ホストを指定します。外部アドレスおよびマスクは、フィルタリングされるトラフィックの外部の宛先を指定します。

これらのアドレスに **0.0.0.0**（短縮形は **0**）を設定して、すべてのホストを指定できます。これらのマスクに **0.0.0.0**（短縮形は **0**）を使用して、すべてのホストを指定できます。

これらのアドレスに **0.0.0.0**（短縮形は **0**）を設定して、すべてのホストを指定できます。これらのマスクに **0.0.0.0**（短縮形は **0**）を使用して、すべてのホストを指定できます。

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、任意のローカル ホストから任意の外部ホストへの接続において、ポート 80 で Web トラフィックに対して Java アプレット ブロッキングを適用することを指定します。

次に、保護されているネットワーク上のホストへの Java アプレットのダウンロードをブロックする例を示します。

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

このコマンドでは、ホスト 192.168.3.3 での Java アプレットのダウンロードが禁止されます。

このコンフィギュレーションを削除するには、次の例で示すように、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

この項では、外部サーバを使用して URL および FTP 要求をフィルタリングする方法について説明します。ここでは、次の内容について説明します。

URL フィルタリングの概要

フィルタリングは、セキュリティの高いネットワークからセキュリティの低いネットワークに発信される接続要求に対して適用できます。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動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、および長い URL のフィルタリング用 Websense Enterprise
- HTTP および HTTPS のフィルタリング用 Secure Computing SmartFilter（旧 N2H2）

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

コンテキストごとに最大 4 つのフィルタリング サーバを指定できます。FWSM は、1 つのサーバが応答するまで、それらの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** コマンドを使用して HTTP または HTTPS のフィルタリングを設定する前に、フィルタリング サーバを追加する必要があります。また、コンフィギュレーションからフィルタリング サーバを削除する前に、フィルタリング コマンドもすべて削除する必要があります。

url-server コマンドを次のように使用して、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) vendor websense host local_ip [timeout seconds]
[protocol {TCP | UDP | connections num_conns] | version 4][context-name]
```

（注）　**context-name** オプションは Websense バージョン 4.0 でだけ使用でき、バージョン 1.0 では使用できません。また、この機能はマルチコンテキスト モードでだけ設定できます。

Secure Computing SmartFilter（従来の N2H2）の場合は次のとおりです。

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

<*if_name*> は、フィルタリング サーバに接続されたセキュリティ アプライアンス インターフェイスの名前です。

vendor {smartfilter | n2h2} には、ベンダー文字列として「smartfilter」を使用できます。ただし、「n2h2」も下位互換性のために許容されます。コンフィギュレーション エントリが生成されると、「smartfilter」がベンダー文字列として保存されます。

host <*local_ip*> には、URL フィルタリング サーバの IP アドレスを指定します。

port <*number*> には、フィルタリング サーバの Secure Computing SmartFilter サーバ ポート番号を指定します。また、FWSM は、このポートの UDP 応答をリッスンします。

（注）　デフォルト ポートは 4005 です。これは、Secure Computing SmartFilter サーバが TCP または UDP で FWSM と通信するために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。

timeout <*seconds*> は、セキュリティ アプライアンスがフィルタリング サーバへの接続試行を継続する秒数です。

connections <*number*> は、ホストとサーバの間で接続を試行する回数です。

Context-name は、Websense サーバでのポリシー検索用の各 Websense クエリーを送信します。

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
```

これは、FWSM の境界インターフェイス上の、IP アドレス 10.0.1.1 を持つ Websense フィルタリング サーバを指定しています。この例でイネーブルになっている version 4 は、キャッシュをサポートするため、Websense によって推奨されています。

冗長 Secure Computing SmartFilter 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
```

これは、2 つの Sentian フィルタリング サーバを指定しています。いずれも FWSM の境界インターフェイス上にあります。

コンテンツ サーバ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した場合、その要求は、FWSM によって、コンテンツ サーバとフィルタリング サーバの両方に同時に送信されます。フィルタリング サーバがコンテンツ サーバより早く応答しなかった場合、サーバ応答はドロップされます。これにより、Web クライアント側の視点で Web サーバ応答が表示されます。これは、クライアントが要求を再発行する必要があるためです。

HTTP 応答バッファをイネーブルにすると、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、バッファリングしない場合に発生する可能性のある遅延が回避されます。

HTTP 要求または FTP 要求に対する応答のバッファリングを設定するには、次の手順を実行します。

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）　1,159 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 サーバでだけ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すると、フィルタリング サーバは FWSM に対して、サーバ アドレスを一定時間キャッシュすることを許可できます。ただし、そのアドレスでホストされているサイトはいずれも、常に許可されるカテゴリに属している必要があります。これにより、そのユーザがそのサーバに再度アクセスするか、別のユーザがそのサーバにアクセスしたときに、FWSM がフィルタリング サーバに再度照会する必要がなくなります。

（注）　キャッシュされた IP アドレス要求は、フィルタリング サーバに渡されず、記録もされません。そのため、このアクティビティはどのレポートにも表示されません。**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、**url-cache** コマンドを使用して、次のように入力します。

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

範囲 1 ～ 128（KB）のキャッシュ サイズの値を、*size* に指定します。

dst キーワードを使用して、URL 宛先アドレスに基づいて、エントリをキャッシュします。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。

src_dst キーワードを使用して、URL 要求を開始した送信元アドレスと 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します。このモードは、Websense サーバ上でユーザが同じ URL フィルタリング ポリシーを共有しない場合に選択します。

HTTP URL のフィルタリング

この項では、外部フィルタリング サーバを使用する HTTP フィルタリングを設定する方法について説明します。ここでは、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.18-7）
- 「長い HTTP URL のフィルタリングのイネーブル化」（P.18-8）
- 「長い HTTP URL の短縮」（P.18-8）
- 「フィルタリングから除外するトラフィックの指定」（P.18-8）

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 サーバを指定し、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 接続要求を承認した場合、FWSM は Web サーバからの応答が発信元クライアントに到達することを許可します。フィルタリング サーバが要求を拒否した場合、FWSM は、ユーザをブロック ページにリダイレクトし、アクセスが拒否されたことを示します。

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][cgi-truncate][longurl-deny][longurl-truncate][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は、1 つ以上のポート番号を、*port* に指定します。*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。

（注）　フィルタリングの例外ルールは、デフォルト ポートを使用している場合にだけ有効です。

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。**proxy-block** コマンドを使用して、プロキシ サーバへの要求をすべてドロップします。

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 HTTP URL を長い URL と見なします。Websense サーバの場合、最大長を増加できます。

（Websense だけ）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 だけ）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

デフォルトでは、URL が最大許容サイズを超えると、その URL はドロップされます。これを回避するには、次のコマンドを入力して、長い URL を切り捨てるように FWSM を設定します。

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

longurl-truncate オプションを指定すると、FWSM では、URL が許可されている最大長よりも長い場合、URL のホスト名または IP アドレスの部分のみがフィルタリング サーバに送信されて、評価されます。URL が許可されている最大長よりも長い場合に発信 URL トラフィックを拒否するには、**longurl-deny** オプションを使用します。

CGI URL を切り捨てて、CGI スクリプトの場所とスクリプト名のみを含め、すべてのパラメータを削除するには、**cgi-truncate** オプションを使用します。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の指定

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、次のコマンドは、10.0.2.54 からの HTTP 要求を除くすべての HTTP 要求がフィルタリング サーバに転送されるように設定しています。

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

（注）　**filter java except** コマンドを設定しており、同じ送信元/宛先ペアに対して **filter activex** コマンドを設定している場合、この送信元/宛先ペアのポート 80 ではフィルタリングは実行されません。

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 サーバを指定し、イネーブルにする必要があります。

HTTPS の内容は暗号化されるため、FWSM はディレクトリ情報およびファイル名情報なしで URL ルックアップを送信します。フィルタリング サーバが HTTPS 接続要求を承認した場合、FWSM は SSL 接続ネゴシエーションの完了を許可し、Web サーバからの応答が発信元クライアントに到達することを許可します。フィルタリング サーバが要求を拒否した場合、FWSM は SSL 接続ネゴシエーションの完了を許可しません。ブラウザには、「The Page or the content cannot be displayed.」のようなエラー メッセージが表示されます。

（注）　FWSM は、HTTPS 用の認証プロンプトを表示しないため、ユーザは HTTPS サーバにアクセスする前に、HTTP または FTP を使用して FWSM で認証を受ける必要があります。

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。フィルタリングの例外ルールは、デフォルト ポートを使用している場合にだけ有効です。

（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 サーバを指定し、イネーブルにする必要があります。

（注）　Secure Computing SmartFilter（旧 N2H2）では、FTP フィルタリングはサポートされていません。

フィルタリング サーバが FTP 接続要求を承認した場合、FWSM は、成功を示す FTP リターン コードが発信元クライアントに到達することを許可します。たとえば、成功を示すリターン コードは「250: CWD command successful」です。フィルタリング サーバが要求を拒否した場合、FTP リターン コードは接続が拒否されたことを示すように変更されます。たとえば、FWSM の場合、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。

（注）　フィルタリングの例外ルールは、デフォルト ポートを使用している場合にだけ有効です。

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。

完全なディレクトリ パスを提供しない対話型の FTP セッションをブロックするには、**interact-block** オプションを使用します。対話形式の FTP クライアントを使用すると、ユーザは、完全なパスを入力しないでディレクトリを変更できます。たとえば、ユーザは、**cd /public/files** ではなく、**cd ./files** と入力できます。

フィルタリング統計情報とフィルタリング設定の表示

この項では、フィルタリング統計情報をモニタする方法について説明します。ここでは、次の内容について説明します。

フィルタリング サーバ統計情報の表示

フィルタリング サーバの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 running-config 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。

```
hostname# show url-server statistics
```

次に、**show url-server statistics** コマンドの出力例を示します。このコマンドでは、フィルタリング統計情報が表示されます。

```
hostname# show url-server statistics
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 コンフィギュレーションと統計情報の表示

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
```

URL ブロック バッファのコンフィギュレーションが表示されています。

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

```
hostname# show url-block block statistics

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
```

これは、URL ブロック統計情報を示しています。

キャッシュ統計情報の表示

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

This shows how the cache is used.
```

フィルタリング性能統計情報の表示

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
```

これは、URL フィルタリング性能統計情報とその他の性能統計情報を示しています。フィルタリング統計情報は URL Access 行および URL Server Req 行に表示されます。

フィルタリング コンフィギュレーションの表示

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
```