侵入ルールの理解と作成

*侵入ルール*は特定のキーワードと引数のセットです。これを使用すると、ネットワーク トラフィックを分析してそれがルール内の基準を満たしているかどうか検査することにより、ネットワークの脆弱性を悪用しようとする試みを検出できます。システムは各ルールで指定された条件をパケットに照らし合わせます。ルールで指定されたすべての条件にパケット データが一致する場合、ルールがトリガーされます。ルールが *アラート ルール*である場合は、侵入イベントが生成されます。*パス ルール*である場合は、トラフィックが無視されます。侵入イベントは、防御センターの Web インターフェイスから表示して評価できます。

注意 作成した侵入ルールを実稼働環境で使用する前に、制御されたネットワーク環境で必ずテストしてください。不適切に作成された侵入ルールは、システムのパフォーマンスに重大な影響を与える可能性があります。

次の点に注意してください。

- インライン展開の *ドロップ ルール*では、システムがパケットを破棄してイベントを生成します。廃棄ルールの詳細については、ルール状態の設定（32-23 ページ）を参照してください。
- シスコ は、2 つのタイプの侵入ルール 共有オブジェクトのルール と 標準テキスト ルール を提供します。シスコ 脆弱性調査チーム（VRT）は 共有オブジェクトのルール を使用することで、従来の 標準テキスト ルール では不可能な方法で脆弱性に対する攻撃を検出できます。共有オブジェクトのルール を作成することはできません。独自の侵入ルールを作成するときには、標準テキスト ルール を作成します。

発生する可能性のあるイベントのタイプを調整するために、カスタム 標準テキスト ルール を作成することができます。このマニュアルでは特定のエクスプロイトの検出を目的とするルールについて説明することもありますが、優秀なルールのほとんどは、特定の既知のエクスプロイトではなく既知の脆弱性を悪用しようとするトラフィックをターゲットとすることに注意してください。ルールを作成してルールのイベント メッセージを指定することにより、攻撃とポリシー回避を示唆するトラフィックをより簡単に識別できます。イベントの評価の詳細については、侵入イベントの操作（41-1 ページ）を参照してください。

カスタム侵入ポリシーでカスタム標準テキスト ルールを有効にすると、一部のルール キーワードと引数では、トラフィックを特定の方法で最初に復号化または前処理する必要があることに留意してください。この章では、前処理を制御するネットワーク分析ポリシーで設定する必要があるオプションについて説明します。注意点として、必要なプリプロセッサを無効にすると、システムは自動的に現在の設定でプリプロセッサを使用します。ただし、ネットワーク分析ポリシーの Web インターフェイスではプリプロセッサは無効のままになります。

(注)　前処理と侵入インスペクションは非常に密接に関連しているため、単一パケットを検査するネットワーク分析ポリシーと侵入ポリシーは、相互補完する**必要があります**。前処理の調整、特に複数のカスタム ネットワーク分析ポリシーを使用して調整することは、**高度な**タスクです。詳細については、カスタム ポリシーに関する制約事項（23-13 ページ）を参照してください。

詳細については、次の各項を参照してください。

- ルール構造について（36-2 ページ）では、ルール ヘッダーやルール オプションなど、有効な標準テキスト ルール を構成するコンポーネントについて説明します。
- ルール ヘッダーについて（36-3 ページ）では、ルール ヘッダーの内容について詳しく説明します。
- ルールでのキーワードと引数について（36-11 ページ）では、FireSIGHT システムで使用可能な侵入ルール キーワードの使い方と構文について説明します。
- ルールの構築（36-116 ページ）では、ルール エディタを使用して新しいルールを作成する方法を説明します。
- ルールの検索（36-121 ページ）では、既存のルールの検索方法について説明します。
- [ルール エディタ（Rule Editor）] ページでのルールのフィルタリング（36-123 ページ）では、特定のルールを見つけやすくするためにルールのサブセットを表示する方法について説明します。

ルール構造について

ライセンス：Protection

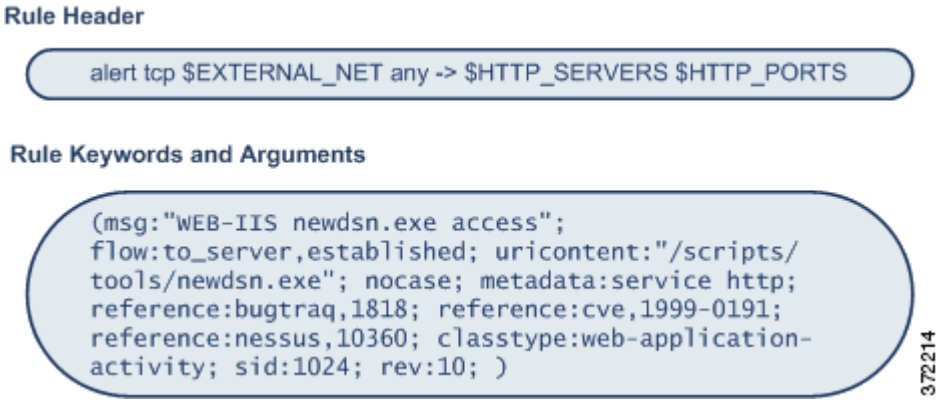
すべての 標準テキスト ルール には、ルール ヘッダーとルール オプションという 2 つの論理セクションが含まれています。ルール ヘッダーの内容は次のとおりです。

- ルールのアクションまたはタイプ
- プロトコル
- 送信元および宛先の IP アドレスとネットマスク
- 送信元から宛先へのトラフィック フローを示す方向インジケータ
- 送信元ポートと宛先ポート

ルール オプション セクションの内容は次のとおりです。

- イベント メッセージ
- キーワードとそのパラメータおよび引数
- ルールをトリガーとして使用するためにパケットのペイロードが一致する必要があるパターン
- パケットのどの部分をルール エンジンで検査するかの指定

次の図に、ルールの構成要素を示します。

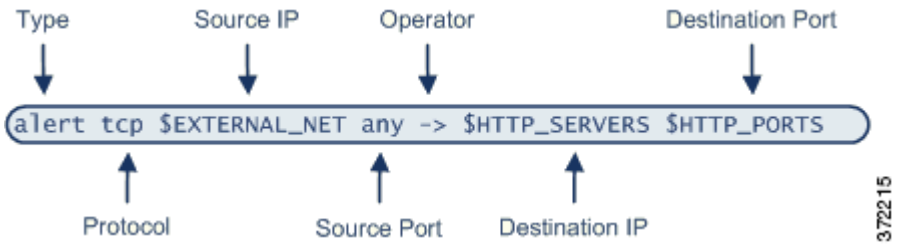


ルールのオプション セクションは、カッコで囲まれたセクションであることに注意してください。ルール エディタは、標準テキスト ルールの作成を支援する使いやすいインターフェイスを備えています。

ルール ヘッダーについて

ライセンス：Protection

それぞれの 標準テキスト ルール と 共有オブジェクトのルール には、パラメータと引数からなるルール ヘッダーが含まれています。ルール ヘッダーの構成要素を以下に示します。



次の表では、上記のルール ヘッダーの各部分について説明します。

表 36-1　ルール ヘッダーの値

ルール ヘッダーのコンポーネント	値の例	機能
操作	`alert`	トリガー時に侵入イベントを生成します。
プロトコル	`tcp`	TCP トラフィックのみをテストします。
送信元 IP アドレス	`$EXTERNAL_NET`	内部ネットワーク上に存在しないホストから送られてきたトラフィックをテストします。
送信元ポート	任意	発信元ホスト上の任意のポートから送られてきたトラフィックをテストします。
演算子	`->`	（このネットワーク上の Web サーバに向かう）外部トラフィックをテストします。

表 36-1　ルール ヘッダーの値（続き）

ルール ヘッダーのコンポーネント	値の例	機能
宛先 IP アドレス	$HTTP_SERVERS	この内部ネットワーク上の Web サーバとして指定された任意のホストに送られるトラフィックをテストします。
宛先ポート	$HTTP_PORTS	この内部ネットワーク上の HTTP ポートに送られるトラフィックをテストします。

(注)　前述の例では、ほとんどの侵入ルールの場合と同様に、デフォルト変数が使用されています。変数のリスト、機能、および設定方法の詳細については、変数セットの使用（3-19 ページ）を参照してください。

ルール ヘッダー パラメータの詳細については、以下の項を参照してください。

- ルール アクションの指定（36-4 ページ）では、ルール タイプについて説明し、ルールのトリガー時に実行されるアクションを指定する方法について説明します。
- プロトコルの指定（36-5 ページ）では、ルールによるテスト対象となるトラフィックのトラフィック プロトコルを定義する方法について説明します。
- 侵入ルールでの IP アドレスの指定（36-5 ページ）では、ルール ヘッダーで個別の IP アドレスと IP アドレス ブロックを定義する方法について説明します。
- 侵入ルールでのポートの定義（36-9 ページ）では、ルール ヘッダーで個別のポートとポート範囲を定義する方法について説明します。
- 方向の指定（36-10 ページ）では、使用可能な演算子について説明し、ルールでテストすべきトラフィック伝送方向を指定する方法について説明します。

ルール アクションの指定

ライセンス：Protection

各ルール ヘッダーには、パケットがルールをトリガーとして使用したときにシステムで行われるアクションを指定するパラメータが 1 つ含まれています。アクションが *alert* に設定されたルールは、それをトリガーとして使用したパケットに対する侵入イベントを生成し、そのパケットの詳細をログに記録します。アクションが *pass* に設定されたルールは、それをトリガーとして使用したパケットに関するイベントを生成せず、そのパケットの詳細も記録しません。

(注)　インライン展開において、ルール状態が [ドロップしてイベントを生成する（Drop and Generate Events）] に設定されたルールは、それをトリガーとして使用したパケットに対する侵入イベントを生成します。また、パッシブ展開で廃棄ルールを適用した場合は、ルールがアラート ルールとして機能します。廃棄ルールの詳細については、ルール状態の設定（32-23 ページ）を参照してください。

デフォルトでは、パス ルールがアラート ルールをオーバーライドします。パス ルールを作成することで、アラート ルールを無効にする代わりに、パス ルールで定義された基準を満たすパケットが特定の状況でアラート ルールをトリガーとして使用しないことを指定できます。たとえば、ユーザ "anonymous" として FTP サーバにログインする試行を検索するルールをアクティブのままにする必要があるとします。ただし、1 つ以上の正式な匿名 FTP サーバがネットワークに存在する場合、そのような特定のサーバで匿名ユーザにより最初のルールがトリガーとして使用されないことを指定するパス ルールを作成し、アクティブにすることができます。

ルール エディタで、[アクション（Action）] リストからルール タイプを選択します。ルール エディタを使ってルール ヘッダーを作成する手順の詳細については、ルールの構築（36-116 ページ）を参照してください。

プロトコルの指定

ライセンス：Protection

各ルール ヘッダーで、ルールにより検査されるトラフィックのプロトコルを指定する必要があります。次のネットワーク プロトコルを分析対象として指定できます。

- ICMP（Internet Control Message Protocol）
- インターネット プロトコル（IP）

（注） プロトコルが ip に設定されている場合、システムは侵入ルール ヘッダー内のポート定義を無視します。詳細については、侵入ルールでのポートの定義（36-9 ページ）を参照してください。

- 伝送制御プロトコル（TCP）
- ユーザ データグラム プロトコル（UDP）

TCP、UDP、ICMP、IGMP など、IANA によって割り当てられたすべてのプロトコルを検査するには、プロトコル タイプとして **IP** を使用します。IANA によって割り当てられたプロトコルの完全なリストについては、http://www.iana.org/assignments/protocol-numbers を参照してください。

（注） 現在のところ、IP ペイロード内の次のヘッダー（TCP ヘッダーなど）でパターンを照合するルールを作成することはできません。代わりに、最後にデコードされたプロトコルからコンテンツ照合が始まります。次善策として、ルール オプションを使用して TCP ヘッダー内のパターンを照合できます。

ルール エディタで、[プロトコル（Protocol）] リストからプロトコル タイプを選択します。ルール エディタを使用してルール ヘッダーを作成する手順の詳細については、ルールの構築（36-116 ページ）を参照してください。

侵入ルールでの IP アドレスの指定

ライセンス：Protection

パケット検査の対象を、特定の IP アドレスから発信されたパケットまたは特定の IP アドレスに向かうパケットに制限すると、システムが実行しなければならないパケット検査の量が減ります。さらに、ルールをより具体化し、送信元および宛先 IP アドレスが疑わしい動作を示していないパケットに対してルールがトリガーとして使用される可能性をなくすと、誤検出も減ります。

ヒント　システムは IP アドレスのみを認識し、送信元/宛先 IP アドレスのホスト名を受け入れません。

ルール エディタの [送信元 IP(Source IPs)] フィールドと [宛先 IP(Destination IPs)] フィールドで、送信元および宛先の IP アドレスを指定します。ルール エディタを使用してルール ヘッダーを作成する手順の詳細については、ルールの構築(36-116 ページ)を参照してください。

標準テキスト ルール の作成時には、必要に応じて、さまざまな方法で IPv4 アドレスと IPv6 アドレスを指定できます。単一の IP アドレス、any(オプション)、IP アドレス リスト、CIDR 表記、プレフィクス長、ネットワーク変数、またはネットワーク オブジェクトあるいはネットワーク オブジェクト グループを指定できます。加えて、1 つの特定の IP アドレスまたは IP アドレスのセットを除外するよう指定できます。IPv6 アドレスを指定するときには、RFC 4291 で定義された任意のアドレス指定規則を使用できます。

次の表では、送信元と宛先の IP アドレスを指定するさまざまな方法を要約します。

表 36-2　送信元/宛先 IP アドレスの構文

指定する項目	使用するフィルタ	例
任意の IP アドレス	任意	任意
1 つの特定の IP アドレス	IP アドレス 同じルール内に IPv4 と IPv6 の送信元アドレスと宛先アドレスを混在させないでください。	192.168.1.1 2001:db8::abcd
IP アドレスのリスト	複数の IP アドレスをカンマで区切り、それを大カッコ([])で囲む	[192.168.1.1,192.168.1.15] [2001:db8::b3ff, 2001:db8::0202]
IP アドレスのブロック	IPv4 CIDR ブロックまたは IPv6 アドレス プレフィクス表記	192.168.1.0/24 2001:db8::/32
特定の 1 つの IP アドレスまたはアドレス セットを除くすべて	拒否する IP アドレスの前に付ける「!」記号	!192.168.1.15 !2001:db8::0202:b3ff:fe1e
特定の 1 つ以上の IP アドレスを除く、IP アドレス ブロック内のすべて	アドレス ブロックの後に、除外アドレスのリストまたはブロック	[10.0.0/8, !10.2.3.4, !10.1.0.0/16] [2001:db8::/32, !2001:db8::8329, !2001:db8::0202]
ネットワーク変数で定義された IP アドレス	$ で始まる大文字の変数名 プリプロセッサ ルールは、侵入ルールで使われているネットワーク変数で定義されたホストとは無関係に、イベントをトリガーできることに注意してください。詳細については、変数セットの使用(3-19 ページ)を参照してください。	$HOME_NET
IP アドレス変数で定義されたアドレスを除く、すべての IP アドレス	大文字の変数名の前に !$ を付ける 詳細については、侵入ルールにおける IP アドレスの除外(36-8 ページ)を参照してください。	!$HOME_NET

表 36-2　送信元/宛先 IP アドレスの構文（続き）

指定する項目	使用するフィルタ	例
ネットワーク オブジェクトまたはネットワーク オブジェクト グループで定義された IP アドレス	`!{object_name}` という形式でオブジェクト名またはグループ名。 詳細については、ネットワーク オブジェクトの操作（3-4 ページ）を参照してください。	`${192.168sub16}`
ネットワーク オブジェクトまたはネットワーク オブジェクト グループで定義されたアドレスを除く、すべての IP アドレス	オブジェクト名またはグループ名を中カッコ（{}）で囲み、その前に `!$` を付ける。 詳細については、ネットワーク オブジェクトの操作（3-4 ページ）を参照してください。	`!${192.168sub16}`

送信元や宛先の IP アドレスの指定に使用できる構文の詳細、および変数を使って IP アドレスを指定する方法については、以下の項を参照してください。

- IP アドレスの表記規則（1-24 ページ）。
- 変数セットの使用（3-19 ページ）
- 任意の IP アドレスの指定（36-7 ページ）
- 複数の IP アドレスの指定（36-7 ページ）
- ネットワーク オブジェクトの指定（36-8 ページ）
- 侵入ルールにおける IP アドレスの除外（36-8 ページ）

任意の IP アドレスの指定

ライセンス：Protection

任意の IPv4 または IPv6 アドレスを示す「`any`」という単語を、ルールの送信元 IP アドレスまたは宛先 IP アドレスとして指定できます。

たとえば、次のルールでは [Source IPs] フィールドと [Destination IPs] フィールドで引数 **any** を使用して、任意の IPv4 または IPv6 の送信元または宛先アドレスを持つパケットを評価します。

```
alert tcp any any -> any any
```

また、任意の IPv6 アドレスを示すために `::` を指定することもできます。

複数の IP アドレスの指定

ライセンス：Protection

次の例に示すように、複数の IP アドレスをカンマで区切ることで、個別の IP アドレスを列挙できます。必要に応じて、非拒否リストを大カッコで囲むこともできます。

```
[192.168.1.100,192.168.1.103,192.168.1.105]
```

IPv4 アドレスと IPv6 アドレスのいずれかだけを列挙することも、任意に組み合わせて列挙することもできます（次の例を参照）。

```
[192.168.1.100,2001:db8::1234,192.168.1.105]
```

以前のソフトウェア リリースでは IP アドレス リストを大カッコで囲む必要がありましたが、現在ではこれが必須でないことに注意してください。また、オプションで、リストを入力するときに各カンマの前または後にスペースを含めることができます。

(注)　否定リストは、大カッコで囲む必要があります。詳細については、侵入ルールにおける IP アドレスの除外(36-8 ページ)を参照してください。

また、IPv4 クラスレス ドメイン間ルーティング(CIDR)表記または IPv6 プレフィクス長を使用して、アドレス ブロックを指定することもできます。次に例を示します。

- 192.168.1.0/24 は、サブネット マスク 255.255.255.0 の 192.168.1.0 ネットワーク内の IPv4 アドレス、つまり 192.168.1.0 ～ 192.168.1.255 を指定します。詳細については、IP アドレスの表記規則(1-24 ページ)を参照してください。
- 2001:db8::/32 は、プレフィクス長 32 ビットの 2001:db8:: ネットワーク内の IPv6 アドレス、つまり 2001:db8:: ～ 2001:db8:ffff:ffff:ffff:ffff:ffff:ffff を指定します。

ヒント　IP アドレスのブロックを指定する必要があるが、CIDR またはプレフィクス長表記を単独で使ってそれを表現できない場合は、1 つの IP アドレス リスト内でいくつかの CIDR ブロックとプレフィクス長を使用できます。

ネットワーク オブジェクトの指定

ライセンス: Protection

次の構文を使用して、ネットワーク オブジェクトまたはネットワーク オブジェクト グループを指定できます。

```
${object_name | group_name}
```

引数の説明

- *object_name* はネットワーク オブジェクトの名前です
- *group_name* はネットワーク オブジェクト グループの名前です

ネットワーク オブジェクトとネットワーク オブジェクト グループの作成方法については、ネットワーク オブジェクトの操作(3-4 ページ)を参照してください。

`192.168sub16` という名前のネットワーク オブジェクトと `all_subnets` という名前のネットワーク オブジェクト グループをすでに作成済みであるとします。ネットワーク オブジェクトを使用して IP アドレスを特定するには、たとえば次のように指定できます。

```
${192.168sub16}
```

ネットワーク オブジェクト グループを使用するには、次のように指定できます。

```
${all_subnets}
```

さらに、ネットワーク オブジェクトとネットワーク オブジェクト グループで否定を使用することもできます。次に例を示します。

```
!${192.168sub16}
```

詳細については、侵入ルールにおける IP アドレスの除外(36-8 ページ)を参照してください。

侵入ルールにおける IP アドレスの除外

ライセンス: Protection

特定の IP アドレスを否定するために感嘆符(!)を使用できます。つまり、1 つ以上の特定の IP アドレスを除く、すべての IP アドレスに一致させることができます。たとえば、`!192.168.1.1` は 192.168.1.1 以外の任意の IP アドレスを、`!2001:db8:ca2e::fa4c` は 2001:db8:ca2e::fa4c 以外の任意の IP アドレスを指定します。

一連の IP アドレスを拒否するには、大かっこで囲んだ IP アドレスのリストの前に「!」記号を付けます。たとえば、`![192.168.1.1,192.168.1.5]` は 192.168.1.1 と 192.168.1.5 を除くすべての IP アドレスを定義します。

(注)　IP アドレスのリストを否定するには、大カッコを使用する必要があります。

否定文字と一緒に IP アドレス リストを使用する場合は注意が必要です。たとえば、192.168.1.1 と 192.168.1.5 を除くすべてのアドレスと一致させるために `[!192.168.1.1,!192.168.1.5]` を使用した場合、システムはこの構文を「192.168.1.1 以外のすべて、**または** 192.168.1.5 以外のすべて」と解釈します。

192.168.1.5 は 192.168.1.1 ではなく、192.168.1.1 は 192.168.1.5 ではないため、この両方の IP アドレスが `[!192.168.1.1,!192.168.1.5]` という IP アドレス値に一致します。つまり、実質的に「any」を使用するのと同じです。

代わりに `![192.168.1.1,192.168.1.5]` を使用してください。システムはこの構文を「192.168.1.1 でなく、**しかも** 192.168.1.5 でない」と解釈し、大カッコ内に列挙されたものを除くすべての IP アドレスに一致します。

論理的に言って、any を除外（negation）と同時に使用できないことに注意してください。any を除外すると「アドレスなし」を意味することになります。

侵入ルールでのポートの定義

ライセンス：Protection

ルール エディタの [送信元ポート（Source Port）] フィールドと [宛先ポート（Destination Port）] フィールドで、送信元および宛先ポートを指定します。ルール エディタを使用してルール ヘッダーを作成する手順の詳細については、ルールの構築（36-116 ページ）を参照してください。

ルール ヘッダー内で使われるポート番号を定義するために、FireSIGHT システムは特殊なタイプの構文を使用します。

(注)　プロトコルが ip に設定されている場合、システムは侵入ルール ヘッダー内のポート定義を無視します。詳細については、プロトコルの指定（36-5 ページ）を参照してください。

次の例に示すように、カンマでポートを区切ることによって、ポートのリストを指定できます。

```
80, 8080, 8138, 8600-9000, !8650-8675
```

オプションで、次の例に示すように、ポート リストを大カッコで囲むこともできます（以前のソフトウェア バージョンではこれが必須でしたが、現在は必須ではありません）。

```
[80, 8080, 8138, 8600-9000, !8650-8675]
```

なお、次の例に示すように、ポート リストの否定を大カッコで囲む**必要がある**ことに注意してください。

```
![20, 22, 23]
```

また、侵入ルール内の送信元ポートや宛先ポートのリストには最大で 64 文字を含めることができます。

次の表に、使用可能な構文を要約します。

表 36-3　送信元/宛先ポートの構文

指定する項目	用途	例
任意のポート	任意	任意
1 つの特定のポート	ポート番号	80
ポートの範囲	範囲内の最初のポート番号と最後のポート番号をダッシュでつなぐ	80-443
1 つの特定のポートに等しい、またはより小さいすべてのポート	ポート番号の前にダッシュを付ける	-21
1 つの特定のポートに等しい、またはより大きいすべてのポート	ポート番号の後ろにダッシュを付ける	80-
1 つの特定のポートまたはポート範囲を除く、すべてのポート	拒否するポート、ポート リスト、またはポート範囲の前に「!」記号を付ける 論理的に言って、*any* を除くすべてのポート指定と一緒に否定を使用できます。any を否定すると「*ポートなし*」を意味することに注意してください。	!20
ポート変数で定義されるすべてのポート	大文字の変数名の前に、$ を付ける 詳細については、ポート変数の操作（3-33 ページ）を参照してください。	$HTTP_PORTS
ポート変数で定義されるポートを除く、すべてのポート	大文字の変数名の前に、!$ を付ける	!$HTTP_PORTS

方向の指定

ライセンス：Protection

ルールによる検査対象となるパケットが進むべき方向を、ルール ヘッダー内で指定できます。以下の表は、それらのオプションを示しています。

表 36-4　ルール ヘッダー内の方向オプション

使用するフィルタ	テスト対象
指向性	指定された送信元 IP アドレスから指定された宛先 IP アドレスに向かうトラフィックのみ
双方向	指定された送信元 IP アドレスと宛先 IP アドレスの間を移動するすべてのトラフィック

ルール エディタを使用してルール ヘッダーを作成する手順の詳細については、ルールの構築（36-116 ページ）を参照してください。

ルールでのキーワードと引数について

ライセンス：Protection

ルール言語では、キーワードを組み合わせることによってルールの動作を指定できます。キーワードとそれに関連する値（*引数*と呼ばれる）を使用して、ルール エンジンで検査されるパケットやパケット関連値をシステムが評価する方法を指定します。FireSIGHT システムでは現在、コンテンツ マッチング、プロトコル固有のパターン マッチング、状態固有のマッチングなど、インスペクション機能を実行するためのキーワードがサポートされています。キーワードあたり最大 100 個の引数を定義し、互換性のある任意の数のキーワードを組み合わせて非常に具体的なルールを作成できます。これにより、誤検出や検出漏れの可能性が減少し、受け取った侵入情報に集中的に取り組むことができます。

また、適応型プロファイルを使用すると、ルール メタデータとホスト情報に基づいて特定のパケットに対するアクティブ ルール処理を動的に調整できます。詳細については、パッシブ展開における前処理の調整（30-1 ページ）を参照してください。

詳細については、次の各項を参照してください。

- 侵入イベント詳細の定義（36-12 ページ）では、イベントのメッセージ、プライオリティ情報、およびルールで検出されたエクスプロイトに関する外部情報への参照を定義するためのキーワードの構文と使用法について説明します。
- コンテンツ一致の検索（36-16 ページ）では、`content` または `protected_content` キーワードを使用して、パケット ペイロードの内容を検査する方法について説明します。
- コンテンツ一致の制約（36-20 ページ）では、`content` または `protected_content` キーワードを変更するキーワードの使用方法について説明します。
- インライン展開でのコンテンツの置換（36-33 ページ）では、インライン展開で `replace` キーワードを使用して、長さの等しい指定されたコンテンツを置換する方法について説明します。
- Byte_Jump と Byte_Test の使用（36-34 ページ）では、`byte_jump` キーワードと `byte_test` キーワードを使用して、パケット内のどの位置でルール エンジンがコンテンツ マッチング検査を開始すべきか、どのバイトを評価すべきかについて計算する方法を説明します。
- PCRE を使用したコンテンツの検索（36-39 ページ）では、`pcre` キーワードを使用して、ルール内で Perl 互換の正規表現を使用する方法について説明します。
- ルールへのメタデータの追加（36-48 ページ）では、`metadata` キーワードを使用して、ルールに情報を追加する方法について説明します。
- IP ヘッダー値の検査（36-53 ページ）では、パケットの IP ヘッダー内の値を検査するキーワードの構文と使用法について説明します。
- ICMP ヘッダー値の検査（36-56 ページ）では、パケットの ICMP ヘッダー内の値を検査するキーワードの構文と使用法について説明します。
- TCP ヘッダー値とストリーム サイズの検査（36-57 ページ）では、パケットの TCP ヘッダー内の値を検査するキーワードの構文と使用法について説明します。
- TCP ストリーム再構築の有効化と無効化（36-62 ページ）では、接続での検査対象トラフィックがルールの条件と一致した場合に、単一接続のストリーム再構築を有効/無効にする方法について説明します。
- セッションからの SSL 情報の抽出（36-62 ページ）では、暗号化されたトラフィックからバージョン情報と状態情報を抽出するキーワードの使用法と構文について説明します。
- パケット データをキーワード引数の中に読み込む（36-92 ページ）では、パケットから変数の中に値を読み込み、あとでそれを同じルール内で使用することにより、その値を特定の他のキーワードの引数として指定する方法を説明します。

- アプリケーション層プロトコル値の検査（36-64 ページ）では、アプリケーション層プロトコル プロパティを検査するキーワードの使用法と構文について説明します。
- パケット特性の検査（36-89 ページ）では、dsize、sameIP、isdataat、fragoffset および cvs キーワードの使用法と構文について説明します。
- ルール キーワードを使用したアクティブ応答の開始（36-96 ページ）では、resp キーワードを使用して TCP 接続または UDP セッションをアクティブに閉じる方法、react キーワードを使用して HTML ページを送信した後で TCP 接続をアクティブに閉じる方法、および config response コマンドを使用してアクティブ応答インターフェイスとパッシブ展開での TCP リセット試行回数を指定する方法について説明します。
- イベントのフィルタリング（36-99 ページ）では、指定された時間内に指定されたパケット数がルールの検出基準を満たさない限り、ルールでイベントがトリガーとして使用されないようにする方法を説明します。
- 攻撃後トラフィックの評価（36-101 ページ）では、ホストまたはセッションに関する追加のトラフィックをログに記録する方法について説明します。
- 複数のパケットに及ぶ攻撃の検出（36-102 ページ）では、単一セッション内の複数パケットに及ぶ攻撃からパケットに状態名を割り当てた後、その状態に応じてパケットを分析および警告する方法について説明します。
- HTTP エンコードのタイプと位置によるイベントの生成（36-107 ページ）では、正規化の前に、HTTP 要求や応答 URI、ヘッダー、または（set-cookie を含む）cookie 内のエンコード タイプに基づいてイベントを生成する方法について説明します。
- ファイル タイプとバージョンの検出（36-109 ページ）では、file_type キーワードまたは file_group キーワードを使用して、特定のファイル タイプまたはファイル バージョンを指し示す方法について説明します。
- 特定のペイロード タイプを指し示す（36-112 ページ）では、HTTP 応答エンティティ本体、SMTP ペイロード、またはエンコードされた電子メール添付ファイルの先頭を指し示す方法について説明します。
- パケット ペイロードの先頭を指し示す（36-113 ページ）では、パケット ペイロードの先頭を指し示す方法について説明します。
- Base64 データのデコードと検査（36-114 ページ）では、base64_decode キーワードと base64_data キーワードを使用して、特に HTTP 要求内の Base64 データをデコードして検査する方法について説明します。

侵入イベント詳細の定義

ライセンス：Protection

標準テキスト ルール を作成するときには、ルールで攻撃試行を検出する対象となる脆弱性についてのコンテキスト情報を含めることができます。また、脆弱性データベースへの外部参照を含めたり、組織内でイベントに設定するプライオリティを定義したりすることもできます。アナリストがイベントを認識すると、そのプライオリティ、エクスプロイト、および既知の対策についての情報をすぐに入手できます。

イベント関連のキーワードの詳細については、以下の項を参照してください。

- イベント メッセージの定義（36-13 ページ）
- イベント プライオリティの定義（36-13 ページ）
- 侵入イベント分類の定義（36-13 ページ）
- イベント参照の定義（36-15 ページ）

イベント メッセージの定義

ライセンス：Protection

ルールのトリガー時にメッセージとして表示される、意味のあるテキストを指定できます。メッセージを読むと、ルールで攻撃試行を検出する対象となった脆弱性の特性をすぐに理解できます。中カッコ（{}）を除く、印字可能な任意の標準 ASCII 文字を使用できます。システムは、メッセージ全体を囲んでいる引用符を取り除きます。

ヒント　ルール メッセージの指定は必須です。また、空白文字のみ、1 つ以上の引用符のみ、1 つ以上のアポストロフィのみ、あるいは空白文字/引用符/アポストロフィだけの組み合わせでメッセージを構成することはできません。

ルール エディタでイベント メッセージを定義するには、[メッセージ（Message）] フィールドにイベント メッセージを入力します。ルール エディタを使用してルールを作成する方法については、ルールの構築（36-116 ページ）を参照してください。

イベント プライオリティの定義

ライセンス：Protection

デフォルトでは、ルールのイベント分類からルールのプライオリティが派生します。ただし、`priority` キーワードをルールに追加すると、ルールの分類プライオリティをオーバーライドできます。

ルール エディタを使ってプライオリティを指定するには、[検出オプション（Detection Options）] リストから [優先順位（priority）] を選択して、ドロップダウン リストから [高（high）]、[中（medium）]、または [低（low）] を選択します。たとえば、Web アプリケーション攻撃を検出するルールに **high** プライオリティを割り当てるには、`priority` キーワードをルールに追加して、プライオリティとして **high** を選択します。ルール エディタを使用してルールを作成する方法については、ルールの構築（36-116 ページ）を参照してください。

侵入イベント分類の定義

ライセンス：Protection

ルールごとに、イベントのパケット表示に含める攻撃分類を指定できます。次の表に、それぞれの分類の名前と番号を示します。

表 36-5　ルールの分類

番号（Number）	分類名	説明
1	not-suspicious	不審ではないトラフィック
2	unknown	不明なトラフィック
3	bad-unknown	有害な可能性のあるトラフィック
4	attempted-recon	情報漏えいが試行された
5	successful-recon-limited	情報漏えいが発生
6	successful-recon-largescale	大規模な情報漏えい

表 36-5　ルールの分類（続き）

番号（Number）	分類名	説明
7	attempted-dos	サービス妨害が試行された
8	successful-dos	サービス妨害（DoS）
9	attempted-user	ユーザ特権の獲得が試行された
10	unsuccessful-user	ユーザ特権の獲得が失敗した
11	successful-user	ユーザ特権の獲得に成功
12	attempted-admin	管理者特権の獲得が試行された
13	successful-admin	管理者特権の獲得に成功
14	rpc-portmap-decode	RPC クエリのデコード
15	shellcode-detect	実行可能コードが検出された
16	string-detect	疑わしい文字列が検出された
17	suspicious-filename-detect	疑わしいファイル名が検出された
18	suspicious-login	疑わしいユーザ名を使用したログイン試行が検出された
19	system-call-detect	システム コールが検出された
20	tcp-connection	TCP 接続が検出された
21	trojan-activity	ネットワーク トロイの木馬が検出された
22	unusual-client-port-connection	通常とは異なるポートをクライアントが使用していた
23	network-scan	ネットワーク スキャンの検出
24	denial-of-service	サービス妨害攻撃の検出
25	non-standard-protocol	非標準プロトコルまたはイベントの検出
26	protocol-command-decode	一般的なプロトコル コマンド デコード
27	web-application-activity	脆弱な可能性のある Web アプリケーションへのアクセス
28	web-application-attack	Web アプリケーション攻撃
29	misc-activity	その他のアクティビティ
30	misc-attack	その他の攻撃
31	icmp-event	一般的な ICMP イベント
32	inappropriate-content	不適切な内容が検出された
33	policy-violation	企業プライバシー侵害の可能性
34	default-login-attempt	デフォルトのユーザ名とパスワードによるログイン試行
35	sdf	機密データ
36	malware-cnc	既知のマルウェア コマンドと制御トラフィック

表 36-5　ルールの分類（続き）

番号（Number）	分類名	説明
37	client-side-exploit	既知のクライアント側エクスプロイト試行
38	file-format	既知の悪意のあるファイルまたはファイル ベースのエクスプロイト

ルール エディタで分類を指定するには、[分類（Classification）] リストから分類を 1 つ選択します。ルール エディタの詳細については、新しいルールの作成（36-116 ページ）を参照してください。

カスタム分類の追加

ライセンス：Protection

定義したルールによって生成されるイベントのパケット表示記述の内容をもっとカスタマイズする必要がある場合には、カスタム分類を作成します。

[分類（Classification）] リストに分類を追加するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ポリシー（Policies）] > [侵入（Intrusion）] > [ルール エディタ（Rule Editor）] の順に選択します。

[ルール エディタ（Rule Editor）] ページが表示されます。

手順 2　[ルールの作成（Create Rule）] をクリックします。

[ルールの作成（Create Rule）] ページが表示されます。

手順 3　[分類（Classification）] ドロップダウンリストで、[分類の編集（Edit Classifications）] をクリックします。

ポップアップ ウィンドウが表示されます。

手順 4　[分類名（Classification Name）] フィールドに分類の名前を入力します。

最大で 255 文字の英数字を使用できますが、40 文字を超えるとページが読みにくくなります。`<>()\'"&$;` 文字および空白文字はサポートされていません。

手順 5　[分類の説明（Classification Description）] フィールドに、分類の説明を入力します。

最大 255 文字の英数字およびスペースを使用できます。`<>()\'"&$;` 文字はサポートされていません。

手順 6　[プライオリティ（Priority）] リストからプライオリティを選択します。

[high]、[medium]、または [low] を選択できます。

手順 7　[追加（Add）] をクリックします。

新しい分類がリストに追加され、ルール エディタで使用できるようになります。

手順 8　[完了（Done）] をクリックします。

イベント参照の定義

ライセンス：Protection

reference キーワードを使用すると、イベントに関する外部 Web サイトや追加情報への参照を追加できます。参照を追加すると、アナリストは参照情報をすぐに利用できるため、パケットがルールをトリガーとして使用した理由を特定するのに役立ちます。次の表に、既知のエクスプロイトや攻撃についてのデータを提供する外部システムをいくつか示します。

表 36-6　外部攻撃識別システム

システム ID（System ID）	説明	ID の例
bugtraq	[Bugtraq] ページ	8550
cve	[Common Vulnerabilities and Exposure] ページ	CAN-2003-0702
mcafee	[McAfee] ページ	98574
URL	Web サイト参照	www.example.com?exploit=14
msb	Microsoft セキュリティ情報	MS11-082
nessus	[Nessus] ページ	10039
secure-url	セキュア Web サイト参照（https://...）	intranet/exploits/exploit=14 任意のセキュア Web サイトで secure-url を使用できることに注意してください。

ルール エディタを使用して参照を指定するには、[検出オプション（Detection Options）] リストから [参照（reference）] を選択し、対応するフィールドに次のように値を入力します。

```
id_system,id
```

ここで、*id_system* はプレフィクスとして使用されるシステム、*id* は Bugtraq ID、CVE 番号、Arachnids ID、または URL（http:// なし）です。

たとえば、Bugtraq ID 17134 に記載されている Microsoft Commerce Server 2002 サーバ上の認証バイパス脆弱性を指定するには、[参照（reference）] フィールドに次のように入力します。

```
bugtraq,17134
```

参照をルールに追加するときには、次の点に注意してください。

- カンマの後ろにスペースを入力しないでください。
- システム ID に大文字を使用しないでください。

ルール エディタを使用してルールを作成する方法については、ルールの構築（36-116 ページ）を参照してください。

コンテンツ一致の検索

ライセンス：Protection

content キーワードまたは protected_content キーワードを使用すると、パケット内から検出するコンテンツを指定できます。詳細については、次の各項を参照してください。

- content キーワードの使用（36-17 ページ）
- protected_content キーワードの使用（36-17 ページ）
- コンテンツ マッチングの設定（36-18 ページ）

content キーワードの使用

content キーワードを使用すると、ルール エンジンはパケット ペイロードまたはストリームでその文字列を検索します。たとえば、いずれかの content キーワードの値として /bin/sh と入力した場合、ルール エンジンはパケット ペイロード内で文字列 /bin/sh を検索します。

ASCII 文字列、16 進コンテンツ（バイナリ バイト コード）、またはその両方の組み合わせを使用してコンテンツを照合できます。キーワード値の中で 16 進コンテンツをパイプ文字（|）で囲みます。たとえば、|90C8 C0FF FFFF|/bin/sh のように 16 進コンテンツと ASCII コンテンツを混在させることができます。

1 つのルール内で複数のコンテンツ マッチングを指定できます。これを行うには、content キーワードの追加のインスタンスを使用します。コンテンツ マッチングごとに、ルールをトリガーとして使用させるにはパケット ペイロードまたはストリームでコンテンツ一致が見つからなければならないことを指定できます。

protected_content キーワードの使用

protected_content キーワードを使用すると、ルール引数を設定する前に、検索コンテンツ文字列をエンコードすることができます。キーワードを設定する前に、ルール作成者がハッシュ関数（SHA512、SHA256、または MD5）を使用して文字列をエンコードします。

content キーワードの代わりに protected_content キーワードを使用した場合でも、ルール エンジンがパケット ペイロードまたはストリームの中で文字列を検索する方法に違いはなく、ほとんどのキーワード オプションが想定どおりに機能します。次の表は、protected_content キーワード オプションと content キーワード オプションの間の例外的な相違点を要約しています。

表 36-7　protected_content オプションの例外

オプション	説明
ハッシュ タイプ（Hash Type）	protected_content ルール キーワードの新しいオプション。詳細については、ハッシュ タイプ（Hash Type）（36-21 ページ）を参照してください。
[大文字小文字の区別なし（Case Insensitive）]	未サポート
次の範囲内（Within）	未サポート
奥行き（Depth）	未サポート
長さ（Length）	protected_content ルール キーワードの新しいオプション。詳細については、長さ（Length）（36-24 ページ）を参照してください。
高速パターン マッチ機能を使用（Use Fast Pattern Matcher）	未サポート
高速パターン マッチ機能のみ（Fast Pattern Matcher Only）	未サポート
高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）	未サポート

シスコでは、protected_content キーワードを含むルールに 1 つ以上の content キーワードを含めることを推奨しています。こうすると、ルール エンジンが常に高速パターン マッチ機能を使用することで処理速度が上がり、パフォーマンスが向上します。ルール内の protected_content キーワードの前に content キーワードを配置します。ルールに 1 つ以上の content キーワードが含まれている場合は、content キーワードの Use Fast Pattern Matcher 引数が有効になっているかどうかに関係なく、ルール エンジンが高速パターン マッチ機能を使用することに注意してください。

コンテンツ マッチングの設定

ほとんどの場合、content または protected_content キーワードの後ろに修飾子を付けて、コンテンツを検索すべき場所、検索で大文字/小文字を区別するかどうか、およびその他のオプションを指定する必要があります。content および protected_content キーワードの修飾子の詳細については、コンテンツ一致の制約を参照してください。

ルールでイベントがトリガーとして使用されるためには、すべてのコンテンツ マッチングが真でなければならないことに注意してください。つまり、各コンテンツ マッチングは相互に AND 関係にあります。

また、インライン展開では、有害なコンテンツを照合した後でそれを同じ長さの独自のテキスト文字列に置き換えるルールをセットアップできることにも注意してください。詳細については、インライン展開でのコンテンツの置換（36-33 ページ）を参照してください。

照合するコンテンツを入力するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[コンテンツ（content）] フィールドに、検索する内容を入力します（たとえば `|90C8 C0FF FFFF|/bin/sh`）。

指定したコンテンツ**以外の**コンテンツを検索するには、[一致しない（Not）] チェック ボックスをオンにします。

注意　Not オプションが選択された 1 つの content キーワードだけを含むルールを作成した場合、侵入ポリシーの効果がなくなる可能性があります。詳細については、一致しない（Not）（36-22 ページ）を参照してください。

手順 2　オプションで、content キーワードを変更したり、キーワードの制約を追加したりするキーワードを追加します。他のキーワードの詳細については、ルールでのキーワードと引数について（36-11 ページ）を参照してください。

content キーワードの制約の詳細については、コンテンツ一致の制約（36-20 ページ）を参照してください。

手順 3　ルールの作成または編集を続けます。

詳細については、新しいルールの作成（36-116 ページ）または既存のルールの変更（36-118 ページ）を参照してください。

照合する保護されたコンテンツを入力するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　SHA512、SHA256、または MD5 ハッシュ ジェネレータを使用して、検索するコンテンツをエンコードします（たとえば、SHA512 ハッシュ ジェネレータを使って文字列 `Sample1` を実行します）。

ジェネレータが文字列のハッシュを出力します。

手順 2　protected_content フィールドに、ステップ 1 で生成したハッシュを入力します（たとえば `B20AABAF59605118593404BD42FE69BD8D6506EE7F1A71CE6BB470B1DF848C814BC5DBEC2081999F15691A71FAECA5FBA4A3F8B8AB56B7F04585DA6D73E5DD15`）。

指定したコンテンツ**以外の**コンテンツを検索するには、[一致しない（Not）] チェック ボックスをオンにします。

注意　**Not** オプションが選択された 1 つの `protected_content` キーワードだけを含むルールを作成した場合、侵入ポリシーの効果がなくなる可能性があります。詳細については、一致しない（Not）（36-22 ページ）を参照してください。

手順 3　[ハッシュ タイプ（Hash Type）] ドロップダウン リストから、ステップ 1 で使用したハッシュ関数（**SHA512** など）を選択します。なお、ステップ 2 で入力されたハッシュ内のビット数がハッシュ タイプと一致する**必要があります**。一致しない場合、システムはルールを保存しません。詳細については、ハッシュ タイプ（Hash Type）（36-21 ページ）を参照してください。

ヒント　シスコ設定の [デフォルト（Default）] を選択した場合、システムはハッシュ関数として SHA512 を想定します。

手順 4　必須の [長さ（Length）] フィールドに値を入力します。この値は、元の（ハッシュされていない）検索文字列の長さに対応する**必要があります**（たとえば、ステップ 2 の文字列 `Sample1` の長さは `7` です）。

詳細については、長さ（Length）（36-24 ページ）を参照してください。

手順 5　[オフセット（Offset）] フィールドまたは [距離（Distance）] フィールドに値を入力します。1 つのキーワード設定内で [オフセット（Offset）] オプションと [距離（Distance）] オプションは併用できません。

詳細については、protected_content キーワードでの検索位置オプションの使用（36-25 ページ）を参照してください。

手順 6　オプションで、`protected_content` キーワードを変更する制約オプションを追加します。

詳細については、コンテンツ一致の制約（36-20 ページ）を参照してください。

手順 7　オプションで、`protected_content` キーワードを変更する追加のキーワードを指定します。

詳細については、ルールでのキーワードと引数について（36-11 ページ）を参照してください。

手順 8　ルールの作成または編集を続けます。

詳細については、新しいルールの作成（36-116 ページ）または既存のルールの変更（36-118 ページ）を参照してください。

コンテンツ一致の制約

ライセンス：Protection

content または protected_content キーワードを変更するパラメータを使用すると、コンテンツ検索の位置や大文字/小文字の区別を制約できます。content または protected_content キーワードを変更するオプションを設定して、検索対象となるコンテンツを指定します。

詳細については、次の項を参照してください。

- 大文字小文字の区別なし（Case Insensitive）（36-20 ページ）
- ハッシュ タイプ（Hash Type）（36-21 ページ）
- raw データ（36-21 ページ）
- 一致しない（Not）（36-22 ページ）
- 検索位置オプション（Search Location Options）（36-23 ページ）
- HTTP コンテンツ オプション（36-26 ページ）
- 高速パターン マッチ機能を使用（Use Fast Pattern Matcher）（36-30 ページ）

大文字小文字の区別なし（Case Insensitive）

ライセンス：Protection

（注） このオプションは protected_content キーワードの設定では**サポートされません**。詳細については、protected_content キーワードの使用（36-17 ページ）を参照してください。

ASCII 文字列でコンテンツ一致を検索するときに大文字/小文字の区別を無視するようルール エンジンに指示できます。検索で大文字と小文字を区別しないようにするには、コンテンツ検索を指定するときに [大文字小文字の区別なし（Case Insensitive）] をオンにします。

コンテンツ検索時に [大文字小文字の区別なし（Case Insensitive）] を指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　追加する content キーワードに関して [大文字小文字の区別なし（Case Insensitive）] を選択します。

手順 2　ルールの作成または編集を続けます。

詳細については、コンテンツ一致の制約、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

ハッシュ タイプ（Hash Type）

ライセンス：Protection

（注）　このオプションは protected_content キーワードで**のみ**設定できます。詳細については、protected_content キーワードの使用（36-17 ページ）を参照してください。

[ハッシュ タイプ（Hash Type）] ドロップダウンを使用して、検索文字列のエンコードに使用されたハッシュ関数を特定します。システムは、protected_content 検索文字列のハッシュ方式として SHA512、SHA256、および MD5 をサポートしています。選択したハッシュ タイプとハッシュされたコンテンツの長さが一致しない場合、システムはルールを保存**しません**。

システムは自動的に、シスコ設定のデフォルト値を選択します。[デフォルト（Default）] を選択した場合、ルールには特定のハッシュ関数が含まれず、システムはハッシュ関数として SHA512 を想定します。

保護されたコンテンツ検索の実行時にハッシュ関数を指定するには、次の手順を実行します。

手順 1　[ハッシュタイプ（Hash Type）] ドロップダウン リストから、追加する protected_content キーワードのハッシュとして [デフォルト（Default）]、[SHA512]、[SHA256]、または [MD5] を選択します。

ヒント　シスコ設定の [デフォルト（Default）] を選択した場合、システムはハッシュ関数として SHA512 を想定します。詳細については、ハッシュ タイプ（Hash Type）（36-21 ページ）を参照してください。

手順 2　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または 既存のルールの変更（36-118 ページ）を参照してください。

raw データ

ライセンス：Protection

Raw Data オプションを使用すると、ルール エンジンは、正規化されたペイロード データ（ネットワーク分析ポリシーによってデコードされたデータ）を分析する前に、オリジナルのパケット ペイロードを分析します。引数値は使用されません。正規化の前に、ペイロード内の Telnet ネゴシエーション オプションを検査するために Telnet トラフィックを分析する場合に、このキーワードを使用できます。

同じ content または protected_content キーワードで、**Raw Data** オプションを HTTP コンテンツ オプションと一緒に使用することはできません。詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください。

ヒント　HTTP トラフィック内で raw データを検査するかどうか、および検査する raw データの量を決定するために、HTTP Inspect プリプロセッサの [クライアント フローの深さ（Client Flow Depth）] オプションと [サーバ フローの深さ（Server Flow Depth）] オプションを設定できます。詳細については、サーバレベル HTTP 正規化オプションの選択（27-36 ページ）を参照してください。

raw データを分析するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　追加する content または protected_content キーワードの [生データ（Raw Data）] チェック ボックスを選択します。

手順 2　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

一致しない（Not）

ライセンス：Protection

指定したコンテンツと一致しないコンテンツを検索するには、**Not** オプションを選択します。[一致しない（Not）] オプションが選択された content または protected_content キーワードを含むルールを作成する場合には、そのルール内に、[一致しない（Not）] オプションが選択されていない別の content または protected_content キーワードを 1 つ以上含める必要があります。

注意　content または protected_content キーワードに対して **Not** オプションを選択した場合は、そのキーワードだけを含むルールを作成しないでください。侵入ポリシーの効果がなくなる可能性があります。

たとえば、SMTP ルール 1:2541: 9 に 3 つの content キーワードが含まれており、そのうちの 1 つで [一致しない（Not）] オプションが選択されているとします。[一致しない（Not）] オプションが選択されているキーワード以外のすべての content キーワードを削除すると、このルールに基づくカスタム ルールが無効になります。このようなルールを侵入ポリシーに追加すると、そのポリシーの効果がなくなる可能性があります。

指定したコンテンツに一致しないコンテンツを検索するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　追加する content または protected_content キーワードの [一致しない（Not）] チェック ボックスを選択します。

ヒント　同じ content キーワードで、[一致しない（Not）] チェック ボックスと [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] チェック ボックスを同時に選択することはできません。

手順 2　[一致しない（Not）] オプションが選択されていない他の 1 つ以上の content または protected_content キーワードをルールに含めます。

手順 3　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

検索位置オプション（Search Location Options）

ライセンス：Protection

検索位置オプションを使用すると、指定したコンテンツの検索をどこから開始するか、どこまで検索するかを指定できます。各オプションの詳細については、以下を参照してください。

- 奥行き（Depth）（36-23 ページ）
- 距離（Distance）（36-23 ページ）
- 長さ（Length）（36-24 ページ）
- オフセット（Offset）（36-24 ページ）
- 次の範囲内（Within）（36-24 ページ）

content または protected_content キーワード内で検索位置オプションを使用する方法については、以下を参照してください。

- content キーワードでの検索位置オプションの使用（36-24 ページ）
- protected_content キーワードでの検索位置オプションの使用（36-25 ページ）

奥行き（Depth）

（注）　このオプションは、content キーワードを設定する場合に**のみ**サポートされます。詳細については、content キーワードの使用（36-17 ページ）を参照してください。

オフセット値の先頭からの（またはオフセットが設定されていない場合はパケット ペイロード先頭からの）コンテンツ検索の最大の深さをバイト単位で指定します。

たとえば、ルールのコンテンツ値が cgi-bin/phf、offset 値が 3、depth 値が 22 である場合、ルール ヘッダーで指定されたパラメータに合致するパケットでは、cgi-bin/phf 文字列に一致する文字列の検索がバイト位置 3 から 開始され、22 バイト処理した後（バイト位置 25 で）停止します。

指定したコンテンツの長さ以上の、最大 65535 バイトまでの値を指定する必要があります。値 0 は指定できません。

デフォルトの深さは、「パケットの末尾まで検索」です。

距離（Distance）

以前に見つかったコンテンツ一致から数えて、指定されたバイト数の後に出現する後続のコンテンツ一致を見つけるようルール エンジンに指示します。

Distance（距離）カウンタはバイト 0 から始まるため、最後に見つかったコンテンツ一致から順方向に移動すべきバイト数よりも 1 つ少ない数値を指定してください。たとえば 4 を指定した場合、5 番目のバイトから検索が始まります。

-65535 ～ 65535 バイトを値として指定できます。負の Distance 値を指定した場合は、検索を開始するバイト位置がパケットの先頭から外れる可能性があります。実際にはパケットの第 1 バイトから検索が開始されますが、計算ではパケットの外側のバイトも考慮されます。たとえば、パケット内の現在の位置が第 5 バイトで、次のコンテンツ ルール オプションで Distance 値 -10 および Within 値 20 が指定された場合、検索はペイロードの先頭から開始され、[Within] オプションが 15 に調整されます。

デフォルトの距離は 0 で、これは最後のコンテンツ一致の後のパケット内の現在位置という意味です。

長さ（Length）

（注）　このオプションは protected_content キーワードを設定する場合に**のみ**サポートされます。詳細については、protected_content キーワードの使用（36-17 ページ）を参照してください。

Length protected_content キーワード オプションは、ハッシュされていない検索文字列の長さをバイト単位で示します。

たとえば、コンテンツ Sample1 を使ってセキュア ハッシュを生成した場合には、**Length** 値として 7 を使用します。このフィールドに値を入力することは**必須**です。

オフセット（Offset）

パケット ペイロードの先頭を基準とする、コンテンツの検索を開始するパケット ペイロード内の位置をバイト単位で指定します。-65535 ～ 65535 バイトを値として指定できます。

オフセット カウンタはバイト 0 から始まるため、パケット ペイロードの先頭から順方向に移動すべきバイト数よりも 1 つ少ない数値を指定してください。たとえば 7 を指定した場合は、8 番目のバイトから検索が始まります。

デフォルトのオフセットは 0 で、これはパケットの先頭を意味します。

次の範囲内（Within）

（注）　このオプションは、content キーワードを設定する場合に**のみ**サポートされます。詳細については、content キーワードの使用（36-17 ページ）を参照してください。

Within オプションを使用すると、ルールをトリガーとして使用させるには、最後に見つかったコンテンツ一致の末尾以降、指定のバイト数以内に次のコンテンツ一致が発生する必要があることを指示できます。たとえば **Within** 値として 8 を指定した場合、次のコンテンツ一致がパケット ペイロードの次の 8 バイト以内に発生する必要があります。発生しない場合は、ルールをトリガーとして使用する基準が満たされません。

指定したコンテンツの長さ以上の、最大 65535 バイトまでの値を指定できます。

[Within] のデフォルトは「パケットの末尾まで検索」です。

content キーワードでの検索位置オプションの使用

次のように、2 つの content 位置ペアのいずれかを使用すると、指定したコンテンツの検索をどこから開始するか、どこまで検索するかを指定できます。

- パケット ペイロードの先頭を基準にして検索する場合は、[オフセット（Offset）] と [奥行き（Depth）] を一緒に使用します。
- 現在の検索位置を基準にして検索する場合は、[距離（Distance）] と [次の範囲内（Within）] を一緒に使用します。

ペアに含まれるオプションのどちらか 1 つだけを指定した場合は、そのペアのもう 1 つのオプションのデフォルトが想定されます。

Offset および **Depth** オプションと、**Distance** および **Within** オプションを混合することはできません。たとえば、**Offset** と **Within** をペアにすることはできません。1 つのルール内で任意の数の位置オプションを使用できます。

位置が指定されない場合は、[オフセット（Offset）] と [奥行き（Depth）] のデフォルトが想定されます。つまり、コンテンツ検索はパケット ペイロードの先頭から始まってパケットの末尾まで続きます。

また、既存の byte_extract 変数を使用して位置オプションの値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。

Web インターフェイスを使用して content キーワードで検索位置の値を指定する方法:

アクセス: Admin/Intrusion Admin

手順 1　追加する content キーワードのフィールドに値を入力します。次の選択肢があります。

- **Offset**
- **奥行（Depth）**
- **距離（Distance）**
- **次の範囲内（Within）**

1 つのルール内で任意の数の位置オプションを使用できます。

手順 2　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約（36-20 ページ）、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

protected_content キーワードでの検索位置オプションの使用

次のように、必須の [長さ（Length）] protected_content 位置オプションを [オフセット（Offset）] または [距離（Distance）] 位置オプションと組み合わせて使用すると、指定されたコンテンツの検索をどこから開始するか、どこまで検索するかを指定できます。

- パケット ペイロードの先頭を基準にして、保護された文字列を検索するには、[長さ（Length）] と [オフセット（Offset）] を一緒に使用します。
- 現在の検索位置を基準にして、保護された文字列を検索するには、[長さ（Length）] と [距離（Distance）] を一緒に使用します。

ヒント　1 つのキーワード設定内で [オフセット（Offset）] オプションと [距離（Distance）] オプションを併用することはできませんが、1 つのルール内では任意の数の位置オプションを使用できます。

位置が指定されない場合は、デフォルトが想定されます。つまり、コンテンツ検索はパケット ペイロードの先頭から始まってパケットの末尾まで続きます。

また、既存の byte_extract 変数を使用して位置オプションの値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。

Web インターフェイスを使用して protected_content キーワードで検索位置の値を指定する方法:

アクセス: Admin/Intrusion Admin

手順 1　追加する protected_content キーワードのフィールドに値を入力します。次の選択肢があります。

- **長さ（Length）**（必須）
- **Offset**
- **距離（Distance）**

1 つの protected_content キーワード内で [オフセット(Offset)] オプションと [距離(Distance)] オプションを混合することはできませんが、1 つのルール内では任意の数の位置オプションを使用できます。

手順 2　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約(36-20 ページ)、コンテンツ一致の検索(36-16 ページ)、新しいルールの作成(36-116 ページ)、または既存のルールの変更(36-118 ページ)を参照してください。

HTTP コンテンツ オプション

ライセンス: Protection

HTTP content または protected_content キーワード オプションを使用すると、HTTP Inspect プリプロセッサによってデコードされた HTTP メッセージ内でコンテンツ一致を検索する位置を指定できます。

次の 2 つのオプションは、HTTP 応答内のステータス フィールドを検索します。

- **HTTP ステータス コード(HTTP Status Code)**
- **HTTP ステータス メッセージ(HTTP Status Message)**

ルール エンジンは未加工の正規化されていないステータス フィールドを検索しますが、ここでは、他の Raw HTTP フィールドと正規化された HTTP フィールドを併用する際に考慮すべき制限についての説明を簡略化するために、これらのオプションが別個に列挙されていることに注意してください。

次の 5 つのオプションは、必要に応じて HTTP 要求、応答、またはその両方の中で正規化フィールドを検索します(詳細については、HTTP コンテンツ オプション(36-26 ページ)を参照してください)。

- **HTTP URI**
- **HTTP メソッド(HTTP Method)**
- **HTTP ヘッダー(HTTP Header)**
- **HTTP Cookie**
- **HTTP クライアント ボディ(HTTP Client Body)**

次の 3 つのオプションは、必要に応じて HTTP 要求、応答、またはその両方の中で未加工の(正規化されていない)非ステータス フィールドを検索します(詳細については、HTTP コンテンツ オプション(36-26 ページ)を参照してください)。

- **HTTP Raw URI**
- **HTTP Raw ヘッダー(HTTP Raw Header)**
- **HTTP Raw Cookie**

HTTP content オプションを選択する場合は、次のガイドラインに従ってください。

- HTTP content オプションは TCP トラフィックにのみ適用されます。
- パフォーマンスへの悪影響を避けるために、指定したコンテンツが出現する可能性のあるメッセージ部分だけを選択してください。

  たとえば、ショッピング カート メッセージの場合のように大きな cookie がトラフィックに含まれている可能性がある場合は、HTTP cookie ではなく HTTP ヘッダーの中で指定のコンテンツを検索することができます。

- HTTP Inspect プリプロセッサの正規化機能を活用し、パフォーマンスを向上させるには、作成するすべての HTTP 関連ルールの中に、**HTTP URI**、**HTTP Method**、**HTTP Header**、または **HTTP Client Body** オプションが選択された少なくとも 1 つの `content` または `protected_content` キーワードを含めてください。
- HTTP `content` または `protected_content` キーワード オプションと組み合わせて `replace` キーワードを使用することはできません。

単一の正規化された HTTP オプションまたはステータス フィールドを指定できます。または、複数の正規化 HTTP オプションとステータス フィールドを任意に組み合わせて、コンテンツ領域をマッチング対象にすることもできます。ただし、HTTP フィールド オプションを使用する場合には次の制限事項に注意してください。

- 同じ `content` または `protected_content` キーワードの中で、[生データ（Raw Data）] オプションを HTTP オプションと一緒に使用することはできません。
- Raw HTTP フィールド オプション（[HTTP　Raw URI]、[　HTTP Raw ヘッダー（HTTP Raw Header）]、または [HTTP Raw Cookie]）と、それぞれに対応する正規化されたオプション（[HTTP URI]、[HTTP ヘッダー（HTTP Header）]、または [HTTP Cookie]）を同じ `content` または `protected_content` キーワード内で一緒に使用することはできません。
- [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を、次の 1 つ以上の HTTP フィールド オプションと組み合わせて選択することはできません。

  [HTTP Raw URI]、[HTTP raw ヘッダー（HTTP Raw Header）]、[HTTP Raw Cookie]、[HTTP Cookie]、[HTTP メソッド（HTTP Method）]、[HTTP ステータス メッセージ（HTTP Status Message）]、[HTTP ステータス コード（HTTP Status Code）]

  ただし、次のいずれかの正規化フィールドを検索するために高速パターン マッチ機能を使用する `content` または `protected_content` キーワードでは、上記のオプションを含めることができます。

  [HTTP URI]、[HTTP ヘッダー（HTTP Header）]、[HTTP クライアント ボディ（HTTP Client Body）]

  たとえば、[HTTP Cookie]、[HTTP ヘッダー（HTTP Header）]、および [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を選択した場合、ルール エンジンは HTTP cookie と HTTP ヘッダーの両方でコンテンツを検索しますが、高速パターン マッチ機能は HTTP cookie ではなく、HTTP ヘッダーにのみ適用されます。
- 制限付きオプションと制限なしオプションを併用した場合、高速パターン マッチ機能は、指定された制限なしフィールドのみを検索することで、ルール エディタにルールを渡して（制限付きフィールドの評価を含む）完全な評価を行うべきかどうかを検査します。詳細については、高速パターン マッチ機能を使用（Use Fast Pattern Matcher）（36-30 ページ）を参照してください。

HTTP `content` および `protected_content` キーワード オプションに関する以下のリストでは、前述した制限事項が各オプションの説明に反映されています。

HTTP URI

正規化された要求 URI フィールド内でコンテンツ一致を検索するには、このオプションを選択します。

このオプションと `pcre` キーワードの HTTP URI（U）オプションを一緒に使用して、同じコンテンツを検索できないことに注意してください。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。

(注)　パイプライン処理された HTTP 要求パケットには複数の URI が含まれています。[HTTP URI] が選択されている場合、パイプライン処理された HTTP 要求パケットをルール エンジンが検出すると、そのパケット内のすべての URI でコンテンツ一致が検索されます。

HTTP Raw URI

正規化された要求 URI フィールド内でコンテンツ一致を検索するには、このオプションを選択します。

このオプションと pcre キーワードの HTTP URI(U)オプションを一緒に使用して、同じコンテンツを検索できないことに注意してください。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。

(注)　パイプライン処理された HTTP 要求パケットには複数の URI が含まれています。[HTTP URI] が選択されている場合、パイプライン処理された HTTP 要求パケットをルール エンジンが検出すると、そのパケット内のすべての URI でコンテンツ一致が検索されます。

HTTP メソッド(HTTP Method)

(URI で識別されるリソースに対して行う GET や POST などのアクションを特定する)要求メソッド フィールド内のコンテンツ一致を検索するには、このオプションを選択します。

HTTP ヘッダー(HTTP Header)

HTTP 要求内の(cookie を除く)正規化されたヘッダー フィールドでコンテンツ一致を検索するには、このオプションを選択します。また、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションが有効になっている場合は応答内でも検索されます。

このオプションと pcre キーワードの HTTP ヘッダー(H)オプションを一緒に使用して、同じコンテンツを検索できないことに注意してください。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。

HTTP Raw ヘッダー(HTTP Raw Header)

HTTP 要求内の(cookie を除く)raw ヘッダー フィールドでコンテンツ一致を検索するには、このオプションを選択します。また、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションが有効になっている場合は応答内でも検索されます。

このオプションと pcre キーワードの HTTP raw ヘッダー(D)オプションを一緒に使用して、同じコンテンツを検索できないことに注意してください。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。

HTTP Cookie

正規化された HTTP クライアント要求ヘッダー内で識別される cookie でコンテンツ一致を検索するには、このオプションを選択します。また、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションが有効になっている場合は応答 set-cookie データ内でも検索されます。システムは、メッセージ本文に含まれる cookie を本文の内容として扱うことに注意してください。

cookie 内だけで一致を検索するには、HTTP Inspect プリプロセッサの [HTTP Cookie の検査(Inspect HTTP Cookies)] オプションを有効にする必要があります。これを有効にしない場合、ルール エンジンは cookie を含むヘッダー全体を検索します。詳細については、サーバレベル HTTP 正規化オプションの選択(27-36 ページ)を参照してください。

次の点に注意してください。

- このオプションと pcre キーワードの HTTP cookie(C)オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。
- Cookie: ヘッダー名と Set-Cookie: ヘッダー名、ヘッダー行の先行スペース、およびヘッダー行の終わりを示す CRLF は cookie の一部としてではなく、ヘッダーの一部として検査されます。

HTTP Raw Cookie

未加工 HTTP クライアント要求ヘッダー内で識別される cookie でコンテンツ一致を検索するには、このオプションを選択します。また、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションが有効になっている場合は応答 set-cookie データ内でも検索されます。システムは、メッセージ本文に含まれる cookie を本文の内容として扱うことに注意してください。

cookie 内だけで一致を検索するには、HTTP Inspect プリプロセッサの [HTTP Cookie の検査(Inspect HTTP Cookies)] オプションを有効にする必要があります。これを有効にしない場合、ルール エンジンは cookie を含むヘッダー全体を検索します。詳細については、サーバレベル HTTP 正規化オプションの選択(27-36 ページ)を参照してください。

次の点に注意してください。

- このオプションと pcre キーワードの HTTP 未加工 cookie(K)オプションを一緒に使用して同じコンテンツを検索することはできません。詳細については、Snort 固有の正規表現後の修飾子の表を参照してください。
- Cookie: ヘッダー名と Set-Cookie: ヘッダー名、ヘッダー行の先行スペース、およびヘッダー行の終わりを示す CRLF は cookie の一部としてではなく、ヘッダーの一部として検査されます。

HTTP クライアント ボディ(HTTP Client Body)

HTTP クライアント要求内のメッセージ本文でコンテンツ一致を検索するには、このオプションを選択します。

このオプションが機能するためには、HTTP Inspect プリプロセッサの [HTTP クライアント ボディの抽出の深さ(HTTP Client Body Extraction Depth)] オプションで 0 ～ 65535 の値を指定する必要があることに注意してください。詳細については、サーバレベル HTTP 正規化オプションの選択(27-36 ページ)を参照してください。

HTTP ステータス コード(HTTP Status Code)

HTTP 応答内の 3 桁のステータス コードでコンテンツ一致を検索するには、このオプションを選択します。

このオプションで一致が返されるようにするには、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションを有効にする必要があります。詳細については、サーバレベル HTTP 正規化オプションの選択(27-36 ページ)を参照してください。

HTTP ステータス メッセージ(HTTP Status Message)

HTTP 応答のステータス コードに付加されるテキスト記述の中でコンテンツ一致を検索するには、このオプションを選択します。

このオプションで一致が返されるようにするには、HTTP Inspect プリプロセッサの [HTTP 応答の検査(Inspect HTTP Responses)] オプションを有効にする必要があります。詳細については、サーバレベル HTTP 正規化オプションの選択(27-36 ページ)を参照してください。

TCP トラフィックのコンテンツ検索を実行する場合に HTTP content オプションを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　オプションで、HTTP Inspect プリプロセッサの正規化を活用して、パフォーマンスを向上させるには、以下のように選択します。

- 追加する `content` または `protected_content` キーワードの [HTTP URI]、[HTTP Raw URI]、[HTTP メソッド（HTTP Method）]、[HTTP ヘッダー（HTTP Header）]、[HTTP Raw ヘッダー（HTTP Raw Header）]、または [HTTP クライアント ボディ（HTTP Client Body）] オプションから少なくとも 1 つ
- [HTTP Cookie] または [HTTP Raw Cookie] オプション

手順 2　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約（36-20 ページ）、コンテンツ一致の検索（36-16 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

高速パターン マッチ機能を使用（Use Fast Pattern Matcher）

ライセンス：Protection

（注）　これらのオプションは、`protected_content` キーワードの設定では**サポートされません**。詳細については、protected_content キーワードの使用（36-17 ページ）を参照してください。

高速パターン マッチ機能は、パケットをルール エンジンに渡す前に、評価するルールをすばやく決定します。この初期決定により、パケット評価で使用されるルール数が大幅に減るため、パフォーマンスが向上します。

デフォルトで、高速パターン マッチ機能は、ルールで指定された最長のコンテンツをパケットで検索します。これは、不必要なルール評価をできるだけ減らすためです。次の例のようなルール フラグメントがあるとします。

```
alert tcp any any -> any 80 (msg:"Exploit"; content:"GET";
http_method; nocase; content:"/exploit.cgi"; http_uri;
nocase;)
```

ほとんどすべての HTTP クライアント要求にはコンテンツ `GET` が含まれていますが、コンテンツ `/exploit.cgi` を含む要求は稀です。`GET` を高速パターン コンテンツとして使用した場合、ルール エンジンはほとんどのケースでこのルールを評価し、一致はほとんど検出されないでしょう。しかし、`/exploit.cgi` を使用するとほとんどのクライアントの `GET` 要求は評価されないため、パフォーマンスが向上します。

指定されたコンテンツが高速パターン マッチ機能で検出された場合にのみ、ルール エンジンはパケットをルールに照らして評価します。たとえば、ルール内の 1 つの `content` キーワードでコンテンツ `short` を指定し、別のキーワードで `longer`、さらに 3 番目のキーワードで `longest` を指定した場合、高速パターン マッチ機能はコンテンツ `longest` を使用し、ルール エンジンがペイロード内で `longest` を検出した場合にのみ、ルールが評価されます。

より短い検索パターンを高速パターン マッチ機能で使用するよう指定するには、**Use Fast Pattern Matcher** オプションを使用できます。理論的には、指定したパターンの方が最長パターンよりもパケット内で見つかる可能性が低いため、より的を絞って対象のエクスプロイトを識別できます。

[高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] と他のオプションを同じ content キーワード内で選択する場合は、次の制限事項に注意してください。

- ルールごとに 1 回だけ、[高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を指定できます。
- [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] と [一致しない（Not）] を組み合わせて選択した場合は、[距離（Distance）]、[次の範囲内（Within）]、[オフセット（Offset）]、または [奥行き（Depth）] を使用できません。
- [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を、次のいずれかの HTTP フィールド オプションと組み合わせて選択することはできません。

  HTTP Raw URI、**HTTP Raw Header**、**HTTP Raw Cookie**、**HTTP Cookie**、**HTTP Method**、**HTTP Status Message**、または **HTTP Status Code**

  ただし、次のいずれかの正規化フィールドを検索するために高速パターン マッチ機能を使用する content キーワードでは、上記のオプションを含めることができます。

  HTTP URI、**HTTP Header**、または **HTTP Client Body**

  たとえば、[HTTP Cookie]、[HTTP Header]、および [Use Fast Pattern Matcher] を選択した場合、ルール エンジンは HTTP cookie と HTTP ヘッダーの両方でコンテンツを検索しますが、高速パターン マッチ機能は HTTP cookie ではなく、HTTP ヘッダーにのみ適用されます。

  未加工 HTTP フィールド オプション（**HTTP Raw URI**、**HTTP Raw Header**、または **HTTP Raw Cookie**）と、それぞれに対応する正規化されたオプション（**HTTP URI**、**HTTP Header**、または **HTTP Cookie**）を同じ content キーワード内で一緒に使用できないことに注意してください。詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください。

  制限付きオプションと制限なしオプションを併用した場合、高速パターン マッチ機能は、指定された制限なしフィールドのみを検索することで、ルール エンジンにパケットを渡して（制限付きフィールドの評価を含む）完全な評価を行うべきかどうかを検査します。
- オプションで、[高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を選択した場合には [高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] または [高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）] を選択することもできますが、この両方は選択できません。
- Base64 データの検査時には高速パターン マッチ機能を使用できません（詳細については、Base64 データのデコードと検査（36-114 ページ）を参照してください）。

[高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] の使用

Fast Pattern Matcher Only オプションを使用すると、content キーワードをルール オプションとしてではなく、高速パターン マッチ機能オプションとしてのみ使用できます。指定したコンテンツをルール エンジンで評価する必要がない場合、このオプションを使ってリソースを節約できます。たとえば、ペイロード内のいずれかの場所にコンテンツ 12345 が存在することだけを必要とするルールがあるとします。高速パターン マッチ機能でパターンが検出された場合に、ルール内の追加のキーワードに照らしてパケットを評価できます。パターン 12345 が含まれているかどうかを判断するために、ルール エンジンがパケットを再評価する必要はありません。

指定されたコンテンツに関連する他の条件がルールに含まれている場合は、このオプションを使用しないでください。たとえば、別のルール条件で abcd が 1234 の前に出現するかどうかを判断する場合には、このオプションを使ってコンテンツ 1234 を検索しないでください。[高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] を指定すると、指定されたコンテンツがルール エンジンによって検索されないため、このケースではルール エンジンが相対的な位置を判断できません。

このオプションを使用するときには、次の条件に注意してください。

- 指定されたコンテンツは位置に依存しない、つまり、ペイロードのどこにでも出現する可能性があるため、位置オプション（[距離（Distance）]、[次の範囲内（Within）]、[オフセット（Offset）]、[奥行き（Depth）]、[高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）]）を使用することはできません。
- このオプションを [一致しない（Not）] と組み合わせて使用することはできません。
- このオプションを [高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）] と組み合わせて使用することはできません。
- すべてのパターンは、大文字と小文字を区別しない方法で、高速パターン マッチ機能に挿入されるため、指定したコンテンツは「大文字と小文字の区別なし」として扱われます。これは自動的に処理されるため、このオプションの選択時に [大文字小文字の区別なし（Case Insensitive）] を選択する必要はありません。
- [高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] オプションを使用する `content` キーワードの直後に、現在の検索位置を基準にして検索位置を設定する次のキーワードを続けないようにしてください。
- `isdataat`
- `pcre`
- `content`（[距離（Distance）] または [次の範囲内（Within）] が選択されている場合）
- `content`（[HTTP URI] が選択されている場合）
- `asn1`
- `byte_jump`
- `byte_test`
- `byte_extract`
- `base64_decode`

[高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）] の指定

Fast Pattern Matcher Offset and Length オプションを使用すると、検索するコンテンツの一部分を指定できます。これにより、パターンが非常に長く、ルールの一致の可能性を判断するのにパターンの一部分だけで十分な場合に、メモリ消費を抑えることができます。高速パターン マッチ機能によってルールが選択されたときに、パターン全体がルールに照らして評価されます。

次の構文に従い、検索を開始する位置（オフセット）およびコンテンツ内をどれほど検索するか（長さ）をバイト単位で指定することにより、高速パターン マッチ機能で使用する部分を決定します。

```
offset,length
```

たとえば、次のコンテンツに対して

```
1234567
```

次のようにオフセットと長さのバイト数を指定した場合、

```
1,5
```

高速パターン マッチ機能はコンテンツ `23456` のみを検索します。

このオプションを [高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] と一緒に使用できないことに注意してください。

高速パターン マッチ機能で検索されるコンテンツを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　追加する `content` キーワードに関して [高速パターン マッチ機能を使用（Use Fast Pattern Matcher）] を選択します。

手順 2　オプションで、指定したパターンがパケット内に存在するかどうかをルール エンジン評価なしで判断するには [高速パターン マッチ機能のみ（Fast Pattern Matcher Only）] を選択します。

指定されたコンテンツが高速パターン マッチ機能で検出された場合にのみ、評価が開始されます。

手順 3　オプションで、次の構文に従い、コンテンツの検索場所となるパターンの部分を [高速パターン マッチ機能オフセットおよび長さ（Fast Pattern Matcher Offset and Length）] で指定します。

```
offset,length
```

ここで、*offset* は検索の開始場所となるコンテンツ先頭からのバイト数を指定し、*length* は検索を続けるバイト数を指定します。

手順 4　ルールの作成または編集を続けます。詳細については、コンテンツ一致の制約（36-20 ページ）、PCRE を使用したコンテンツの検索（36-39 ページ）、新しいルールの作成（36-116 ページ）、または既存のルールの変更（36-118 ページ）を参照してください。

インライン展開でのコンテンツの置換

ライセンス：Protection

インライン展開で `replace` キーワードを使用すると、指定したコンテンツを置き換えることができます。

（注）　シスコ SSL アプライアンスによって検出された SSL トラフィック内のコンテンツを置き換えるために `replace` キーワードを使用することは**できません**。置換データではなく、元の暗号化データが送信されます。詳細については、『*シスコ SSL Appliance Administration and Deployment Guide*』を参照してください。

`replace` キーワードを使用するには、`content` キーワードを使って特定の文字列を検索するカスタム標準テキスト ルールを作成します。その後、`replace` キーワードを使用して、コンテンツを置き換える文字列を指定します。置換値とコンテンツ値は同じ長さである必要があります。

（注）　`protected_content` キーワード内でハッシュされたコンテンツを置き換えるために `replace` キーワードを使用することは**できません**。詳細については、protected_content キーワードの使用（36-17 ページ）を参照してください。

オプションで、以前の FireSIGHT システム ソフトウェア バージョンとの下位互換性を維持するために、置換文字列を引用符で囲むことができます。引用符を含めない場合は、それらが自動的にルールに追加されるため、構文的に正しいルールになります。置換テキストの一部として先行引用符または後続引用符を含めるには、次の例に示すように、バックスラッシュを使ってエスケープする必要があります。

```
"replacement text plus \"quotation\" marks""
```

1 つのルール内に複数の `replace` キーワードを含めることができますが、`content` キーワードごとに 1 つずつしか含めることができません。ルールによって検出されたコンテンツの最初のインスタンスだけが置き換えられます。

次に、replace キーワードの使用例を示します。

- エクスプロイトを含んでいる着信パケットをシステムが検出した場合、有害な文字列を無害な文字列に置き換えることができます。このテクニックは、有害なパケットを単に破棄するよりも効果的である場合があります。破棄されたパケットを攻撃者が単に再送信し続け、やがてネットワーク防御を通り抜けるか、ネットワークを氾濫させるという攻撃シナリオがあります。パケットを破棄する代わりに別の文字列に置換することで、脆弱ではないターゲットに対して攻撃が実行されたと攻撃者に思い込ませることができます。
- （たとえば Web サーバの）脆弱なバージョンが稼働しているかどうかを調べる偵察攻撃が懸念される場合は、発信パケットを検出して、バナーを独自のテキストに置換できます。

（注） 置換ルールを使用するインライン侵入ポリシー内でルール状態が [イベントを生成する（Generate Events）] に設定されていることを確認してください。ルールを [ドロップしてイベントを生成する（Drop and Generate Events）] に設定した場合はパケットが破棄され、コンテンツが置き換えられません。

文字列置換プロセスでは、宛先ホストがエラーなしでパケットを受信できるように、パケットチェックサムがシステムによって自動的に更新されます。

replace キーワードを HTTP 要求メッセージ content キーワード オプションと組み合わせて使用できないことに注意してください。詳細については、「コンテンツ一致の検索（36-16 ページ）」と「HTTP コンテンツ オプション（36-26 ページ）」を参照してください。

インライン展開でコンテンツを置換するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [コンテンツ（content）] を選択して、[オプションの追加（Add option）] をクリックします。

content キーワードが表示されます。

手順 2　[コンテンツ（content）] フィールドで、検出するコンテンツを指定します。オプションで、該当する引数を選択します。HTTP 要求メッセージ content キーワード オプションを replace キーワードと一緒に使用できないことに注意してください。

手順 3　ドロップダウン リストから [置き換え（replace）] を選択して、[オプションの追加（Add Option）] をクリックします。

replace キーワードが content キーワードの下に表示されます。

手順 4　[replace:] フィールドで、指定したコンテンツに対する置換文字列を指定します。

Byte_Jump と Byte_Test の使用

ライセンス：Protection

byte_jump と byte_test を使用すると、パケット内のどの位置でルール エンジンがデータ マッチング検査を開始すべきか、どのバイトを評価すべきかを計算できます。

また、byte_jump および byte_test **DCE/RPC** 引数を使用すると、DCE/RPC プリプロセッサで処理されるトラフィック用にいずれかのキーワードを調整できます。**DCE/RPC** 引数を使用するときには、他の特定の DCE/RPC キーワードと組み合わせて byte_jump と byte_test を使用することもできます。詳細については、DCE/RPC トラフィックのデコード（27-2 ページ）と DCE/RPC キーワード（36-67 ページ）を参照してください。

詳細については、次の各項を参照してください。

- byte_jump（36-35 ページ）
- byte_test（36-37 ページ）

byte_jump

ライセンス：Protection

byte_jump キーワードは、指定されたバイト セグメントで定義されるバイト数を計算し、指定したオプションに応じて、指定されたバイト セグメントの末尾から順方向に、またはパケット ペイロードの先頭から、パケット内でそのバイト数だけスキップします。パケットの特定のバイト セグメントが、パケット内の可変データに含まれるバイト数を示す場合には、これが役立ちます。

次の表では、byte_jump キーワードで必要な引数を説明します。

表 36-8　byte_jump の必須の引数

引数	説明
Bytes	パケットから計算するバイト数。
Offset	ペイロード内で処理を開始するバイト数。offset カウンタはバイト 0 から始まるため、パケット ペイロードの先頭、または最後に見つかったコンテンツ一致から順方向にジャンプさせるバイト数から 1 を差し引いて offset 値を計算してください。 また、既存の byte_extract 変数を使用してこの引数の値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。

次の表で説明するオプションを使用すると、必須の引数に指定された値をシステムがどのように解釈するかを定義できます。

表 36-9　byte_jump の追加のオプション引数

引数	説明
Relative	最後に見つかったコンテンツ一致で検出された最後のパターンを基準にしてオフセットを計算します。
Align	変換されたバイト数を次の 32 ビット境界に切り上げます。
Multiplier	ルール エンジンで最終的な byte_jump 値を算出するために、パケットから得られた byte_jump 値に掛ける値を示します。 つまり、ルール エンジンは、指定されたバイト セグメントで定義されるバイト数だけスキップする代わりに、Multiplier 引数で指定される整数を乗算したバイト数だけスキップします。

表 36-9　byte_jump の追加のオプション引数（続き）

引数	説明
Post Jump Offset	他の byte_jump 引数を適用した後に、順方向または逆方向にスキップするバイト数（-63535 ～ 63535）。正の値は順方向にスキップし、負の値は逆方向にスキップします。無効にするには、フィールドを空白のままにするか、0 を入力します。 **DCE/RPC** 引数を選択したときに適用されない byte_jump 引数については、エンディアンネス引数の表の **DCE/RPC** 引数を参照してください。
From Beginning	スキップするバイト数を示すバイト セグメントの末尾からではなく、パケット ペイロードの先頭から数えて、指定されたバイト数だけペイロード内をスキップするようルール エンジンに指示します。

DCE/RPC、**Endian**、または **Number Type** のうち 1 つだけを指定できます。

byte_jump キーワードでどのようにバイト数を計算するかを定義するには、次の表に示す引数から選択できます（どの引数も指定されない場合は、ネットワーク バイト順が使用されます）。

表 36-10　エンディアンネス引数

引数	説明
Big Endian	デフォルトのネットワーク バイト順であるビッグ エンディアン バイト順でデータを処理します。
Little Endian	リトル エンディアン バイト順でデータを処理します。
DCE/RPC	DCE/RPC プリプロセッサで処理されるトラフィック用に byte_jump キーワードを指定します。詳細については、DCE/RPC トラフィックのデコード（27-2 ページ）を参照してください。 DCE/RPC プリプロセッサがビッグ エンディアンまたはリトル エンディアン バイト順を決定します。**Number Type**、**Endian**、および **From Beginning** 引数は適用されません。 この引数を有効にした場合は、他の特定の DCE/RPC キーワードと組み合わせて byte_jump を使用することもできます。詳細については、DCE/RPC キーワード（36-67 ページ）を参照してください。

次の表に示すいずれか 1 つの引数を使用して、パケット内のストリング データをシステムがどのように表示するかを定義します。

表 36-11　Number Type 引数

引数	説明
Hexadecimal String	変換後のストリング データを 16 進形式で表現します。
Decimal String	変換後のストリング データを 10 進形式で表現します。
Octal String	変換後のストリング データを 8 進形式で表現します。

たとえば、次のような値を byte_jump に設定した場合、

- Bytes = 4
- Offset = 12
- Relative enabled
- Align enabled

ルール エンジンは、最後に見つかったコンテンツ一致から 13 バイト後に出現する 4 つのバイトで記述される数値を計算して、そのバイト数だけパケット内を順方向にスキップします。たとえば、ある特定のパケット内で計算される 4 つのバイトが 00 00 00 1F である場合、ルール エンジンはこれを 31 に変換します。(次の 32 ビット境界まで移動するようエンジンに指示する)align が指定されているため、ルール エンジンはパケット内を 32 バイト先までスキップします。

あるいは、次のような値を byte_jump に設定した場合、

- Bytes = 4
- Offset = 12
- From Beginning enabled
- Multiplier = 2

ルール エンジンは、パケットの先頭から 13 バイト後に出現する 4 つのバイトで記述される数値を計算します。その後、その数値に 2 を掛けてスキップする総バイト数を計算します。たとえば、ある特定のパケット内で計算される 4 つのバイトが 00 00 00 1F である場合、ルール エンジンはこれを 31 に変換し、それに 2 を掛けて 62 にします。[From Beginning] が有効になっているため、ルール エンジンはパケット内の最初の 63 バイトをスキップします。

byte_jump を使用するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　ドロップダウン リストから [byte_jump] を選択して、[オプションの追加(Add Option)] をクリックします。

[byte_jump] セクションが、選択された最後のキーワードの下に表示されます。

byte_test

ライセンス: Protection

byte_test キーワードは、指定されたバイト セグメント内のバイト数を計算し、指定した演算子と値に基づいてそれらを比較します。

次の表に、byte_test キーワードで必要な引数を説明します。

表 36-12　byte_test の必須の引数

引数	説明
Bytes	パケットから計算するバイト数。1 ～ 10 バイトを指定できます。
Operator and Value	指定された値を <、>、=、!、&、^、!>、!<、!=、!&、または !^ で比較します。 たとえば `!1024` と指定した場合、`byte_test` は指定された数値を変換し、それが 1024 と等しくなければイベントが生成されます（他のすべてのキーワード パラメータが一致する場合）。 「!」と「!=」は等価であることに注意してください。 また、既存の `byte_extract` 変数を使用してこの引数の値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。
Offset	ペイロード内で処理を開始するバイト数。`offset` カウンタはバイト 0 から始まるため、パケット ペイロードの先頭、または最後に見つかったコンテンツ一致から順方向にカウントするバイト数から 1 を差し引いて `offset` 値を計算してください。 また、既存の `byte_extract` 変数を使用してこの引数の値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。

次の表に示す引数を使用すると、システムで `byte_test` 引数がどのように使用されるかをさらに定義できます。

表 36-13　byte_test の追加のオプション引数

引数	説明
Relative	最後に見つかったパターン一致を基準にしてオフセットを計算します。
Align	変換されたバイト数を次の 32 ビット境界に切り上げます。

DCE/RPC、**Endian**、または **Number Type** のうち 1 つだけを指定できます。

検査対象となるバイトを `byte_test` キーワードでどのように計算するか定義するには、次の表の中から引数を選択します。どの引数も指定しない場合は、ネットワーク バイト順が使用されます。

表 36-14　byte_test のエンディアンネス引数

引数	説明
Big Endian	デフォルトのネットワーク バイト順であるビッグ エンディアン バイト順でデータを処理します。

表 36-14　byte_test のエンディアンネス引数（続き）

引数	説明
Little Endian	リトル エンディアン バイト順でデータを処理します。
DCE/RPC	DCE/RPC プリプロセッサで処理されるトラフィック用に `byte_test` キーワードを指定します。詳細については、DCE/RPC トラフィックのデコード（27-2 ページ）を参照してください。 DCE/RPC プリプロセッサがビッグ エンディアンまたはリトル エンディアン バイト順を決定します。**Number Type** 引数と **Endian** 引数は適用されません。 この引数を有効にした場合は、他の特定の DCE/RPC キーワードと組み合わせて `byte_test` を使用することもできます。詳細については、DCE/RPC キーワード（36-67 ページ）を参照してください。

次の表に示すいずれか 1 つの引数を使用して、パケット内のストリング データをシステムがどのように表示するかを定義できます。

表 36-15　Number Type byte-test 引数

引数	説明
Hexadecimal String	変換後のストリング データを 16 進形式で表現します。
Decimal String	変換後のストリング データを 10 進形式で表現します。
Octal String	変換後のストリング データを 8 進形式で表現します。

たとえば、次のような値を `byte_test` に指定した場合、

- Bytes = 4
- Operator and Value > 128
- Offset = 8
- Relative enabled

ルール エンジンは、最後に見つかったコンテンツ一致から（それを基準にして）9 バイト後に出現する 4 つのバイトで記述される数値を計算し、その計算値が 128 バイトを超えた場合に、ルールがトリガーとして使用されます。

byte_test を使用するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [byte_test] を選択して、[オプションの追加（Add option）] をクリックします。

[byte_test] セクションが、選択された最後のキーワードの下に表示されます。

PCRE を使用したコンテンツの検索

ライセンス：Protection

pcre キーワードを使用すると、指定されたコンテンツをパケット ペイロード内で検査するために Perl 互換正規表現（PCRE）を使用できます。PCRE を使用すると、同じ内容のわずかなバリエーションにそれぞれ一致する複数のルールを作成する手間が省けます。

正規表現は、さまざまな方法で表現されることのあるコンテンツを検索する場合に役立ちます。パケットのペイロード内でコンテンツを検索するときには、コンテンツがさまざまな属性を持つ可能性があることを考慮すべき場合があります。

侵入ルールで使われる正規表現構文は完全な正規表現ライブラリのサブセットであり、完全なライブラリ内のコマンドで使用される構文とはいくつかの点で異なることに注意してください。ルール エディタを使用して pcre キーワードを追加するときには、次の形式で完全な値を入力します。

```
!/pcre/ ismxAEGRBUIPHDMCKSY
```

引数の説明

- 「!」は否定オプションです（正規表現に**一致しない**パターンを照合する場合に使用します）。
- */pcre/* は Perl 互換正規表現です。
- `ismxAEGRBUIPHDMCKSY` は修飾子オプションの任意の組み合わせです。

また、次の表に示す文字をエスケープする必要があることに注意してください。これにより、パケット ペイロード内で特定のコンテンツを検索するために PCRE でこれらの文字を使用した場合、ルール エンジンがそれを正しく解釈するようになります。

表 36-16　エスケープする PCRE 文字

エスケープする必要のある文字	バックスラッシュを使用した場合	16 進コードを使用した場合
#（ナンバー記号）	\#	\x23
;（セミコロン）	\;	\x3B
|（縦棒）	\\|	\x7C
:（コロン）	\:	\x3A

ヒント　必要に応じて、Perl 互換正規表現を引用符で囲むこともできます（例：*pcre_expression* または *"pcre_expression"*）。引用符が任意ではなく必須であった旧バージョンに慣れている経験豊富なユーザのために、引用符を使用するオプションが提供されています。保存後のルールをルール エディタで表示すると、引用符が表示されません。

`m?regex?` を使用することもできます。ここで、`?` は「/」以外のデリミタです。正規表現内でスラッシュと一致させる必要があり、バックスラッシュを使ってそれをエスケープしたくない場合には、これを使用できます。たとえば、「`m?`*regex*`? ismxAEGRBUIPHDMCKSY`」のように使用できます。*regex* は Perl 互換正規表現、`ismxAEGRBUIPHDMCKSY` は修飾子オプションの任意の組み合わせです。正規表現の構文の詳細については、Perl 互換正規表現の基本（36-41 ページ）を参照してください。

以下の項では、有効な `pcre` キーワードの値を作成する方法について詳しく説明します。

- Perl 互換正規表現の基本（36-41 ページ）では、Perl 互換正規表現で使われる一般的な構文について説明します。
- PCRE 修飾子のオプション（36-43 ページ）では、正規表現を変更するために使用できるオプションについて説明します。
- PCRE キーワード値の例（36-46 ページ）では、ルールにおける `pcre` キーワードの使用例を示します。

Perl 互換正規表現の基本

ライセンス：Protection

`pcre` キーワードでは、標準の Perl 互換正規表現（PCRE）構文を使用できます。以下の項では、この構文について説明します。

ヒント　ここでは PCRE で使用可能な基本的な構文について説明しますが、Perl および PCRE 専用のオンライン リファレンスやブックで、さらに詳しい情報を参照することもできます。

メタ文字

ライセンス：Protection

メタ文字は正規表現内で特別な意味を持つリテラル文字です。メタ文字を正規表現内で使用するときには、その前にバックスラッシュを付けて「エスケープする」必要があります。

次の表に、PCRE で使用可能なメタ文字について説明し、それぞれの例を示します。

表 36-17　PCRE メタ文字（PCRE Metacharacters）

メタ文字	説明	例
.	改行以外の任意の文字と一致します。修飾オプションとして s が使用されている場合は、改行文字も含まれます。	abc. は、abcd、abc1、abc# などと一致します。
*	ある文字または式の 0 回以上の出現と一致します。	abc* は、abc、abcc、abccc、abccccc などと一致します。
?	ある文字または式の 0 回または 1 回の出現と一致します。	abc? は abc に一致します。
+	ある文字または式の 1 回以上の出現と一致します。	abc+ は、abc、abcc、abccc、abccccc などと一致します。
()	式をグループ化します。	(abc)+ は、abc、abcabc、abcabcabc などと一致します。
{}	ある文字または式の一致回数の限度を指定します。下限と上限を設定する場合には、下限と上限をカンマで区切ります。	a{4,6} は、aaaa、aaaaa、または aaaaaa と一致します。 (ab){2} は abab と一致します。
[]	文字クラスを定義できます。セットの中で記述される任意の文字または文字の組み合わせに一致します。	[abc123] は、a または b または c などと一致します。
^	文字列の先頭でコンテンツを照合します。また、文字クラスの中で否定としても使用されます。	^in は、info 内の “in” と一致しますが、bin では一致しません。[^a] は、a を含まない任意の文字列と一致します。
$	文字列の末尾でコンテンツを照合します。	ce$ は、announce 内の “ce” と一致しますが、cent では一致しません。
|	OR 式を示します。	(MAILTO|HELP) は、MAILTO または HELP と一致します。
\	メタ文字を実際の文字として使用できます。また、事前定義された文字クラスを指定するためにも使われます。	\. はピリオドと一致し、* はアスタリスクと一致し、\\ はバックスラッシュと一致します。\d は数字と一致し、\w は英数字と一致します。PCRE での文字クラスの使用方法については、文字クラス（36-42 ページ）を参照してください。

文字クラス

ライセンス：Protection

文字クラスには、英字、数字、英数字、および空白文字があります。大カッコで囲んで独自の文字クラスを作成できます（メタ文字（36-42 ページ）を参照）。また、事前定義のクラスをさまざまな文字タイプのショートカットとして使用することもできます。追加の修飾子なしで文字クラスを使用すると、1 つの文字クラスは 1 桁または 1 文字に一致します。

次の表に、PCRE で使用できる事前定義の文字クラスの説明と例を示します。

表 36-18　PCRE 文字クラス

文字クラス	説明	文字クラスの定義
\d	数字(桁)と一致します。	[0-9]
\D	数字以外の任意の文字と一致します。	[^0-9]
\w	英数字(語)と一致します。	[a-zA-Z0-9_]
\W	英数字以外の任意の文字と一致します。	[^a-zA-Z0-9_]
\s	スペース、復帰、タブ、改行、および改ページを含む空白文字と一致します。	[\r\t\n\f]
\S	空白文字以外の任意の文字と一致します。	[^ \r\t\n\f]

PCRE 修飾子のオプション

ライセンス：Protection

pcre キーワードの値の中で正規表現構文を指定した後、修飾オプションを使用できます。これらの修飾子は、Perl、PCRE、および Snort 固有の処理機能を実行します。修飾子は、常に PCRE 値の末尾に、次の形式で出現します。

/*pcre*/ismxAEGRBUIPHDMCKSY

ここで、ismxAEGRBUPHMC には、次の表に示す任意の修飾オプションを含めることができます。

ヒント　オプションで、正規表現と修飾オプションを引用符で囲むことができます(たとえば "/pcre/ismxAEGRBUIPHDMCKSY")。引用符が任意ではなく必須であった旧バージョンに慣れている経験豊富なユーザのために、引用符を使用するオプションが提供されています。保存後のルールをルール エディタで表示すると、引用符が表示されません。

次の表に、Perl 処理機能を実行するために使用できるオプションを説明します。

表 36-19　Perl 関連の正規表現後オプション

オプション	説明
i	正規表現で大文字と小文字を区別しないようにします。
s	ドット文字(.)は、改行または \n 文字を除くすべての文字を表します。オプションとして "s" を使用すると、これをオーバーライドして、改行文字を含むすべての文字をドット文字に一致させることができます。
m	デフォルトで、1 つの文字列は複数文字からなる単一行として扱われ、^ と $ は特定の文字列の先頭および末尾に一致します。オプションとして "m" を使用すると、^ および $ はバッファの先頭または末尾だけでなく、バッファ内の改行文字の直前または直後のコンテンツとも一致します。
x	エスケープされた(バックスラッシュが先行する)場合、および文字クラスに含まれる場合を除き、空白データ文字がパターン内に出現してもそれを無視します。

次の表に、正規表現の後ろに使用できる PCRE 修飾子の説明を示します。

表 36-20　PCRE 関連の正規表現後オプション

オプション	説明
A	文字列の先頭でパターンが一致する必要があります(正規表現で ^ を使用した場合と同じ)。
E	対象の文字列の末尾でのみ一致するように $ を設定します(E を伴わない $ は、それが改行である場合には最後の文字の直前とも一致しますが、他の改行文字の直前とは一致しません)。
G	デフォルトでは、* + と ? は「最長マッチ」を実行します。つまり、複数の一致が見つかった場合は最も長い一致が選択されます。G 文字を使用するとこの動作が変更され、常に最初の一致がこれらの文字で選択されます。ただし後ろに疑問符(?)が続く場合を除きます。たとえば、*? +? と ?? は G 修飾子を使った構造内で最長マッチを実行し、疑問符が付いていない *、+、または ? は最長マッチではありません。

次の表は、正規表現の後ろに付加できる Snort 固有の修飾子を示しています。

表 36-21　Snort 固有の正規表現後の修飾子

オプション	説明
R	ルール エンジンで見つかった最後の一致の末尾を基準にして、一致するコンテンツを検索します。
B	プリプロセッサによってデコードされる前のデータ内のコンテンツを検索します(このオプションは、content または protected_content キーワードとともに 生データ(Raw Data) 引数を使用する場合に似ています)。
U	HTTP Inspect プリプロセッサによってデコードされた正規化済み HTTP 要求メッセージの URI 内のコンテンツを検索します。このオプションと content または protected_content キーワードの **HTTP URI** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション(36-26 ページ)を参照してください。 (注)　パイプライン処理された HTTP 要求パケットには複数の URI が含まれています。U オプションを含む PCRE 式を使用すると、ルール エンジンは、パイプライン処理された HTTP 要求パケット内の最初の URI でのみコンテンツ一致を検索します。パケット内のすべての URI を検索するには、U オプションを使った PCRE 式を一緒に使用するかどうかに関係なく、[HTTP URI] を選択した content または protected_content キーワードを使用してください。
I	HTTP Inspect プリプロセッサによってデコードされた raw HTTP 要求メッセージの URI 内のコンテンツを検索します。このオプションと content または protected_content キーワードの **HTTP Raw URI** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション(36-26 ページ)を参照してください。
P	HTTP Inspect プリプロセッサによってデコードされた正規化済み HTTP 要求メッセージ本文の中でコンテンツを検索します。詳細については、HTTP コンテンツ オプション(36-26 ページ)で、content および protected_content キーワードの [HTTP クライアント ボディ(HTTP Client Body)] オプションを参照してください。
H	HTTP Inspect プリプロセッサによってデコードされた HTTP 要求または応答メッセージの(cookie を除く)ヘッダー内のコンテンツを検索します。このオプションと content または protected_content キーワードの **HTTP Header** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション(36-26 ページ)を参照してください。

表 36-21　Snort 固有の正規表現後の修飾子（続き）

オプション	説明
D	HTTP Inspect プリプロセッサによってデコードされた未加工の HTTP 要求または応答メッセージの（cookie を除く）ヘッダー内のコンテンツを検索します。このオプションと content または protected_content キーワードの **HTTP Raw Header** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください。
M	HTTP Inspect プリプロセッサによってデコードされた正規化済み HTTP 要求メッセージのメソッド フィールド内のコンテンツを検索します。メソッド フィールドは、URI で識別されるリソースに対して実行すべきアクション（GET、PUT、CONNECT など）を特定します。詳細については、HTTP コンテンツ オプション（36-26 ページ）で、content および protected_content キーワードの [HTTP メソッド（HTTP Method）] オプションを参照してください。
C	HTTP Inspect プリプロセッサの [HTTP Cookie を検査（Inspect HTTP Cookies）] オプションが有効になっている場合は、HTTP 要求ヘッダーの cookie 内の正規化済みコンテンツを検索します。さらに、プリプロセッサの [HTTP 応答を検査（Inspect HTTP Responses）] オプションが有効になっている場合は、HTTP 応答ヘッダーの set-cookie 内も検索します。[HTTP 応答を検査（Inspect HTTP Responses）] が有効になっていない場合は、cookie または set-cookie データを含めて、ヘッダー全体を検索します。 次の点に注意してください。 • メッセージ本文に含まれる cookie は、本文のコンテンツとして扱われます。 • このオプションと content または protected_content キーワードの **HTTP Cookie** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください。 • Cookie: ヘッダー名と Set-Cookie: ヘッダー名、ヘッダー行の先行スペース、およびヘッダー行の終わりを示す CRLF は cookie の一部としてではなく、ヘッダーの一部として検査されます。
K	HTTP Inspect プリプロセッサの [HTTP Cookie を検査（Inspect HTTP Cookies）] オプションが有効になっている場合は、HTTP 要求ヘッダーの cookie 内の未加工コンテンツを検索します。さらに、プリプロセッサの [HTTP 応答を検査（Inspect HTTP Responses）] オプションが有効になっている場合は、HTTP 応答ヘッダーの set-cookie 内も検索します。[HTTP 応答を検査（Inspect HTTP Responses）] が有効になっていない場合は、cookie または set-cookie データを含めて、ヘッダー全体を検索します。 次の点に注意してください。 • メッセージ本文に含まれる cookie は、本文のコンテンツとして扱われます。 • このオプションと content または protected_content キーワードの **HTTP Raw Cookie** オプションを一緒に使用して、同じコンテンツを検索することはできません。詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください。 • Cookie: ヘッダー名と Set-Cookie: ヘッダー名、ヘッダー行の先行スペース、およびヘッダー行の終わりを示す CRLF は cookie の一部としてではなく、ヘッダーの一部として検査されます。
S	HTTP 応答内の 3 桁のステータス コードを検索します。詳細については、HTTP コンテンツ オプション（36-26 ページ）で、content および protected_content キーワードの [HTTP ステータス コード（HTTP Status Code）] オプションを参照してください。
Y	HTTP 応答内のステータス コードに付加されるテキスト記述を検索します。詳細については、HTTP コンテンツ オプション（36-26 ページ）で、content および protected_content キーワードの [HTTP ステータス メッセージ（HTTP Status Message）] オプションを参照してください。

(注)　U オプションと R オプションを組み合わせて使用しないでください。パフォーマンスの問題が発生する可能性があります。また、他の HTTP コンテンツ オプション（I、P、H、D、M、C、K、S または Y）と組み合わせて U オプションを使用しないでください。

PCRE キーワード値の例

ライセンス：Protection

次に、pcre で入力できる値の例を示し、それぞれの例で何が一致するかを説明します。

- **`/feedback[(\d{0,1})]?\.cgi/U`**

  この例では、URI データにのみ配置された、feedback の後に 0 個または 1 個の数字、さらに .cgi が続くインスタンスをパケット ペイロード内で検索します。

  この例は以下のものと一致します。

- `feedback.cgi`
- `feedback1.cgi`
- `feedback2.cgi`
- `feedback3.cgi`

  この例は、以下のものとは一致**しません**。

- `feedbacka.cgi`
- `feedback11.cgi`
- `feedback21.cgi`
- `feedbackzb.cgi`
- **`/^ez(\w{3,5})\.cgi/iU`**

  この例では、先頭の ez の後に 3 ～ 5 文字の単語、さらに .cgi が続く文字列をパケット ペイロード内で検索します。この検索では大文字と小文字は区別されず、URI データだけが検索されます。

  この例は以下のものと一致します。

- `EZBoard.cgi`
- `ezman.cgi`
- `ezadmin.cgi`
- `EZAdmin.cgi`

  この例は、以下のものとは一致**しません**。

- `ezez.cgi`
- `fez.cgi`
- `abcezboard.cgi`
- `ezboardman.cgi`
- **`/mail(file|seek)\.cgi/U`**

  この例では、URI データ内の mail の後に file と seek のどちらかが続く文字列をパケット ペイロードで検索します。

  この例は以下のものと一致します。

- `mailfile.cgi`
- `mailseek.cgi`

  この例は、以下のものとは一致**しません**。

- `MailFile.cgi`

- mailfilefile.cgi

- **m?http\\x3a\x2f\x2f.*(\n|\t)+?U**

  この例では、任意の数の文字の後ろにある、HTTP 要求内のタブまたは改行文字を示す URI コンテンツをパケット ペイロード内で検索します。この例では、式で m?*regex*? を使用して、http\:\/\/ を使用しないようにしています。コロンの前にバックスラッシュがあることに注意してください。

  この例は以下のものと一致します。

- http://www.example.com?scriptvar=x&othervar=\n\..\..
- http://www.example.com?scriptvar=\t

  この例は、以下のものとは一致**しません**。

- ftp://ftp.example.com?scriptvar=&othervar=\n\..\..
- http://www.example.com?scriptvar=|/bin/sh -i|
- **m?http\\x3a\x2f\x2f.*=\|.*\|+?sU**

  この例では、(改行を含む)任意の数の文字の後に 1 つの等号、さらに任意の数の文字または空白を含むパイプ文字が続くという構成の URL をパケット ペイロード内で検索します。この例では、式で m?*regex*? を使用して、http\:\/\/ を使用しないようにしています。

  この例は以下のものと一致します。

- http://www.example.com?value=|/bin/sh/ -i|
- http://www.example.com?input=|cat /etc/passwd|

  この例は、以下のものとは一致**しません**。

- ftp://ftp.example.com?value=|/bin/sh/ -i|
- http://www.example.com?value=x&input?|cat /etc/passwd|
- **/[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}\:[0-9a-f]{2}/i**

  この例では、MAC アドレスをパケット ペイロード内で検索します。コロン文字がバックスラッシュでエスケープされていることに注意してください。

ルールへのメタデータの追加

ライセンス: Protection

metadata キーワードを使用すると、記述情報をルールに追加できます。追加した情報を使用して、ニーズに合う方法でルールを整理/識別したり、ルールを検索したりできます。

システムは次の形式に基づいてメタデータを検証します。

key value

ここで、*key* と *value* は、スペースで区切られた記述の組み合わせです。これは、シスコ 提供のルールにメタデータを追加するために シスコ VRT で使用されている形式です。

または、次の形式を使用することもできます。

key=value

たとえば、*key value* 形式で次のようにカテゴリとサブカテゴリを使用して、作成者と日付によってルールを識別できます。

```
author SnortGuru_20050406
```

1 つのルール内で複数の metadata キーワードを使用できます。また、以下の例に示すように、単一の metadata キーワード内で複数の *key value* ステートメントをカンマで区切ることもできます。

```
author SnortGuru_20050406, revised_by SnortUser1_20050707,
revised_by SnortUser2_20061003, revised_by
SnortUser1_20070123
```

使用できる形式は *key value* または *key=value* だけに限定されません。ただし、これらの形式に基づく検証に起因する制限事項を知っておく必要があります。

制限されている文字の回避

ライセンス：Protection

次の文字制限に注意してください。

- metadata キーワード内でセミコロン(;)やコロン(:)を使用しないでください。
- カンマを使用する場合には、複数の *key value* または *key=value* ステートメントの区切り文字としてカンマが解釈されることに注意してください。次に例を示します。

```
key value, key value, key value
```

- 等号(=)または空白文字を使用する場合には、それらの文字が *key* と *value* の間の区切り文字として解釈されることに注意してください。次に例を示します。

```
key value
key=value
```

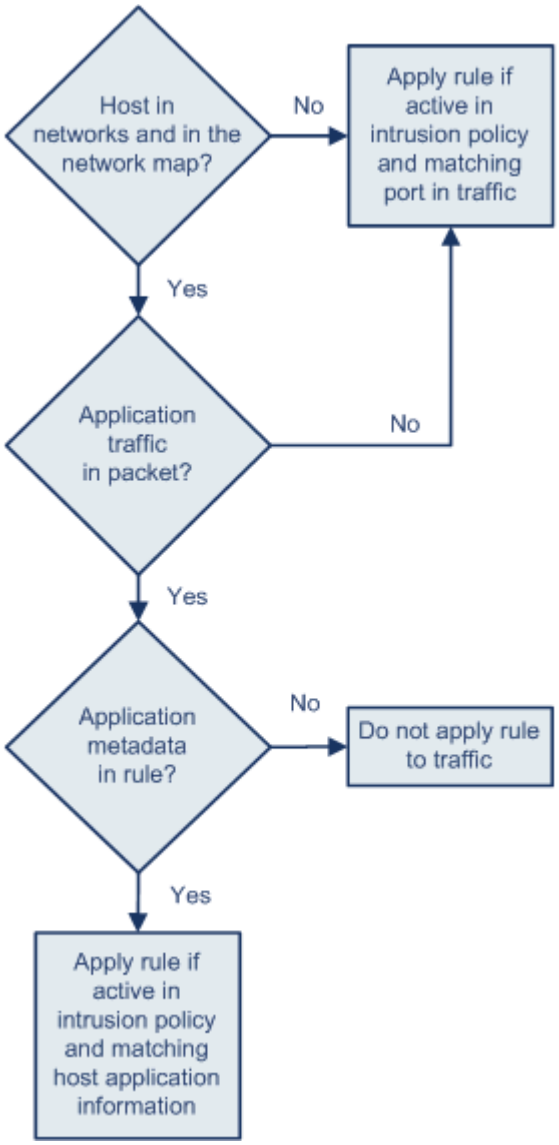
その他のすべての文字が使用可能です。

service メタデータの追加

ライセンス：Protection

ルール エンジンは、トラフィックを分析して処理するために、パケット内のホストに関するアプリケーション プロトコル情報に一致する service メタデータ付きのアクティブ ルールを適用します。これが一致しない場合、システムはルールをトラフィックに適用しません。ホストにアプリケーション プロトコル情報が存在しない場合、またはルールに service メタデータが含まれない場合、システムはルール内のポートに照らしてトラフィック内のポートを検査し、ルールをトラフィックに適用するかどうかを判断します。

次の図は、アプリケーション情報に基づくトラフィックとルールの照合を示しています。



アプリケーション プロトコルの識別によってルールを照合するには、`metadata` キーワードと *`key value`* ステートメントを定義する必要があります。その際、*`key`* として `service`、および *`value`* としてアプリケーションを指定します。たとえば、次に示す `metadata` キーワード内の *`key value`* ステートメントは、ルールを HTTP トラフィックに関連付けます。

```
service http
```

次の表では、最も一般的なアプリケーション値について説明します。

(注)　表に含まれないアプリケーションを定義するために支援が必要な場合は、サポート担当にお問い合わせください。

表 36-22　service 値

値	説明
dcerpc	分散コンピューティング環境/リモート プロシージャ コール システム
dns	ドメイン ネーム システム
finger	Finger ユーザ情報プロトコル
FTP	ファイル転送プログラム
ftp-data	ファイル転送プログラム(データ チャネル)
http	ハイパーテキスト転送プロトコル
imap	Internet Message Access Protocol
isakmp	Internet Security Association and Key Management Protocol
netbios-dgm	NETBIOS データグラム サービス
netbios-ns	NETBIOS ネーム サービス
netbios-ssn	NETBIOS セッション サービス
nntp	Network News Transfer Protocol
oracle	Oracle Net Services
pop2	Post Office Protocol バージョン 2
pop3	Post Office Protocol バージョン 3
smtp	Simple Mail Transfer Protocol
ssh	セキュア シェル ネットワーク プロトコル
telnet	Telnet ネットワーク プロトコル
tftp	トリビアル ファイル転送プロトコル
x11	X Window システム

予約済みメタデータの回避

ライセンス: Protection

metadata キーワードでは、次の単語を単一の引数として、または *key value* ステートメント内のキーとして使用しないでください。これらは VRT 用に予約されています。

```
application
engine
impact_flag
os
policy
rule-type
rule-flushing
soid
```

(注) ローカル ルールを適切に機能させるために制限付きメタデータをどうしても追加する必要がある場合は、サポート担当にお問い合わせください。詳細については、ローカル ルール ファイルのインポート(66-22 ページ)を参照してください。

メタデータを使用するルールの検索

ライセンス：Protection

`metadata` キーワードを使用するルールを検索するには、ルールの [検索（Search）] ページで `metadata` キーワードを選択して、オプションで、メタデータの一部分を入力します。たとえば次のように入力できます。

- `author` と入力すると、*`key`* として `author` が使用されているすべてのルールが表示されます。
- `author snortguru` と入力すると、*`key`* として `author`、*`value`* として `SnortGuru` がそれぞれ使用されているすべてのルールが表示されます。
- `author s` と入力すると、*`key`* として `author`、さらに *`value`* として `SnortGuru`、`SnortUser1`、`SnortUser2` などの語が使用されているすべてのルールが表示されます。

ヒント　*`key`* と *`value`* の両方を検索するときには、ルール内の *`key value`* ステートメントで使用されているのと同じ接続演算子（等号 [=] または空白文字）を検索で使用してください。*`key`* の後に等号（=）または空白文字のどちらを入力するかに応じて、異なる結果が検索で返されます。

なお、メタデータ追加のために使用する形式とは無関係に、システムはメタデータ検索語を *`key value`* または *`key=value`* ステートメントの全部または一部として解釈します。たとえば、次に示すメタデータは *`key value`* または *`key=value`* 形式に従っていませんが、有効なメタデータです。

```
ab cd ef gh
```

ただし、この例に含まれる各スペースは key と value の間の区切り文字としてシステムで解釈されます。次に示す並列語や単一語を検索で使用すると、この例のメタデータを含むルールを正しく検出できます。

```
cd ef
ef gh
ef
```

一方、次の検索を使用した場合、単一の *`key value`* ステートメントとしてシステムによって解釈されるため、ルールを検出できません。

```
ab ef
```

詳細については、ルールの検索（36-121 ページ）を参照してください。

影響レベル 1 の設定

ライセンス：Protection

次に示す予約済み *`key value`* ステートメントを `metadata` キーワードの中で使用できます。

```
impact_flag red
```

この *`key value`* ステートメントは、インポートしたローカル ルールまたはルール エディタを使って作成したカスタム ルールに関する影響フラグを赤（レベル 1）に設定します。

「送信元または宛先のホストがウイルス、トロイの木馬、その他の有害ソフトウェアによって侵害されている可能性があることを、ルールをトリガーしているパケットが示している」と判断した場合、VRT はシスコ提供のルールに `impact_flag red` ステートメントを含めます。詳細については、影響レベルを使用してイベントを評価する（41-41 ページ）を参照してください。

IP ヘッダー値の検査

ライセンス：Protection

キーワードを使用すると、パケットの IP ヘッダーの中で攻撃やセキュリティ ポリシー違反の可能性を識別できます。詳細については、次の各項を参照してください。

- フラグメント ビットと予約済みビットの検査（36-53 ページ）
- IP ヘッダー識別値の検索（36-54 ページ）
- 指定された IP オプションの識別（36-54 ページ）
- 指定された IP プロトコル番号の識別（36-55 ページ）
- パケットのタイプ オブ サービスの検査（36-55 ページ）
- パケットの存続可能時間値の検査（36-55 ページ）

フラグメント ビットと予約済みビットの検査

ライセンス：Protection

`fragbits` キーワードは、IP ヘッダーのフラグメント ビットと予約ビットを検査します。パケットごとに、予約ビット、More Fragments ビット、および Don't Fragment ビットを任意に組み合わせて検査できます。

表 36-23　Fragbits 引数の値

引数	説明
R	予約済みビット
M	More Fragments ビット
D	Don’t Fragment ビット

`fragbits` キーワードを使ってルールを微調整するために、次の表に示す演算子をルール内の引数値の後ろに指定できます。

表 36-24　Fragbit 演算子

演算子	説明
プラス記号（`+`）	パケットは、指定されたすべてのビットと一致する必要があります。
アスタリスク（`*`）	パケットは、指定されたどのビットと一致することもできます。
感嘆符（`!`）	指定されたどのビットも設定されていない場合、パケットが基準を満たします。

たとえば、（他のビットの有無とは無関係に）少なくとも予約済みビットが設定されたパケットに対してイベントを生成するには、`fragbits` 値として `R+` を使用します。

IP ヘッダー識別値の検索

ライセンス: Protection

`id` キーワードは、IP ヘッダーのフラグメント識別フィールドを検査して、キーワード引数で指定された値と照合します。一部のサービス拒否ツールやスキャナは、このフィールドに容易に検出できる特定の番号を設定します。たとえば、Synscan ポートスキャンを検出する SID 630 では、`id` 値が `39426`（スキャナから伝送されるパケットの ID 番号として使われる静的な値）に設定されます。

（注）　`id` 引数値は数値でなければなりません。

指定された IP オプションの識別

ライセンス: Protection

`IPopts` キーワードを使用すると、指定された IP ヘッダー オプションをパケット内で検索できます。次の表に、使用可能な引数値を示します。

表 36-25　IPoption 引数

引数	説明
rr	経路を記録
eol	リストの末尾
nop	オペレーションなし
ts	タイム スタンプ
sec	IP セキュリティ オプション
lsrr	厳密でない送信元ルーティング
ssrr	厳密な送信元ルーティング
satid	ストリーム識別子

アナリストが最も頻繁に監視するのは、厳密な送信元ルーティングと厳密でない送信元ルーティングです。これらのオプションは送信元 IP アドレスのスプーフィングを示している可能性があるためです。

指定された IP プロトコル番号の識別

ライセンス：Protection

ip_proto キーワードを使用すると、キーワードの値として指定された IP プロトコルを含むパケットを識別できます。IP プロトコルは 0 ～ 255 の数値として指定できます。プロトコル番号の完全なリストについては、http://www.iana.org/assignments/protocol-numbers を参照してください。これらの番号を、<、>、または ! 演算子と組み合わせることができます。たとえば、ICMP 以外のプロトコルを使用しているトラフィックを検査するには、ip_proto キーワードの値として !1 を使用します。1 つのルール内で ip_proto キーワードを複数回にわたって使用できます。ただし、ルール エンジンはキーワードの複数インスタンスをブール和関係（AND）と解釈することに注意してください。たとえば、ip_proto:!3; ip_proto:!6 を含むルールを作成した場合、このルールは GGP プロトコルおよび TCP プロトコルを使用するトラフィックを無視します。

パケットのタイプ オブ サービスの検査

ライセンス：Protection

一部のネットワークでは、ネットワーク上を移動するパケットの優先度を設定するタイプ オブ サービス（ToS）値が使用されます。tos キーワードを使用すると、キーワードの引数で指定された値に照らしてパケットの IP ヘッダーの ToS 値を検査できます。tos キーワードを使用するルールは、ToS が指定の値に設定され、しかもルール内の残りの基準を満たすパケットに対してトリガーとして使用されます。

（注）　tos の引数値は数値でなければなりません。

[ToS] フィールドは IP ヘッダー プロトコルでは非推奨になり、[Differentiated Services Code Point (DSCP)] フィールドに置き換えられています。

パケットの存続可能時間値の検査

ライセンス：Protection

パケットの存続可能時間（time-to-live、ttl）値は、パケットが破棄される前に生成できるホップ数を示します。ttl キーワードを使用すると、キーワードの引数として指定された値または値の範囲に照らしてパケットの IP ヘッダーの ttl 値を検査できます。ttl キーワード パラメータを 0 や 1 などの低い値に設定すると役立つことがあります。低い存続可能時間値がトレースルートや侵入を回避する試みを示している場合があるためです（ただし、このキーワードの適切な値は、管理対象デバイスの配置やネットワーク トポロジによって異なります）。次のように構文を使用します。

- TTL 値に特定の 1 つの値を設定するには、0 ～ 255 の整数を使用します。値の前に等号（=）を付けることもできます（たとえば 5 または =5 を指定できます）。
- TTL 値の範囲を指定するには、ハイフン（-）を使用します（たとえば、0-2 は 0 ～ 2 のすべての値、-5 は 0 ～ 5 のすべての値、5- は 5 ～ 255 のすべての値をそれぞれ指定します）。
- 特定の値より大きい TTL 値を指定するには、「大なり」記号（>）を使用します（たとえば、>3 は 3 より大きいすべての値を指定します）。
- 特定の値以上の TTL 値を指定するには、「大なりイコール」記号（>=）を使用します（たとえば、>=3 は 3 以上のすべての値を指定します）。

- 特定の値より小さい TTL 値を指定するには、「小なり」記号(<)を使用します(たとえば、<3 は 3 より小さいすべての値を指定します)。
- 特定の値以下の TTL 値を指定するには、「小なりイコール」記号(<=)を使用します(たとえば、<=3 は 3 以下のすべての値を指定します)。

ICMP ヘッダー値の検査

ライセンス: Protection

FireSIGHT システムでサポートされるキーワードを使用すると、ICMP パケットのヘッダー内の攻撃やセキュリティ ポリシー違反を識別できます。なお、ほとんどの ICMP タイプおよびコードを検出する事前定義ルールがあることに注意してください。既存のルールを有効にするか、既存のルールに基づいてローカル ルールを作成することを考慮してください。ICMP ルールを最初から作成するよりも、ニーズを満たすルールを見つける方が時間の節約になる可能性があります。

ICMP 固有のキーワードの詳細については、以下の項を参照してください。

- 静的な ICMP ID 値とシーケンス値の識別(36-56 ページ)
- ICMP メッセージ タイプの検査(36-56 ページ)
- ICMP メッセージ コードの検査(36-57 ページ)

静的な ICMP ID 値とシーケンス値の識別

ライセンス: Protection

ICMP の識別番号とシーケンス番号は、ICMP 応答と ICMP 要求を関連付けるうえで役立ちます。通常のトラフィックでは、これらの値はパケットに動的に割り当てられます。一部のコバート チャネルおよび Distributed Denial of Server(DDoS)プログラムは、静的な ICMP ID およびシーケンス値を使用します。次のキーワードを使用すると、静的な値を含む ICMP パケットを識別できます。

icmp_id

`icmp_id` キーワードは、ICMP エコー要求または応答パケットの ICMP ID 番号を検査します。ICMP ID 番号に対応する数値を `icmp_id` キーワードの引数として使用します。

icmp_seq

`icmp_seq` キーワードは、ICMP エコー要求または応答パケットの ICMP シーケンスを検査します。ICMP シーケンス番号に対応する数値を `icmp_seq` キーワードの引数として使用します。

ICMP メッセージ タイプの検査

ライセンス: Protection

`itype` キーワードを使用して、特定の ICMP メッセージ タイプ値を含むパケットを検索します。有効な ICMP タイプ値または無効な ICMP タイプ値を指定して、さまざまなタイプのトラフィックを検査できます(ICMP タイプ番号の完全なリストについては http://www.iana.org/assignments/icmp-parameters または http://www.faqs.org/rfcs/rfc792.html を参照してください)。たとえば、サービス拒否攻撃やフラッディング攻撃を発生させるために攻撃者が範囲外の ICMP タイプ値を設定することがあります。

「小なり」(<)と「大なり」(>)を使用して `itype` 引数値の範囲を指定できます。

次に例を示します。

- <35
- >36
- 3<>55

ヒント　ICMP タイプ番号の完全なリストについては、http://www.iana.org/assignments/icmp-parameters または http://www.faqs.org/rfcs/rfc792.html を参照してください。

ICMP メッセージ コードの検査

ライセンス：Protection

ICMP メッセージには、宛先が到達不能である場合の詳細を示すコード値が含まれることがあります。（ICMP メッセージ コードの完全なリストと、それらを使用できる関連するメッセージ タイプについては、http://www.iana.org/assignments/icmp-parameters の第 2 項を参照してください）。

`icode` キーワードを使用すると、特定の ICMP コード値を含むパケットを識別できます。有効な ICMP コード値と無効な ICMP コード値のいずれかを指定することにより、さまざまなタイプのトラフィックを検査できます。

「小なり」（<）と「大なり」（>）を使用して `icode` 引数値の範囲を指定できます。

次に例を示します。

- 35 より小さい値を検索するには `<35` と指定します。
- 36 より大きい値を検索するには `>36` と指定します。
- 3 ～ 55 の間にある値を検索するには、`3<>55` と指定します。

ヒント　`icode` キーワードと `itype` キーワードを一緒に使用すると、両方に一致するトラフィックを識別できます。たとえば、ICMP 宛先到達不能コード タイプと ICMP ポート到達不能コード タイプを含む ICMP トラフィックを特定するには、値 3 の `itype` キーワード（宛先到達不能）と、値 3 の `icode` キーワード（ポート到達不能）を指定します。

TCP ヘッダー値とストリーム サイズの検査

ライセンス：Protection

FireSIGHT システムでは、パケットの TCP ヘッダーと TCP ストリーム サイズを使って試行される攻撃を識別するためのキーワードを使用できます。TCP 固有のキーワードの詳細については、以下の項を参照してください。

- TCP 確認応答値の検査（36-58 ページ）
- TCP フラグ組み合わせの検査（36-58 ページ）
- TCP または UDP クライアントまたはサーバ フローへのルールの適用（36-59 ページ）
- 静的な TCP シーケンス番号の識別（36-60 ページ）
- 特定のサイズの TCP ウィンドウの識別（36-61 ページ）
- 特定のサイズの TCP ストリームの識別（36-61 ページ）

TCP 確認応答値の検査

ライセンス：Protection

ack キーワードを使用して、パケットの TCP 確認応答番号と特定の値を比較できます。パケットの TCP 確認応答番号が、ack キーワードに指定された値と一致した場合に、ルールがトリガーとして使用されます。

ack の引数値は数値でなければなりません。

TCP フラグ組み合わせの検査

ライセンス：Protection

flags キーワードを使用すると、複数の TCP フラグを任意に組み合わせて指定できます。検査対象のパケットでこれらが設定されている場合、ルールがトリガーとして使用されます。

（注）　従来、flags の値として A+ を使用していたケースでは、代わりに flow キーワードおよび値 established を使用してください。一般に、フラグのすべての組み合わせが検出されるようにするには、フラグの使用時に flow キーワードおよび値 stateless を使用する必要があります。flow キーワードの詳細については、TCP または UDP クライアントまたはサーバ フローへのルールの適用（36-59 ページ）を参照してください。

次の表に示す flags キーワードの値を確認または無視することができます。

表 36-26　flag の引数

引数	TCP フラグ
ACK	データを確認応答します。
Psh	このパケットでデータが送信される必要があります。
Syn	新しい接続。
Urg	パケットに緊急データが含まれています。
Fin	接続が閉じられました。
Rst	接続が異常終了しました。
CWR	ECN 輻輳ウィンドウが減少しました。旧 R1 引数（下位互換性を維持するために引き続きサポートされています）。
ECE	ECN エコー。旧 R2 引数（下位互換性を維持するために引き続きサポートされています）。

ヒント　明示的輻輳通知（ECN）の詳細については、http://www.faqs.org/rfcs/rfc3168.html の情報を参照してください。

flags キーワードを使用する場合、複数のフラグに対する照合方法をシステムに指示するための演算子を使用できます。次の表に、これらの演算子の説明を示します。

表 36-27　flags と一緒に使用する演算子

演算子	説明	例
すべて	パケットは、指定されたすべてのフラグを含んでいる必要があります。	Urg と all を選択すると、パケットが緊急フラグを含んでいる必要があること、および他のフラグが含まれる可能性があることを指定できます。
任意	パケットは、指定された任意のフラグを含むことができます。	Ack、Psh、および any を選択すると、ルールをトリガーとして使用するためには Ack と Psh のどちらか(または両方)のフラグが設定される必要があること、およびパケット内で他のフラグも設定されている可能性があることを指定できます。
ノット	パケットは、指定されたフラグ セットを含んでは**なりません**。	Urg と not を選択すると、このルールをトリガーとして使用するパケットに関して緊急フラグが設定**されない**ことを指定できます。

TCP または UDP クライアントまたはサーバ フローへのルールの適用

ライセンス:Protection

flow キーワードを使用すると、セッション特性に基づいてルールで検査されるパケットを選択できます。flow キーワードを使用することで、ルールの適用対象となるトラフィック フロー方向を指定して、クライアント フローとサーバ フローのどちらかにルールを適用できます。flow キーワードによるパケット検査の方法を指定するには、分析すべきトラフィックの方向、検査するパケットの状態、およびパケットが再構築ストリームの一部かどうかを設定できます。

ルールの処理時に、パケットのステートフル インスペクションが実行されます。ステートレス トラフィック(セッション コンテキストが確立されていないトラフィック)を TCP ルールで無視するには、flow キーワードをルールに追加して、そのキーワードで **Established** 引数を選択する必要があります。UDP ルールでステートレス トラフィックを無視するには、flow キーワードをルールに追加して、**Established** 引数と方向引数のどちらか(または両方)を選択する必要があります。これにより、TCP または UDP ルールでパケットのステートフル インスペクションが実行されます。

方向引数を追加した場合、ルール エンジンは、指定された方向と一致するフローを伴う確立された状態のパケットだけを検査します。たとえば、TCP または UDP 接続が検出されたときトリガーとして使用されるルールに、flow キーワードおよび established 引数と From Client 引数を追加した場合、ルール エンジンはクライアントから送信されたパケットだけを検査します。

ヒント　パフォーマンスを最大にするには、必ず TCP ルールまたは UDP セッション ルールに flow キーワードを含めてください。

フローを指定するには、[ルールの作成(Create Rule)] ページの [検出オプション(Detection Options)] リストで flow キーワードを選択し、[オプションの追加(Add Option)] をクリックします。次に、フィールドごとに表示されるリストから引数を選択します。

次の表に、flow キーワードで指定できるストリーム関連引数の説明を示します。

表 36-28　状態に関連する flow 引数

引数	説明
Established	確立された接続でトリガーとして使用されます。
Stateless	ストリーム プロセッサの状態に関係なくトリガーとして使用されます。

次の表に、flow キーワードで指定できる方向オプションの説明を示します。

表 36-29　flow の方向引数

引数	説明
To Client	サーバ応答でトリガーとして使用されます。
To Server	クライアント応答でトリガーとして使用されます。
From Client	クライアント応答でトリガーとして使用されます。
From Server	サーバ応答でトリガーとして使用されます。

From Server と To Client の機能が同じであること、および To Server と From Client の機能も同じであることに注意してください。これらのオプションは、ルールに文脈と読みやすさを加味するために提供されています。たとえば、サーバからクライアントへの攻撃を検出するよう設計されたルールを作成する場合は、From Server を使用します。一方、クライアントからサーバへの攻撃を検出するように設計されたルールを作成する場合は、From Client を使用します。

次の表に、flow キーワードで指定できるストリーム関連引数の説明を示します。

表 36-30　flow のストリーム関連引数

引数	説明
Ignore Stream Traffic	再構築されたストリーム パケットでトリガーとして使用されません。
Only Stream Traffic	再構築されたストリーム パケットでのみトリガーとして使用されます。

たとえば、flow キーワードの値として To Server, Established, Only Stream Traffic を使用すると、ストリーム プリプロセッサで再構築された、確立済みセッションでクライアントからサーバに移動するトラフィックを検出できます。

静的な TCP シーケンス番号の識別

ライセンス：Protection

seq キーワードを使用すると、静的なシーケンス番号値を指定できます。パケットのシーケンス番号が、指定された引数と一致する場合、そのキーワードを含むルールがトリガーとして使用されます。このキーワードはあまり使用されませんが、静的シーケンス番号付きの生成済みパケットを使用する攻撃やネットワーク スキャンを識別するうえでこれが役立ちます。

特定のサイズの TCP ウィンドウの識別

ライセンス：Protection

window キーワードを使用すると、特定の TCP ウィンドウ サイズを指定できます。このキーワードを含むルールは、指定された TCP ウィンドウ サイズのパケットが検出されるたびにトリガーされます。このキーワードはあまり使用されませんが、静的 TCP ウィンドウ サイズ付きの生成済みパケットを使用する攻撃やネットワーク スキャンを識別するうえでこれが役立ちます。

特定のサイズの TCP ストリームの識別

ライセンス：Protection

次に示す形式で、stream_size キーワードとストリーム プリプロセッサを組み合わせて使用すると、TCP ストリームのサイズをバイト単位で特定できます。

direction,*operator*,*bytes*

ここで、*bytes* はバイト数です。引数内の各オプションをカンマ(,)で区切る必要があります。

次の表は、stream_size キーワードで指定できる大文字と小文字を区別しない方向オプションを示しています。

表 36-31　stream_size キーワードの方向引数

引数	説明
client	指定されたストリーム サイズに一致するクライアントからのストリームでトリガーとして使用されます。
server	指定されたストリーム サイズに一致するサーバからのストリームでトリガーとして使用されます。
both	指定されたストリーム サイズに一致するクライアントからのトラフィックとサーバからのトラフィックの両方によってトリガーとして使用されます。 たとえば both, >, 200 という引数は、クライアントからのトラフィックが 200 バイトを超え、しかもサーバからのトラフィックが 200 バイトを超えている場合にトリガーとして使用されます。
either	指定されたストリーム サイズに一致するクライアントまたはサーバからのトラフィック(どちらか先に出現した方)によってトリガーとして使用されます。 たとえば either, >, 200 という引数は、クライアントからのトラフィックが 200 バイトを超えている、またはサーバからのトラフィックが 200 バイトを超えている場合にトリガーとして使用されます。

次の表に、stream_size キーワードで使用できる演算子の説明を示します。

表 36-32　stream_size キーワードの引数演算子

演算子	説明
=	次の値と等しい
!=	等しくない
>	より大きい
<	より少ない

表 36-32　stream_size キーワードの引数演算子（続き）

演算子	説明
>=	右辺と比較して大きいか等しい
<=	右辺と比較して小さいか等しい

たとえば、クライアントからサーバに移動する 5001216 バイト以上の TCP ストリームを検出するには、stream_size キーワードの引数として client, >=, 5001216 を使用できます。

TCP ストリーム再構築の有効化と無効化

ライセンス：Protection

stream_reassemble キーワードを使用すると、接続での検査対象トラフィックがルールの条件と一致した場合に、1 つの接続の TCP ストリーム再構築を有効/無効にすることができます。オプションで、このキーワードを 1 つのルール内で複数回使用することができます。

ストリーム再構築を有効または無効にするには、次の構文を使用します。

```
enable|disable, server|client|both, option, option
```

次の表に、stream_reassemble キーワードで使用できるオプション引数の説明を示します。

表 36-33　stream_reassemble のオプション引数

引数	説明
noalert	ルールで他にどの検出オプションが指定されているかに関係なく、イベントを生成しません。
fastpath	一致の検出時に残りの接続トラフィックを無視します。

たとえば、次のルールは、HTTP 応答で 200 OK ステータス コードが検出された接続に対してイベントを生成せずに、TCP クライアント側ストリームの再構築を無効にします。

```
alert tcp any 80 -> any any (flow:to_client, established; content: "200 OK";
stream_reassemble:disable, client, noalert
```

stream_reassemble を使用するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [stream_reassemble] を選択して、[オプションの追加（Add option）] をクリックします。

[stream_reassemble] セクションが表示されます。

セッションからの SSL 情報の抽出

ライセンス：Protection

SSL ルール キーワードを使用すると、Secure Sockets Layer（SSL）プリプロセッサを呼び出し、暗号化セッションのパケットから SSL のバージョンとセッション状態に関する情報を抽出できます。

SSL または Transport Layer Security（TLS）を使用する暗号化セッションを確立するためにクライアントとサーバが通信するとき、ハンドシェイク メッセージが交換されます。セッション中に伝送されるデータは暗号化されますが、ハンドシェイク メッセージは暗号化されません。

SSL プリプロセッサは、特定のハンドシェイク フィールドから状態とバージョンの情報を抽出します。ハンドシェイク内の 2 つのフィールドは、セッション暗号化に使われる SSL または TLS のバージョンとハンドシェイクのステージを示します。

詳細については、次の項を参照してください。

- ssl_state（36-63 ページ）
- ssl_version（36-64 ページ）

ssl_state

ライセンス：Protection

`ssl_state` キーワードを使用すると、暗号化されたセッションの状態情報と照合することができます。同時に使用される複数の SSL バージョンを検査するには、1 つのルール内で複数の `ssl_version` キーワードを使用します。

ルールで `ssl_state` キーワードが使用されている場合、ルール エンジンは SSL プリプロセッサを呼び出して、トラフィック内の SSL 状態情報を検査します。

たとえば、チャレンジ長が非常に長く、データが多すぎる `ClientHello` メッセージを送信することによってサーバ上のバッファ オーバーフローを引き起こそうとする攻撃者の試みを検出するには、`ssl_state` キーワードと引数 `client_hello` を使用し、異常に大きなパケットを検査することができます。

SSL 状態に関する複数の引数を指定するには、カンマ区切りリストを使用します。複数の引数を列挙した場合、システムは OR 演算子を使ってそれらを評価します。たとえば、引数として `client_hello` および `server_hello` を指定すると、システムは `client_hello` または `server_hello` のどちらかを含むトラフィックに照らしてルールを評価します。

次のように、引数を除外することもできます。

```
!client_hello, !unknown
```

接続が一連の状態のそれぞれに到達したことを確認するには、ssl_state ルール オプションを使用する複数のルールを使う必要があります。`ssl_state` キーワードは、次の識別子を引数として受け入れます。

表 36-34　`ssl_state` の引数

引数	目的
`client_hello`	クライアントが暗号化セッションを要求する、メッセージ タイプ `ClientHello` のハンドシェイク メッセージを照合します。
`server_hello`	クライアントからの暗号化セッション要求に対してサーバが応答する、メッセージ タイプ `ServerHello` のハンドシェイク メッセージを照合します。
`client_keyx`	サーバからのキーの受信を確認するためにクライアントがサーバにキーを伝送する、メッセージ タイプ `ClientKeyExchange` のハンドシェイク メッセージを照合します。
`server_keyx`	サーバからのキーの受信を確認するためにクライアントがサーバにキーを伝送する、メッセージ タイプ `ServerKeyExchange` のハンドシェイク メッセージを照合します。
`unknown`	任意のハンドシェイク メッセージ タイプを照合します。

ssl_version

ライセンス：Protection

ssl_version キーワードを使用すると、暗号化セッションのバージョン情報を照合できます。ルールで ssl_version キーワードが使用されている場合、ルール エンジンは SSL プリプロセッサを呼び出して、トラフィック内の SSL バージョン情報を検査します。

たとえば、SSL バージョン 2 にバッファ オーバーフロー脆弱性があることがわかっている場合、ssl_version キーワードで sslv2 引数を使用して、その SSL バージョンを使用するトラフィックを識別できます。

SSL バージョンに関する複数の引数を指定するには、カンマ区切りリストを使用します。複数の引数を列挙した場合、システムは OR 演算子を使ってそれらを評価します。たとえば、SSLv2 を使用していない暗号化トラフィックを識別するには、ssl_version:ssl_v3,tls1.0,tls1.1,tls1.2 をルールに追加できます。このルールは、SSL バージョン 3、TLS バージョン 1.0、TLS バージョン 1.1、または TLS バージョン 1.2 を使用するトラフィックを評価します。

ssl_version キーワードは、次の SSL/TLS バージョン識別子を引数として受け入れます。

表 36-35　ssl_version の引数

引数	目的
sslv2	Secure Sockets Layer（SSL）バージョン 2 を使用してエンコードされたトラフィックを照合します。
sslv3	Secure Sockets Layer（SSL）バージョン 3 を使用してエンコードされたトラフィックを照合します。
tls1.0	Transport Layer Security（TLS）バージョン 1.0 を使用してエンコードされたトラフィックを照合します。
tls1.1	Transport Layer Security（TLS）バージョン 1.1 を使用してエンコードされたトラフィックを照合します。
tls1.2	Transport Layer Security（TLS）バージョン 1.2 を使用してエンコードされたトラフィックを照合します。

アプリケーション層プロトコル値の検査

ライセンス：Protection

アプリケーション層プロトコル値の正規化と検査はプリプロセッサによってほとんど実行されますが、以下の項で説明するキーワードを使用すると、アプリケーション層値をさらに検査できます。

- RPC（36-65 ページ）
- ASN.1（36-65 ページ）
- urilen（36-66 ページ）
- DCE/RPC キーワード（36-67 ページ）
- SIP キーワード（36-71 ページ）
- GTP キーワード（36-73 ページ）
- Modbus キーワード（36-83 ページ）
- DNP3 キーワード（36-85 ページ）

RPC

ライセンス：Protection

rpc キーワードは、TCP または UDP パケットのオープン ネットワーク コンピューティング リモート プロシージャ コール（ONC RPC）サービスを識別します。これにより、ホスト上の RPC プログラムの識別試行を検出することができます。ネットワークで実行中のいずれかの RPC サービスを悪用できるかどうか判断するために、侵入者は RPC ポートマッパーを使用できます。また、ポートマッパーを使用せずに RPC を実行中の他のポートへのアクセスを試みることもできます。次の表に、rpc キーワードで使用できる引数を列挙します。

表 36-36　rpc キーワードの引数

引数	説明
アプリケーション	RPC アプリケーション番号
手順	呼び出される RPC プロシージャ
version	RPC バージョン

rpc キーワードの引数を指定するには、次の構文を使用します。

application,*procedure*,*version*

ここで、*application* は RPC アプリケーション番号、*procedure* は RPC プロシージャ番号、*version* は RPC バージョン番号です。rpc キーワードのすべての引数を指定する必要があります。引数のいずれかを指定できない場合は、アスタリスク（*）で置き換えてください。

たとえば、任意のプロシージャまたはバージョンの RPC ポートマッパー（100000 という番号で示される RPC アプリケーション）を検索するには、引数として 100000,*,* を使用します。

ASN.1

ライセンス：Protection

asn1 キーワードを使用すると、さまざまな有害エンコードを検索しながら、パケットまたはパケットの一部分をデコードできます。

次の表に、asn1 キーワードの引数について説明します。

表 36-37　asn.1 キーワードの引数

引数	説明
Bitstring Overflow	無効な、リモートで悪用可能なビットストリング エンコードを検出します。
Double Overflow	標準バッファより大きい二重 ASCII エンコードを検出します。これは、Microsoft Windows の悪用可能な機能であることが分かっていますが、どのサービスが悪用可能かは現時点では不明です。
Oversize Length	指定された引数より大きい ASN.1 タイプ長を検出します。たとえば Oversize Length を 500 に設定した場合、500 を上回る ASN.1 タイプによってルールがトリガーとして使用されます。

表 36-37　asn.1 キーワードの引数（続き）

引数	説明
Absolute Offset	パケット ペイロードの先頭からの絶対オフセットを設定します（offset カウンタがバイト 0 から始まることに注意してください）。たとえば SNMP パケットをデコードするには、Absolute Offset を 0 に設定し、Relative Offset を設定しません。Absolute Offset として正または負の値が可能です。
Relative Offset	これは、最後に見つかったコンテンツ一致、pcre、または byte_jump からの相対オフセットです。コンテンツ "foo" の直後の ASN.1 シーケンスをデコードするには、Relative Offset を 0 に設定し、Absolute Offset を設定しません。Relative Offset として正または負の値が可能です。（オフセット カウンタが 0 から始まることに注意してください。）

たとえば、Microsoft ASN.1 ライブラリにおける既知の脆弱性ではバッファ オーバーフローが発生し、攻撃者は特別に細工した認証パケットを使ってその状態を悪用できます。システムが asn.1 データをデコードするとき、パケット内のエクスプロイト コードは、システム レベル権限付きでホスト上で動作したり、DoS 状態を引き起こしたりすることができます。次のルールは、asn1 キーワードを使用して、この脆弱性を悪用する試みを検出します。

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(flow:to_server, established; content:"|FF|SMB|73|"; nocase;
offset:4; depth:5;
asn1:bitstring_overflow,double_overflow,oversize_length
100,relative_offset 54;)
```

上記のルールの場合、任意のポートおよび $EXTERNAL_NET 変数で定義された任意の IP アドレスから発信され、ポート 445 を使用する $HOME_NET 変数で定義された任意の IP アドレスに向かう TCP トラフィックに対して、イベントが生成されます。加えて、サーバへの TCP 接続が確立された時点でのみルールを実行します。その後、ルールは特定の位置にある特定のコンテンツを検査します。最後に、ルールは asn1 キーワードを使用して、ビットストリング エンコードと二重 ASCII エンコードを検出し、最後に見つかったコンテンツ一致の末尾から 55 バイト目以降、長さ 100 バイトを超える asn.1 タイプ長を識別します（offset カウンタがバイト 0 から始まることに注意してください。）

urilen

ライセンス：Protection

urilen キーワードと HTTP Inspect プリプロセッサを組み合わせて使用すると、特定の長さ、最大長を下回る、最小長を上回る、または指定された範囲内の URI を HTTP トラフィック内で検査できます。

HTTP Inspect プリプロセッサがパケットを正規化して検査した後、ルール エンジンはルールに照らしてそのパケットを評価し、urilen キーワードで指定された長さ条件に URI が一致するかどうか判断します。このキーワードを使用すると、URI 長の脆弱性を悪用しようとする試みを検出できます。たとえばバッファ オーバーフローを発生させて、攻撃者が DoS 状態を引き起こしたり、システム レベル権限付きでホスト上でコードを実行したりしようと試みる可能性があります。

ルール内で urilen キーワードを使用するときには、次の点に注意してください。

- 必ず flow:established キーワードおよび他の 1 つ以上のキーワードを組み合わせて、urilen キーワードを使用してください。
- ルール プロトコルは常に TCP です。詳細については、プロトコルの指定（36-5 ページ）を参照してください。
- ターゲット ポートは常に HTTP ポートです。詳細については、侵入ルールでのポートの定義（36-9 ページ）と定義済みのデフォルトの変数の最適化（3-20 ページ）を参照してください。

URI 長を指定するときには、10 進のバイト数、「小なり」（<）、および「大なり」（>）を使用します。

次に例を示します。

- 5 バイト長の URI を検出するには、5 を指定します。
- 5 バイト長を下回る URI を検出するには、< 5 （1 つの空白文字で区切る）を指定します。
- 5 バイト長を上回る URI を検出するには、> 5（1 つの空白文字で区切る）を指定します。
- 3 ～ 5 バイト長の URI を検出するには、3 <> 5（<> の前後に空白文字を 1 つずつ含む）を指定します。

たとえば、Novell の eDirectory バージョン 8.8 に付属のサーバ モニタリングおよび診断ユーティリティ iMonitor バージョン 2.4 に脆弱性があることが知られています。長すぎる URI を含むパケットはバッファ オーバーフローを発生させます。攻撃者はそのような状況を悪用して、特別に細工したパケットをシステム レベル権限によりホスト上で実行したり、そのようなパケットで DoS 状態を引き起こしたりします。次のルールは、urilen キーワードを使用して、この脆弱性を悪用する試みを検出します。

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS
(msg:"EXPLOIT eDirectory 8.8 Long URI iMonitor buffer
overflow attempt";flow:to_server,established;
urilen:> 8192; uricontent:"/nds/"; nocase;
classtype:attempted-admin; sid:x; rev:1;)
```

上記のルールの場合、任意のポートおよび $EXTERNAL_NET 変数で定義された任意の IP アドレスから発信され、$HTTP_PORTS 変数で定義されたポートを使用して、$HOME_NET 変数で定義された任意の IP アドレスに向かう TCP トラフィックに対して、イベントが生成されます。加えて、サーバへの TCP 接続が確立された時点でのみ、パケットがルールに照らして評価されます。ルールは、urilen キーワードを使用して、長さ 8192 バイトを超える URI を検出します。最後に、ルールは URI を検索して、大文字と小文字を区別しない特定のコンテンツ /nds/ を検索します。

DCE/RPC キーワード

ライセンス：Protection

次の表に記載された 3 つの DCE/RPC キーワードを使用すると、エクスプロイトの DCE/RPC セッション トラフィックをモニタすることができます。これらのキーワードを含むルールを処理するとき、システムは DCE/RPC プリプロセッサを呼び出します。詳細については、DCE/RPC トラフィックのデコード（27-2 ページ）を参照してください。

表 36-38　DCE/RPC キーワード

使用するフィルタ	使用方法	検出対象
dce_iface	単独	特定の DCE/RPC サービスを識別するパケット
dce_opnum	dce_iface の後ろ	特定の DCE/RPC サービス オペレーションを識別するパケット
dce_stub_data	dce_iface + dce_opnum の後ろ	特定の処理要求または応答を定義するスタブ データ

表に示されているように、dce_opnum の前に必ず dce_iface を配置し、dce_stub_data の前に必ず dce_iface + dce_opnum を配置する必要があることに注意してください。

また、これらの DCE/RPC キーワードを他のルール キーワードと組み合わせて使用することもできます。DCE/RPC ルールでは、**DCE/RPC** 引数が選択された状態で byte_jump、byte_test、byte_extract の各キーワードを使用することに注意してください。詳細については、Byte_Jump と Byte_Test の使用（36-34 ページ）およびパケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。

シスコでは、DCE/RPC キーワードを含むルールに 1 つ以上の content キーワードを含めることを推奨しています。こうすると、ルール エンジンが常に高速パターン マッチ機能を使用することで処理速度が上がり、パフォーマンスが向上します。ルールに 1 つ以上の content キーワードが含まれている場合は、content キーワードの **Use Fast Pattern Matcher** 引数が有効になっているかどうかに関係なく、ルール エンジンが高速パターン マッチ機能を使用することに注意してください。詳細については、コンテンツ一致の検索（36-16 ページ）と高速パターン マッチ機能を使用（Use Fast Pattern Matcher）（36-30 ページ）を参照してください。

次のケースでは、DCE/RPC バージョンおよび隣接ヘッダー情報を一致コンテンツとして使用できます。

- ルールに他の content キーワードが含まれていない
- ルールにもう 1 つ content キーワードが含まれているが、DCE/RPC バージョンおよび隣接情報が、他方の content よりも特有のパターンを表している

  たとえば、DCE/RPC バージョンおよび隣接情報は通常、1 バイトのコンテンツよりも特有です。

次に示すバージョンおよび隣接情報コンテンツ一致のいずれか 1 つを使用して、ルール限定を終了する必要があります。

- コネクション型 DCE/RPC ルールでは、コンテンツ |05 00 00|（メジャー バージョン 05、マイナー バージョン 00、および要求 PDU（プロトコル データ ユニット）タイプ 00）を使用します。
- コネクションレス型 DCE/RPC ルールでは、コンテンツ |04 00|（バージョン 04、要求 PDU タイプ 00）を使用します。

いずれの場合も、DCE/RPC プリプロセッサで完了済みの処理を繰り返すことなく高速パターン マッチ機能を呼び出すために、ルール内の最後のキーワードとしてバージョンおよび隣接情報の content キーワードを配置してください。ルールの末尾に配置される content キーワードは、高速パターン マッチ機能を呼び出す手段として使われるバージョン コンテンツに当てはまりますが、ルール内の他のコンテンツ一致には必ずしも当てはまらないことに注意してください。

詳細については、次の各項を参照してください。

- dce_iface（36-69 ページ）
- dce_opnum（36-70 ページ）
- dce_stub_data（36-70 ページ）

dce_iface

ライセンス：Protection

dce_iface キーワードを使用すると、特定の DCE/RPC サービスを識別できます。

オプションで、dce_iface キーワードを dce_opnum キーワードおよび dce_stub_data キーワードと組み合わせて使用すると、検査する DCE/RPC トラフィックをさらに限定することができます。詳細については、dce_opnum（36-70 ページ）と dce_stub_data（36-70 ページ）を参照してください。

固定型 16 バイト Universally Unique Identifier（UUID）は、それぞれの DCE/RPC サービスに割り当てられているアプリケーション インターフェイスを識別します。たとえば、UUID 4b324fc8-670-01d3-1278-5a47bf6ee188 は、srvsvc サービスとしても知られる DCE/RPC lanmanserver サービスを識別します。このサービスは、ピアツーピア プリンタ、ファイル、および SMB 名前付きパイプを共有するためのさまざまな管理機能を提供します。DCE/RPC プリプロセッサは UUID および関連するヘッダー値を使用して DCE/RPC セッションを追跡します。

インターフェイス UUID は、次のように、ハイフンで区切られた 5 つの 16 進文字列で構成されます。

```
<4hexbytes>-<2hexbytes>-<2hexbytes>-<2hexbytes>-<6hexbytes>
```

次に示す netlogon インターフェイスの UUID のように、ハイフンを含む UUID 全体を入力することで、インターフェイスを指定します。

```
12345678-1234-abcd-ef00-01234567cffb
```

UUID 内の最初の 3 つの文字列はビッグ エンディアン バイト順で指定される必要があることに注意してください。通常、公開されたインターフェイス リストやプロトコル アナライザには UUID が正しいバイト順で表示されますが、それを入力する前に UUID バイト順を変更しなければならない場合もあります。次に示すメッセンジャー サービス UUID の場合、リトル エンディアン バイト順の最初の 3 つの文字列を含む未加工 ASCII テキストで表示されることがあります。

```
f8 91 7b 5a 00 ff d0 11 a9 b2 00 c0 4f b6 e6 fc
```

この同じ UUID を dce_iface キーワードに指定するには、次のようにハイフンを挿入し、最初の 3 つの文字列をビッグ エンディアン バイト順で配置できます。

```
5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc
```

1 つの DCE/RPC セッションに複数のインターフェイスへの要求を含めることができますが、1 つのルールには 1 つの dce_iface キーワードだけを含めてください。追加のインターフェイスを検出するには、追加のルールを作成します。

DCE/RPC アプリケーション インターフェイスにはインターフェイス バージョン番号も割り当てられます。オプションで、インターフェイス バージョンを指定できます。その際、バージョンが指定値に等しい、等しくない、指定値より小さい、または大きいことを示す演算子を使用します。

TCP セグメンテーションや IP フラグメンテーションに加えて、コネクション型とコネクションレス型の両方の DCE/RPC をフラグメント化することができます。通常、先頭以外の DCE/RPC フラグメントを指定のインターフェイスに関連付けるのはあまり効率的ではありません。このようにすると、多数の誤検出が発生する可能性があります。ただし、柔軟性を維持するために、オプションで、指定されたインターフェイスに照らしてすべてのフラグメントを評価できます。

次の表に、dce_iface キーワードの引数を要約します。

表 36-39　dce_iface の引数

引数	説明
Interface UUID	DCE/RPC トラフィック内で検出対象となる特定のサービスのアプリケーション インターフェイスを識別する、ハイフンを含む UUID。指定されたインターフェイスに関連付けられた任意の要求がインターフェイス UUID に一致します。
Version	オプションで、アプリケーション インターフェイス バージョン番号 0 ～ 65535 と、検出対象のバージョンが指定値より大きい(>)、小さい(<)、等しい(=)、または等しくない(!)を示す演算子。
All Fragments	オプションで、関連するすべての DCE/RPC フラグメント内のインターフェイスの照合、およびインターフェイス バージョン(指定されている場合)での照合を有効にします。この引数はデフォルトで無効になっています。これは、最初のフラグメントまたはフラグメント化されていないパケット全体が指定のインターフェイスに関連付けられている場合にのみ、キーワードが一致することを意味します。この引数を有効にすると、誤検出が発生する可能性があることに注意してください。

dce_opnum

ライセンス: Protection

dce_opnum キーワードを DCE/RPC プリプロセッサと組み合わせて使用すると、DCE/RPC サービスが提供する 1 つ以上の特定のオペレーションを識別するパケットを検出できます。

クライアント関数呼び出しは、DCE/RPC 仕様で「*オペレーション*」と呼ばれる特定のサービス関数を要求します。オペレーション番号(opnum)は DCE/RPC ヘッダー内の特定のオペレーションを識別します。エクスプロイトは特定のオペレーションを標的にすることがあります。

たとえば UUID 12345678-1234-abcd-ef00-01234567cffb は、数十種類のオペレーションを提供する netlogon サービスのインターフェイスを識別します。その 1 つがオペレーション 6 (NetrServerPasswordSet オペレーション)です。

オペレーション用のサービスを識別するには、dce_opnum キーワードの前に dce_iface キーワードを指定する必要があります。詳細については、dce_iface(36-69 ページ)を参照してください。

特定のオペレーションを示す 1 つの 10 進数値(0 ～ 65535)、ハイフンで区切ったオペレーション範囲、またはカンマで区切ったオペレーション/範囲のリストを任意の順序で指定できます。

次の例は、すべて有効な netlogon オペレーション番号を表しています。

```
15
15-18
15, 18-20
15, 20-22, 17
15, 18-20, 22, 24-26
```

dce_stub_data

ライセンス: Protection

dce_stub_data キーワードを DCE/RPC プリプロセッサと組み合わせて使用すると、他のルール オプションとは無関係に、スタブ データの先頭からインスペクションを開始するようルール エンジンに指示できます。dce_stub_data キーワードの後に続くパケット ペイロード ルール オプションは、スタブ データ バッファを基準にして適用されます。

DCE/RPC スタブ データは、クライアント プロシージャ コールと DCE/RPC ランタイム システム（DCE/RPC の中核をなすルーチンとサービスを提供するメカニズム）の間にインターフェイスを提供します。DCE/RPC エクスプロイトは、DCE/RPC パケットのスタブ データ部分で識別されます。スタブ データは特定のオペレーションまたは関数呼び出しに関連付けられているため、必ず `dce_stub_data` の前に `dce_iface` と `dce_opnum` を指定して、関連するサービスとオペレーションを識別してください。

`dce_stub_data` キーワードには引数がありません。詳細については、dce_iface（36-69 ページ）と dce_opnum（36-70 ページ）を参照してください。

SIP キーワード

ライセンス：Protection

4 つの SIP キーワードを使用すると、SIP セッション トラフィックでエクスプロイトを監視できます。

SIP プロトコルはサービス拒否（DoS）攻撃に対して脆弱であることに注意してください。このような攻撃に対処するルールでは、レート ベース攻撃の防止を活用できます。詳細については、動的ルール状態の追加（32-34 ページ）と レート ベース攻撃の防止（34-10 ページ）を参照してください。

詳細については、次の各項を参照してください。

- sip_header（36-71 ページ）
- sip_body（36-71 ページ）
- sip_method（36-72 ページ）
- sip_stat_code（36-72 ページ）

sip_header

ライセンス：Protection

`sip_header` キーワードを使用すると、抽出された SIP 要求または応答ヘッダーの先頭から検査を開始し、検査対象をヘッダー フィールドに限定することができます。

`sip_header` キーワードには引数がありません。詳細については、「sip_method（36-72 ページ）」と「sip_stat_code（36-72 ページ）」を参照してください。

次の例のルール フラグメントは SIP ヘッダーを指し示し、CSeq ヘッダー フィールドに一致します。

```
alert udp any any -> any 5060 ( sip_header; content:"CSeq"; )
```

sip_body

ライセンス：Protection

`sip_body` キーワードを使用すると、抽出された SIP 要求または応答メッセージ本文の先頭から検査を開始し、検査対象をメッセージ本文に限定することができます。

`sip_body` キーワードには引数がありません。

次の例のルール フラグメントは SIP メッセージ本文を指し示し、抽出された SDP データの c（接続情報）フィールド内の特定の IP アドレスに一致します。

```
alert udp any any -> any 5060 ( sip_body; content:"c=IN 192.168.12.14"; )
```

ルールが SDP コンテンツの検索だけに限定されないことに注意してください。SIP プリプロセッサはメッセージ本文全体を抽出し、それをルール エンジンで使用できるようにします。

sip_method

ライセンス：Protection

各 SIP 要求内の *method* フィールドは要求の目的を識別します。`sip_method` キーワードを使用すると、SIP 要求の中で特定のメソッドを検査することができます。複数のメソッドはカンマで区切ります。

次に示す現在定義されている SIP メソッドを指定できます。

```
ack, benotify, bye, cancel, do, info, invite, join, message, notify, options, prack,
publish, quath, refer, register, service, sprack, subscribe, unsubscribe, update
```

メソッドでは大文字と小文字が区別されません。複数のメソッドをカンマで区切ることができます。

今後、新しい SIP メソッドが定義される可能性があるため、カスタム メソッド、つまり現在定義されている SIP メソッド以外のメソッドを指定することもできます。可能なフィールド値は RFC 2616 で定義されています。=、(、) などの制御文字と区切り文字を除いて、すべての文字を使用できます。除外されている区切り文字の完全なリストについては、RFC 2616 を参照してください。指定されたカスタム メソッドがトラフィックで検出されると、システムはパケット ヘッダーを検査しますが、メッセージは検査されません。

システムでは最大 32 個のメソッド（現在定義されている 21 個のメソッドと追加の 11 個のメソッド）がサポートされます。システムは、設定される未定義のメソッドをすべて無視します。合計 32 個のメソッドには、**Methods to Check** SIP プリプロセッサ オプションを使って指定されるメソッドが含まれることに注意してください。詳細については、SIP プリプロセッサ オプションの選択（27-53 ページ）を参照してください。

否定を使用する場合は、1 つのメソッドだけを指定できます。次に例を示します。

```
!invite
```

ただし、1 つのルール内の複数の `sip_method` キーワードが **AND** 演算で結合されることに注意してください。たとえば、`invite` と `cancel` を除くすべての抽出されたメソッドを検査するには、次のような 2 つの除外付き `sip_method` キーワードを使用します。

```
sip_method: !invite
sip_method: !cancel
```

シスコでは、`sip_method` キーワードを含むルールに 1 つ以上の `content` キーワードを含めることを推奨しています。こうすると、ルール エンジンが常に高速パターン マッチ機能を使用することで処理速度が上がり、パフォーマンスが向上します。ルールに 1 つ以上の `content` キーワードが含まれている場合は、`content` キーワードの **Use Fast Pattern Matcher** 引数が有効になっているかどうかに関係なく、ルール エンジンが高速パターン マッチ機能を使用することに注意してください。詳細については、コンテンツ一致の検索（36-16 ページ）と高速パターン マッチ機能を使用（Use Fast Pattern Matcher）（36-30 ページ）を参照してください。

sip_stat_code

ライセンス：Protection

各 SIP 応答内の 3 桁のステータス コードは、要求されたアクションの結果を示します。`sip_stat_code` キーワードを使用すると、SIP 応答の中で特定のステータス コードを検査することができます。

1 桁の応答タイプ番号（1 ～ 9）、特定の 3 桁の番号（100 ～ 999）、またはこれらを任意に組み合わせたカンマ区切りリストを指定できます。リスト内のいずれか 1 つの番号が SIP 応答内のコードに一致する場合、そのリストが一致します。

次の表に、指定可能な SIP ステータス コード値の説明を示します。

表 36-40　sip_stat_code 値

検出対象	指定する内容	例	検出結果
1 つの特定のステータス コード	3 桁のステータス コード	189	189
指定された 1 つの数字から始まる 3 桁のコード	1 桁	1	1xx、つまり 100、101、102 など
値のリスト	特定のコードと 1 つの数字を任意に組み合わせてカンマで区切ったもの	222, 3	222 および 300、301、302 など

また、ルールに content キーワードが含まれているかどうかに関係なく、sip_stat_code キーワードを使って指定された値を検索するためにルール エンジンが高速パターン マッチ機能を使用しないことにも注意してください。

GTP キーワード

ライセンス：Protection

3 つの GSRP トンネリング プロトコル（GTP）キーワードを使用すると、GTP バージョン、メッセージ タイプ、および情報要素をコマンド チャネル内で検査できます。content や byte_jump などの他の侵入ルール キーワードと組み合わせて GTP キーワードを使用することはできません。gtp_info または gtp_type キーワードを使用するそれぞれのルールで、gtp_version キーワードを使用する必要があります。

詳細については、次の各項を参照してください。

- gtp_version（36-73 ページ）
- gtp_type（36-74 ページ）
- gtp_info（36-78 ページ）

gtp_version

gtp_version キーワードを使用すると、GTP 制御メッセージの中で GTP バージョン 0、1、または 2 を検査することができます。

定義されているメッセージ タイプと情報要素は GTP バージョンによって異なるため、gtp_type または gtp_info キーワードを使用するときには、このキーワードを使用する必要があります。値として 0、1、または 2 を指定できます。

GTP バージョンを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [gtp_version] を選択して、[オプションの追加（Add Option）] をクリックします。

gtp_version キーワードが表示されます。

手順 2　GTP バージョンを特定するために、0、1、または 2 を指定します。

gtp_type

それぞれの GTP メッセージは、数値と文字列で構成されるメッセージ タイプによって識別されます。gtp_type キーワードを gtp_version キーワードと組み合わせて使用すると、トラフィック内で特定の GTP メッセージ タイプを検査できます。

次の例に示すように、メッセージ タイプとして定義済みの 10 進数値、定義済みの文字列、あるいはどちらか(または両方)を任意に組み合わせたカンマ区切りリストを指定できます。

```
10, 11, echo_request
```

リスト内のそれぞれの値または文字列を照合するとき、システムは OR 演算を使用します。値と文字列を列挙する順序は重要ではありません。リスト内のいずれか 1 つの値または文字列の一致により、キーワードが一致します。認識されない文字列または範囲外の値を含むルールを保存しようとすると、エラーが発生します。

表に示されているように、GTP バージョンに応じて、同じメッセージ タイプの値が異なる場合があることに注意してください。たとえば sgsn_context_request メッセージ タイプの値は GTPv0 と GTPv1 では 50 ですが、GTPv2 では 130 です。

パケット内のバージョン番号に応じて、gtp_type キーワードは異なる値と一致します。上記の例の場合、GTPv0 または GTPv1 パケットではキーワードがメッセージ タイプ値 50 と一致しますが、GTPv2 パケットでは値 130 と一致します。パケット内のメッセージ タイプ値が、パケットで指定されたバージョンの既知の値でない場合は、キーワードがパケットと一致しません。

メッセージ タイプに整数を指定した場合、パケット内で指定されたバージョンとは無関係に、キーワード内のメッセージ タイプが GTP パケット内の値と一致すればキーワードが一致します。

次の表に、GTP メッセージ タイプごとにシステムで認識される定義済みの値と文字列を示します。

表 36-41　GTP メッセージタイプ

値	Version 0	Version 1	Version 2
1	echo_request	echo_request	echo_request
2	echo_response	echo_response	echo_response
3	version_not_supported	version_not_supported	version_not_supported
4	node_alive_request	node_alive_request	該当なし
5	node_alive_response	node_alive_response	該当なし
6	redirection_request	redirection_request	該当なし
7	redirection_response	redirection_response	該当なし
16	create_pdp_context_request	create_pdp_context_request	該当なし
17	create_pdp_context_response	create_pdp_context_response	該当なし
18	update_pdp_context_request	update_pdp_context_request	該当なし
19	update_pdp_context_response	update_pdp_context_response	該当なし
20	delete_pdp_context_request	delete_pdp_context_request	該当なし
21	delete_pdp_context_response	delete_pdp_context_response	該当なし
22	create_aa_pdp_context_request	init_pdp_context_activation_request	該当なし
23	create_aa_pdp_context_response	init_pdp_context_activation_response	該当なし
24	delete_aa_pdp_context_request	該当なし	該当なし
25	delete_aa_pdp_context_response	該当なし	該当なし

表 36-41　GTP メッセージ タイプ(続き)

値	Version 0	Version 1	Version 2
26	error_indication	error_indication	該当なし
27	pdu_notification_request	pdu_notification_request	該当なし
28	pdu_notification_response	pdu_notification_response	該当なし
29	pdu_notification_reject_request	pdu_notification_reject_request	該当なし
30	pdu_notification_reject_response	pdu_notification_reject_response	該当なし
31	該当なし	supported_ext_header_notification	該当なし
32	send_routing_info_request	send_routing_info_request	create_session_request
33	send_routing_info_response	send_routing_info_response	create_session_response
34	failure_report_request	failure_report_request	modify_bearer_request
35	failure_report_response	failure_report_response	modify_bearer_response
36	note_ms_present_request	note_ms_present_request	delete_session_request
37	note_ms_present_response	note_ms_present_response	delete_session_response
38	該当なし	該当なし	change_notification_request
39	該当なし	該当なし	change_notification_response
48	identification_request	identification_request	該当なし
49	identification_response	identification_response	該当なし
50	sgsn_context_request	sgsn_context_request	該当なし
51	sgsn_context_response	sgsn_context_response	該当なし
52	sgsn_context_ack	sgsn_context_ack	該当なし
53	該当なし	forward_relocation_request	該当なし
54	該当なし	forward_relocation_response	該当なし
55	該当なし	forward_relocation_complete	該当なし
56	該当なし	relocation_cancel_request	該当なし
57	該当なし	relocation_cancel_response	該当なし
58	該当なし	forward_srns_contex	該当なし
59	該当なし	forward_relocation_complete_ack	該当なし
60	該当なし	forward_srns_contex_ack	該当なし
64	該当なし	該当なし	modify_bearer_command
65	該当なし	該当なし	modify_bearer_failure_indication
66	該当なし	該当なし	delete_bearer_command
67	該当なし	該当なし	delete_bearer_failure_indication
68	該当なし	該当なし	bearer_resource_command
69	該当なし	該当なし	bearer_resource_failure_indication
70	該当なし	ran_info_relay	downlink_failure_indication
71	該当なし	該当なし	trace_session_activation
72	該当なし	該当なし	trace_session_deactivation

表 36-41　GTP メッセージタイプ(続き)

値	Version 0	Version 1	Version 2
73	該当なし	該当なし	stop_paging_indication
95	該当なし	該当なし	create_bearer_request
96	該当なし	mbms_notification_request	create_bearer_response
97	該当なし	mbms_notification_response	update_bearer_request
98	該当なし	mbms_notification_reject_request	update_bearer_response
99	該当なし	mbms_notification_reject_response	delete_bearer_request
100	該当なし	create_mbms_context_request	delete_bearer_response
101	該当なし	create_mbms_context_response	delete_pdn_request
102	該当なし	update_mbms_context_request	delete_pdn_response
103	該当なし	update_mbms_context_response	該当なし
104	該当なし	delete_mbms_context_request	該当なし
105	該当なし	delete_mbms_context_response	該当なし
112	該当なし	mbms_register_request	該当なし
113	該当なし	mbms_register_response	該当なし
114	該当なし	mbms_deregister_request	該当なし
115	該当なし	mbms_deregister_response	該当なし
116	該当なし	mbms_session_start_request	該当なし
117	該当なし	mbms_session_start_response	該当なし
118	該当なし	mbms_session_stop_request	該当なし
119	該当なし	mbms_session_stop_response	該当なし
120	該当なし	mbms_session_update_request	該当なし
121	該当なし	mbms_session_update_response	該当なし
128	該当なし	ms_info_change_request	identification_request
129	該当なし	ms_info_change_response	identification_response
130	該当なし	該当なし	sgsn_context_request
131	該当なし	該当なし	sgsn_context_response
132	該当なし	該当なし	sgsn_context_ack
133	該当なし	該当なし	forward_relocation_request
134	該当なし	該当なし	forward_relocation_response
135	該当なし	該当なし	forward_relocation_complete
136	該当なし	該当なし	forward_relocation_complete_ack
137	該当なし	該当なし	forward_access
138	該当なし	該当なし	forward_access_ack
139	該当なし	該当なし	relocation_cancel_request
140	該当なし	該当なし	relocation_cancel_response
141	該当なし	該当なし	configuration_transfer_tunnel

表 36-41　GTP メッセージ タイプ(続き)

値	Version 0	Version 1	Version 2
149	該当なし	該当なし	detach
150	該当なし	該当なし	detach_ack
151	該当なし	該当なし	cs_paging
152	該当なし	該当なし	ran_info_relay
153	該当なし	該当なし	alert_mme
154	該当なし	該当なし	alert_mme_ack
155	該当なし	該当なし	ue_activity
156	該当なし	該当なし	ue_activity_ack
160	該当なし	該当なし	create_forward_tunnel_request
161	該当なし	該当なし	create_forward_tunnel_response
162	該当なし	該当なし	suspend
163	該当なし	該当なし	suspend_ack
164	該当なし	該当なし	復帰
165	該当なし	該当なし	resume_ack
166	該当なし	該当なし	create_indirect_forward_tunnel_request
167	該当なし	該当なし	create_indirect_forward_tunnel_response
168	該当なし	該当なし	delete_indirect_forward_tunnel_request
169	該当なし	該当なし	delete_indirect_forward_tunnel_response
170	該当なし	該当なし	release_access_bearer_request
171	該当なし	該当なし	release_access_bearer_response
176	該当なし	該当なし	downlink_data
177	該当なし	該当なし	downlink_data_ack
179	該当なし	該当なし	pgw_restart
180	該当なし	該当なし	pgw_restart_ack
200	該当なし	該当なし	update_pdn_request
201	該当なし	該当なし	update_pdn_response
211	該当なし	該当なし	modify_access_bearer_request
212	該当なし	該当なし	modify_access_bearer_response
231	該当なし	該当なし	mbms_session_start_request
232	該当なし	該当なし	mbms_session_start_response
233	該当なし	該当なし	mbms_session_update_request
234	該当なし	該当なし	mbms_session_update_response
235	該当なし	該当なし	mbms_session_stop_request
236	該当なし	該当なし	mbms_session_stop_response
240	data_record_transfer_request	data_record_transfer_request	該当なし
241	data_record_transfer_response	data_record_transfer_response	該当なし

表 36-41　GTP メッセージ タイプ(続き)

値	Version 0	Version 1	Version 2
254	該当なし	end_marker	該当なし
255	pdu	pdu	該当なし

GTP メッセージ タイプを指定するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ルールの作成(Create Rule)] ページで、ドロップダウン リストから [gtp_type] を選択して、[オプションの追加(Add Option)] をクリックします。

`gtp_type` キーワードが表示されます。

手順 2　メッセージ タイプとして定義済みの 10 進数値(0 ～ 255 の範囲)、定義済み文字列、あるいはそのいずれか(または両方)を任意に組み合わせたカンマ区切りリストを指定します。システムで認識される値と文字列については、GTP メッセージ タイプの表を参照してください。

gtp_info

1 つの GTP メッセージには多数の情報要素が含まれることがあり、それぞれの要素は定義済み数値および定義済み文字列によって識別されます。`gtp_info` キーワードを `gtp_version` キーワードと組み合わせて使用すると、指定された情報要素の先頭から検査を開始し、検査対象を指定の情報要素に限定することができます。

情報要素に対して定義された 10 進数値と定義された文字列のどちらでも指定できます。単一の値または文字列を指定することも、1 つのルール内で複数の `gtp_info` キーワードを使って複数の情報要素を検査することもできます。

1 つのメッセージに同じタイプの複数の情報要素が含まれている場合は、すべてが照合対象として検査されます。情報要素が無効な順序で出現する場合は、最後のインスタンスだけが検査されます。

GTP バージョンに応じて、同じ情報要素の値が異なる場合があることに注意してください。たとえば `cause` 情報要素の値は GTPv0 と GTPv1 では 1 ですが、GTPv2 では 2 です。

パケット内のバージョン番号に応じて、`gtp_info` キーワードは異なる値と一致します。上記の例の場合、GTPv0 または GTPv1 パケットではキーワードが情報要素値 1 と一致しますが、GTPv2 パケットでは値 2 と一致します。パケット内の情報要素値が、パケットで指定されたバージョンの既知の値でない場合は、キーワードがパケットと一致しません。

情報要素に整数を指定した場合、パケット内で指定されたバージョンとは無関係に、キーワード内のメッセージ タイプが GTP パケット内の値と一致すればキーワードが一致します。

次の表に、GTP 情報要素ごとにシステムで認識される値と文字列を示します。

表 36-42　GTP 情報要素

値	Version 0	Version 1	Version 2
1	cause	cause	imsi
2	imsi	imsi	cause
3	rai	rai	recovery
4	tlli	tlli	該当なし
5	p_tmsi	p_tmsi	該当なし
6	qos	該当なし	該当なし
8	recording_required	recording_required	該当なし
9	認証	認証	該当なし
11	map_cause	map_cause	該当なし
12	p_tmsi_sig	p_tmsi_sig	該当なし
13	ms_validated	ms_validated	該当なし
14	recovery	recovery	該当なし
15	selection_mode	selection_mode	該当なし
16	flow_label_data_1	teid_1	該当なし
17	flow_label_signalling	teid_control	該当なし
18	flow_label_data_2	teid_2	該当なし
19	ms_unreachable	teardown_ind	該当なし
20	該当なし	nsapi	該当なし
21	該当なし	ranap	該当なし
22	該当なし	rab_context	該当なし
23	該当なし	radio_priority_sms	該当なし
24	該当なし	radio_priority	該当なし
25	該当なし	packet_flow_id	該当なし
26	該当なし	charging_char	該当なし
27	該当なし	trace_ref	該当なし
36	該当なし	trace_type	該当なし
29	該当なし	ms_unreachable	該当なし
71	該当なし	該当なし	apn
72	該当なし	該当なし	ambr
73	該当なし	該当なし	ebi
74	該当なし	該当なし	ip_addr
75	該当なし	該当なし	mei
76	該当なし	該当なし	msisdn
77	該当なし	該当なし	indication
78	該当なし	該当なし	pco
79	該当なし	該当なし	paa

表 36-42　GTP 情報要素（続き）

値	Version 0	Version 1	Version 2
80	該当なし	該当なし	bearer_qos
80	該当なし	該当なし	flow_qos
82	該当なし	該当なし	rat_type
83	該当なし	該当なし	serving_network
84	該当なし	該当なし	bearer_tft
85	該当なし	該当なし	tad
86	該当なし	該当なし	uli
87	該当なし	該当なし	f_teid
88	該当なし	該当なし	tmsi
89	該当なし	該当なし	cn_id
90	該当なし	該当なし	s103pdf
91	該当なし	該当なし	s1udf
92	該当なし	該当なし	delay_value
93	該当なし	該当なし	bearer_context
94	該当なし	該当なし	charging_id
95	該当なし	該当なし	charging_char
96	該当なし	該当なし	trace_info
97	該当なし	該当なし	bearer_flag
99	該当なし	該当なし	pdn_type
100	該当なし	該当なし	pti
101	該当なし	該当なし	drx_parameter
103	該当なし	該当なし	gsm_key_tri
104	該当なし	該当なし	umts_key_cipher_quin
105	該当なし	該当なし	gsm_key_cipher_quin
106	該当なし	該当なし	umts_key_quin
107	該当なし	該当なし	eps_quad
108	該当なし	該当なし	umts_key_quad_quin
109	該当なし	該当なし	pdn_connection
110	該当なし	該当なし	pdn_number
111	該当なし	該当なし	p_tmsi
112	該当なし	該当なし	p_tmsi_sig
113	該当なし	該当なし	hop_counter
114	該当なし	該当なし	ue_time_zone
115	該当なし	該当なし	trace_ref
116	該当なし	該当なし	complete_request_msg
117	該当なし	該当なし	guti

表 36-42　GTP 情報要素（続き）

値	Version 0	Version 1	Version 2
118	該当なし	該当なし	f_container
119	該当なし	該当なし	f_cause
120	該当なし	該当なし	plmn_id
121	該当なし	該当なし	target_id
123	該当なし	該当なし	packet_flow_id
124	該当なし	該当なし	rab_contex
125	該当なし	該当なし	src_rnc_pdcp
126	該当なし	該当なし	udp_src_port
127	charge_id	charge_id	apn_restriction
128	end_user_address	end_user_address	selection_mode
129	mm_context	mm_context	src_id
130	pdp_context	pdp_context	該当なし
131	apn	apn	change_report_action
132	protocol_config	protocol_config	fq_csid
133	gsn	gsn	channel
134	msisdn	msisdn	emlpp_pri
135	該当なし	qos	node_type
136	該当なし	authentication_qu	fqdn
137	該当なし	tft	ti
138	該当なし	target_id	mbms_session_duration
139	該当なし	utran_trans	mbms_service_area
140	該当なし	rab_setup	mbms_session_id
141	該当なし	ext_header	mbms_flow_id
142	該当なし	trigger_id	mbms_ip_multicast
143	該当なし	omc_id	mbms_distribution_ack
144	該当なし	ran_trans	rfsp_index
145	該当なし	pdp_context_pri	uci
146	該当なし	addi_rab_setup	csg_info
147	該当なし	sgsn_number	csg_id
148	該当なし	common_flag	cmi
149	該当なし	apn_restriction	service_indicator
150	該当なし	radio_priority_lcs	detach_type
151	該当なし	rat_type	ldn
152	該当なし	user_loc_info	node_feature
153	該当なし	ms_time_zone	mbms_time_to_transfer
154	該当なし	imei_sv	throttling

表 36-42　GTP 情報要素（続き）

値	Version 0	Version 1	Version 2
155	該当なし	camel	arp
156	該当なし	mbms_ue_context	epc_timer
157	該当なし	tmp_mobile_group_id	signalling_priority_indication
158	該当なし	rim_routing_addr	tmgi
159	該当なし	mbms_config	mm_srvcc
160	該当なし	mbms_service_area	flags_srvcc
161	該当なし	src_rnc_pdcp	nmbr
162	該当なし	addi_trace_info	該当なし
163	該当なし	hop_counter	該当なし
164	該当なし	plmn_id	該当なし
165	該当なし	mbms_session_id	該当なし
166	該当なし	mbms_2g3g_indicator	該当なし
167	該当なし	enhanced_nsapi	該当なし
168	該当なし	mbms_session_duration	該当なし
169	該当なし	addi_mbms_trace_info	該当なし
170	該当なし	mbms_session_repetition_num	該当なし
171	該当なし	mbms_time_to_data	該当なし
173	該当なし	bss	該当なし
174	該当なし	cell_id	該当なし
175	該当なし	pdu_num	該当なし
177	該当なし	mbms_bearer_capab	該当なし
178	該当なし	rim_routing_disc	該当なし
179	該当なし	list_pfc	該当なし
180	該当なし	ps_xid	該当なし
181	該当なし	ms_info_change_report	該当なし
182	該当なし	direct_tunnel_flags	該当なし
183	該当なし	correlation_id	該当なし
184	該当なし	bearer_control_mode	該当なし
185	該当なし	mbms_flow_id	該当なし
186	該当なし	mbms_ip_multicast	該当なし
187	該当なし	mbms_distribution_ack	該当なし
188	該当なし	reliable_inter_rat_handover	該当なし
189	該当なし	rfsp_index	該当なし
190	該当なし	fqdn	該当なし
191	該当なし	evolved_allocation1	該当なし
192	該当なし	evolved_allocation2	該当なし

表 36-42　GTP 情報要素（続き）

値	Version 0	Version 1	Version 2
193	該当なし	extended_flags	該当なし
194	該当なし	uci	該当なし
195	該当なし	csg_info	該当なし
196	該当なし	csg_id	該当なし
197	該当なし	cmi	該当なし
198	該当なし	apn_ambr	該当なし
199	該当なし	ue_network	該当なし
200	該当なし	ue_ambr	該当なし
201	該当なし	apn_ambr_nsapi	該当なし
202	該当なし	ggsn_backoff_timer	該当なし
203	該当なし	signalling_priority_indication	該当なし
204	該当なし	signalling_priority_indication_nsapi	該当なし
205	該当なし	high_bitrate	該当なし
206	該当なし	max_mbr	該当なし
251	charging_gateway_addr	charging_gateway_addr	該当なし
255	private_extension	private_extension	private_extension

次の手順に従って、GTP 情報要素を指定できます。

GTP 情報要素を指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [gtp_info] を選択して、[オプションの追加（Add Option）] をクリックします。

`gtp_info` キーワードが表示されます。

手順 2　情報要素に関する 1 つの定義済み 10 進数値（0 ～ 255）または 1 つの定義済み文字列を指定します。システムで認識される値と文字列については、GTP 情報要素の表を参照してください。

Modbus キーワード

ライセンス：Protection

Modbus キーワードを使用すると、Modbus 要求または応答内の [データ（Data）] フィールドの先頭を指し示したり、Modbus 機能コードと照合したり、Modbus ユニット ID と照合することができます。Modbus キーワードを単独で使用することも、`content` や `byte_jump` など他のキーワードと組み合わせて使用することもできます。

詳細については、次の各項を参照してください。

- modbus_data（36-84 ページ）
- modbus_func（36-84 ページ）
- modbus_unit（36-85 ページ）

modbus_data

modbus_data キーワードを使用すると、Modbus 要求または応答内の [データ（Data）] フィールドの先頭を指し示すことができます。

[Modbus データ（Modbus Data）] フィールドの先頭を指し示すには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [modbus_data] を選択して、[オプションの追加（Add Option）] をクリックします。

modbus_data キーワードが表示されます。

modbus_data キーワードには引数がありません。

modbus_func

modbus_func キーワードを使用すると、Modbus アプリケーション層要求または応答ヘッダー内の Function Code（機能コード）フィールドを照合できます。Modbus 機能コードとして、1 つの定義済み 10 進数値または 1 つの定義済み文字列を指定できます。

次の表に、Modbus 機能コードとしてシステムで認識される定義済みの値と文字列を示します。

表 36-43　Modbus 機能コード

値	文字列
1	read_coils
2	read_discrete_inputs
3	read_holding_registers
4	read_input_registers
5	write_single_coil
6	write_single_register
7	read_exception_status
8	diagnostics
11	get_comm_event_counter
12	get_comm_event_log
15	write_multiple_coils
16	write_multiple_registers
17	report_slave_id
20	read_file_record

表 36-43　Modbus 機能コード(続き)

値	文字列
21	write_file_record
22	mask_write_register
23	read_write_multiple_registers
24	read_fifo_queue
43	encapsulated_interface_transport

Modbus 機能コードを指定するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ルールの作成(Create Rule)] ページで、ドロップダウン リストから [modbus_func] を選択して、[オプションの追加(Add Option)] をクリックします。

`modbus_func` キーワードが表示されます。

手順 2　機能コード用の 1 つの定義済み 10 進数値(0 ～ 255)または 1 つの定義済み文字列を指定します。システムで認識される値と文字列については、Modbus 機能コードの表を参照してください。

modbus_unit

`modbus_unit` キーワードを使用すると、Modbus 要求または応答ヘッダー内の [ユニット ID(Unit ID)] フィールドで 1 つの 10 進数値を照合できます。

Modbus ユニット ID を指定するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ルールの作成(Create Rule)] ページで、ドロップダウン リストから [modbus_unit] を選択して、[オプションの追加(Add Option)] をクリックします。

`modbus_unit` キーワードが表示されます。

手順 2　10 進数値(0 ～ 255 の範囲)を 1 つ指定します。

DNP3 キーワード

ライセンス: Protection

DNP3 キーワードを使用すると、アプリケーション層フラグメントの先頭を指し示したり、DNP3 要求および応答での DNP3 機能コードやオブジェクトを照合したり、DNP3 応答での内部通知フラグを照合することができます。DNP3 キーワードを単独で使用することも、`content` や `byte_jump` など他のキーワードと組み合わせて使用することもできます。

詳細については、次の各項を参照してください。

- dnp3_data（36-86 ページ）
- dnp3_func（36-86 ページ）
- dnp3_ind（36-88 ページ）
- dnp3_obj（36-88 ページ）

dnp3_data

dnp3_data キーワードを使用すると、再構築された DNP3 アプリケーション層フラグメントの先頭を指し示すことができます。

DNP3 プリプロセッサは、リンク層フレームをアプリケーション層フラグメントに再構築します。dnp3_data キーワードは、各アプリケーション層フラグメントの先頭を指し示します。他のルール オプションは、16 バイトごとにデータを分離してチェックサムを追加せずに、フラグメント内の再構築されたデータを照合することができます。

再構築された DNP3 フラグメントの先頭を指すには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [modbus_data] を選択して、[オプションの追加（Add Option）] をクリックします。

dnp3_data キーワードが表示されます。

dnp3_data キーワードには引数がありません。

dnp3_func

dnp3_func キーワードを使用すると、DNP3 アプリケーション層要求または応答ヘッダー内の Function Code（機能コード）フィールドを照合できます。DNP3 機能コードとして、1 つの定義済み 10 進数値または 1 つの定義済み文字列を指定できます。

次の表に、DNP3 機能コードとしてシステムで認識される定義済みの値と文字列を示します。

表 36-44　DNP3 機能コード

値	文字列
0	confirm
1	read
2	write
3	選択
4	operate
5	direct_operate
6	direct_operate_nr
7	immed_freeze
8	immed_freeze_nr
9	freeze_clear

表 36-44　DNP3 機能コード（続き）

値	文字列
10	freeze_clear_nr
11	freeze_at_time
12	freeze_at_time_nr
13	cold_restart
14	warm_restart
15	initialize_data
16	initialize_appl
17	start_appl
18	stop_appl
19	save_config
20	enable_unsolicited
21	disable_unsolicited
22	assign_class
23	delay_measure
24	record_current_time
25	open_file
26	close_file
27	delete_file
28	get_file_info
29	authenticate_file
30	abort_file
31	activate_config
32	authenticate_req
33	authenticate_err
129	response
130	unsolicited_response
131	authenticate_resp

DNP3 機能コードを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [dnp3_func] を選択して、[オプションの追加（Add Option）] をクリックします。

`dnp3_func` キーワードが表示されます。

手順 2　機能コード用の 1 つの定義済み 10 進数値（0 ～ 255）または 1 つの定義済み文字列を指定します。システムで認識される値と文字列については、DNP3 機能コードの表を参照してください。

dnp3_ind

dnp3_ind キーワードを使用すると、DNP3 アプリケーション層応答ヘッダー内の [内部通知（Internal Indications）] フィールド内のフラグを照合できます。

1 つの既知のフラグ、または次の例のように、カンマで区切ったフラグのリストを指定できます。

```
class_1_events, class_2_events
```

複数のフラグを指定した場合、キーワードはリスト内の任意のフラグと一致します。いくつかのフラグの組み合わせを検出するには、1 つのルール内で dnp3_ind キーワードを複数回使用します。

定義済みの DNP3 内部通知フラグとしてシステムによって認識される文字列構文を以下に示します。

```
class_1_events
class_2_events
class_3_events
need_time
local_control
device_trouble
device_restart
no_func_code_support
object_unknown
parameter_error
event_buffer_overflow
already_executing
config_corrupt
reserved_2
reserved_1
```

DNP3 内部通知フラグを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [dnp3_ind] を選択して、[オプションの追加（Add Option）] をクリックします。

dnp3_ind キーワードが表示されます。

手順 2　1 つの既知のフラグまたはカンマ区切ったフラグのリストを指定できます。

dnp3_obj

dnp3_obj キーワードを使用すると、要求または応答内の DNP3 オブジェクト ヘッダーを照合できます。

DNP3 データは、アナログ入力やバイナリ入力など、さまざまなタイプの一連の DNP3 オブジェクトで構成されます。各タイプは、それぞれ 10 進数値で識別される *グループ* を使って区別されます（アナログ入力グループ、バイナリ入力グループなど）。各グループ内のオブジェクトは、それぞれオブジェクト データ形式を特定する *オブジェクト バリエーション*（16 ビット整数、32 ビット整数、短精度浮動小数点など）によってさらに識別されます。また、オブジェクト バリエーションの各タイプは 10 進数値でも識別可能です。

オブジェクト ヘッダーを識別する際には、オブジェクト ヘッダー グループのタイプを示す 10 進数値とオブジェクト バリエーションのタイプを示す 10 進数値を指定します。この 2 つの組み合わせによって DNP3 オブジェクトの特定のタイプが定義されます。

DNP3 オブジェクトを指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [dnp3_obj] を選択して、[オプションの追加（Add Option）] をクリックします。

`dnp3_obj` キーワードが表示されます。

手順 2　既知のオブジェクト グループを識別するために 1 つの 10 進数値（0 ～ 255）を指定し、既知のオブジェクト バリエーション タイプを識別するために別の 10 進数値（0 ～ 255）を指定します。

CIP および ENIP のキーワード

ライセンス：Protection

次のキーワードを単体でまたは組み合わせて使用すると、CIP プリプロセッサで検出された CIP および ENIP トラフィックに対する攻撃を識別するカスタム侵入ルールを作成できます。設定可能なキーワードについては、許容範囲内の単一の整数を指定します。詳細については、CIP プリプロセッサの設定（28-5 ページ）を参照してください。

表 36-45　CIP および ENIP のキーワード

キーワード	範囲
`cip_attribute`	0 ～ 65535
`cip_class`	0 ～ 65535
`cip_conn_path_class`	0 ～ 65535
`cip_instance`	0 ～ 4284927295
`cip_req`	該当なし
`cip_rsp`	該当なし
`cip_service`	0 ～ 127
`cip_status`	0 ～ 255
`enip_command`	0 ～ 65535
`enip_req`	該当なし
`enip_rsp`	該当なし

パケット特性の検査

ライセンス：Protection

特定のパケット特性を持つパケットに対してのみイベントを生成するルールを作成できます。FireSIGHT システムには、パケット特性を評価するための次のキーワードが備わっています。

- dsize（36-90 ページ）
- isdataat（36-90 ページ）
- sameip（36-91 ページ）
- fragoffset（36-91 ページ）

- cvs（36-92 ページ）

dsize

ライセンス：Protection

dsize キーワードはパケット ペイロード サイズを検査します。「大なり」演算子と「小なり」演算子（<、>）を使って値の範囲を指定することができます。次の構文をに従って範囲を指定できます。

```
>number_of_bytes
<number_of_bytes
number_of_bytes<>number_of_bytes
```

たとえば、400 バイトを超えるパケット サイズを指定するには、dtype 値として >400 を使用します。500 バイト未満のパケット サイズを指定するには、<500 を使用します。400 ～ 500 バイトのパケットに対してルールをトリガーとして使用するよう指定するには、400<>500 を使用します。

注意　dsize キーワードは、プリプロセッサによってデコードされる前のパケットを検査します。

isdataat

ライセンス：Protection

isdataat キーワードは、ペイロード内の特定の位置にデータが存在することを確認するよう、ルール エンジンに指示します。

次の表に、isdataat キーワードで使用可能な引数を列挙します。

表 36-46　isdataat の引数

引数	タイプ（Type）	説明
Offset	必須（Required）	ペイロード内の特定の位置。たとえば、パケット ペイロード内のバイト位置 50 にデータが出現することを検査するには、オフセット値として 50 を指定します。! 修飾子は isdataat 検査の結果を否定します。特定量のデータがペイロードに存在しない場合は警告が出されます。 また、既存の byte_extract 変数を使用してこの引数の値を指定することもできます。詳細については、パケット データをキーワード引数の中に読み込む（36-92 ページ）を参照してください。
Relative	オプション	最後に見つかったコンテンツ一致を基準にして相対的な位置を計算します。相対位置を指定する場合は、カウンタがバイト 0 から始まることに注意してください。最後に見つかったコンテンツ一致から順方向に移動するバイト数から 1 を差し引いて位置を計算します。たとえば、最後に見つかったコンテンツ一致から 9 バイト後にデータが出現すべきことを指定するには、相対オフセットとして 8 を指定します。
Raw Data	オプション	FireSIGHT システム プリプロセッサによるデコードやアプリケーション層正規化が行われる前の、元のパケット ペイロードにデータが配置されていることを指定します。前のコンテンツ一致が未加工パケット データ内に存在していた場合は、この引数を **Relative** と一緒に使用できます。

たとえば、foo というコンテンツを検索するルールで isdataat の値が次のように指定される場合、

- Offset = !10
- Relative = enabled

ルール エンジンが foo の後ろからペイロード末尾までに 10 バイトを検出しない場合、システムは警告を出します。

isdataat を使用するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [isdataat] を選択して、[オプションの追加（Add Option）] をクリックします。

[isdataat] セクションが表示されます。

sameip

ライセンス：Protection

sameip キーワードは、パケットの送信元 IP アドレスと宛先 IP アドレスが同じであることを検査します。このキーワードは引数を受け入れません。

fragoffset

ライセンス：Protection

fragoffset キーワードは、フラグメント化されたパケットのオフセットを検査します。一部のエクスプロイト（WinNuke サービス拒否攻撃など）では、特定のオフセットを持つ手動生成されたパケット フラグメントが使われるため、このキーワードが役立ちます。

たとえば、フラグメント化されたパケットのオフセットが 31337 バイトかどうかを検査するには、fragoffset 値として 31337 を指定します。

fragoffset キーワードの引数を指定するときには、次の演算子を使用できます。

表 36-47　fragoffset キーワードの引数演算子

演算子	説明
!	ノット
>	より大きい
<	より少ない

否定（!）演算子を < や > と組み合わせて使用できないことに注意してください。

cvs

ライセンス：Protection

cvs キーワードは、Concurrent Versions System（CVS）トラフィック内で不正な形式の CVS エントリを検査します。攻撃者は不正な形式のエントリを使用して、ヒープ オーバーフローを強制的に発生させ、CVS サーバ上で有害コードを実行することができます。このキーワードを使用すると、2 つの既知の CVS 脆弱性 CVE-2004-0396（CVS 1.11.x ～ 1.11.15 と 1.12.x ～ 1.12.7）および CVS-2004-0414（CVS 1.12.x ～ 1.12.8 と 1.11.x ～ 1.11.16）に対する攻撃を識別できます。cvs キーワードは、正しい形式のエントリであることを検査して、不正な形式のエントリが検出された場合はアラートを生成します。

CVS が動作するポートをルールに含める必要があります。さらに、トラフィックが発生する可能性のあるポートを TCP ポリシー内のストリーム再構築用のポート リストに追加することで、CVS セッションの状態を保持できるようにする必要があります。ストリーム再構築が行われるクライアント ポートのリストには、TCP ポート 2401（pserver）と 514（rsh）が含まれています。ただし、サーバが xinetd サーバ（つまり pserver）として動作する場合は、任意の TCP ポート上で動作できることに注意してください。すべての非標準ポートを、ストリーム再構築の [クライアント ポート（Client Ports）] リストに追加します。詳細については、ストリーム再構成のオプションの選択（29-30 ページ）を参照してください。

不正な形式の CVS エントリを検出するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　cvs オプションをルールに追加し、キーワード引数として「invalid-entry」と入力します。

パケット データをキーワード引数の中に読み込む

ライセンス：Protection

byte_extract キーワードを使用すると、指定したバイト数をパケットから変数の中に読み込むことができます。後で、その変数を、同じルール内で他の検出キーワードの特定の引数の値として使用できます。

たとえば、パケット データに含まれるバイト数が特定のバイト セグメントで記述されている場合、パケットからデータ サイズを抽出するには、これが役立ちます。たとえば、特定のバイト セグメントにおいて、後続データが 4 バイト構成であると記述されている場合、データ サイズ 4 バイトを抽出して変数値として使用できます。

byte_extract を使用するとき、1 つのルール内で最大 2 つの異なる変数を同時に作成できます。byte_extract 変数を何回でも再定義できます。同じ変数名と別の変数定義を使って新しい byte_extract キーワードを入力した場合、その前の変数定義がオーバーライドされます。

次の表で、byte_extract キーワードで必要な引数について説明します。

表 36-48　byte_extract の必須引数

引数	説明
Bytes to Extract	パケットから抽出するバイト数。1、2、3、または 4 バイトを指定できます。
Offset	ペイロード内でデータの抽出を開始するバイト数。-65534 ～ 65535 バイトを指定できます。オフセット カウンタはバイト 0 から始まるため、順方向に数えるバイト数から 1 を差し引いてオフセット値を計算してください。たとえば、順方向に 8 バイト数えるには 7 を指定します。ルール エンジンは、パケット ペイロードの先頭から（**Relative** も一緒に指定した場合は最後に見つかったコンテンツ一致の後から）順方向に数えます。なお、負の数値を指定できるのは、**Relative** を一緒に指定した場合だけです。詳細については、byte_extract の追加のオプション引数の表を参照してください。
Variable Name	他の検出キーワードの引数で使用する変数名。英数字の文字列を指定できます（ただし文字で始まる必要があります）。

抽出対象のデータを見つける方法をさらに詳しく定義するには、次の表に示す引数を使用できます。

表 36-49　byte_extract の追加のオプション引数

引数	説明
Multiplier	パケットから抽出された値の乗数。0 ～ 65535 を指定できます。乗数を指定しない場合のデフォルト値は 1 です。
Align	抽出された値を最も近い 2 バイトまたは 4 バイト境界に切り上げます。**Multiplier** も一緒に選択した場合、システムはこの調整の前に乗数を適用します。
Relative	ペイロードの先頭ではなく、最後に見つかったコンテンツ一致の末尾を基準にして **Offset** を計算します。詳細については、byte_extract の必須引数の表を参照してください。

DCE/RPC、**Endian**、または **Number Type** のうち 1 つだけを指定できます。

検査対象となるバイトを `byte_extract` キーワードでどのように計算するか定義するには、次の表の中から引数を選択できます。どの引数も選択しない場合、ルール エンジンはビッグ エンディアン バイト順を使用します。

表 36-50　byte_extract のエンディアンネス引数

引数	説明
Big Endian	デフォルトのネットワーク バイト順であるビッグ エンディアン バイト順でデータを処理します。

表 36-50　byte_extract のエンディアンネス引数（続き）

引数	説明
Little Endian	リトル エンディアン バイト順でデータを処理します。
DCE/RPC	DCE/RPC プリプロセッサで処理されるトラフィック用に byte_extract キーワードを指定します。詳細については、DCE/RPC トラフィックのデコード（27-2 ページ）を参照してください。 DCE/RPC プリプロセッサがビッグ エンディアンまたはリトル エンディアン バイト順を決定します。**Number Type** 引数と **Endian** 引数は適用されません。 この引数を有効にした場合は、他の特定の DCE/RPC キーワードと組み合わせて byte_extract を使用することもできます。詳細については、DCE/RPC キーワード（36-67 ページ）を参照してください。

データを読み取るときの数値タイプを ASCII 文字列として指定できます。パケット内のストリング データをシステムがどのように認識するかを定義するには、次の表のいずれかの引数を選択できます。

表 36-51　byte_extract の Number Type 引数

引数	説明
Hexadecimal String	抽出されたストリング データを 16 進形式で読み取ります。
Decimal String	抽出されたストリング データを 10 進形式で読み取ります。
Octal String	抽出されたストリング データを 8 進形式で読み取ります。

たとえば、byte_extract の値を次のように指定した場合、

- Bytes to Extract = 4
- Variable Name = var
- Offset = 8
- Relative = enabled

ルール エンジンは、最後に見つかったコンテンツ一致から（それを基準にして）9 バイト後に出現する、4 バイトで表現される数値を var という名前の変数の中に読み込みます。後でこの変数を、特定のキーワード引数の値としてルール内で指定できます。

byte_extract キーワードで定義した変数を指定できるキーワード引数を、次の表に列挙します。

表 36-52　byte_extract 変数を使用できる引数

キーワード	引数	詳細
content	Depth、Offset、Distance、Within	コンテンツ一致の制約（36-20 ページ）
byte_jump	Offset	byte_jump（36-35 ページ）
byte_test	Offset、Value	byte_test（36-37 ページ）
isdataat	Offset	isdataat（36-90 ページ）

byte_extract を使用するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [byte_extract] を選択して、[オプションの追加（Add Option）] をクリックします。

[byte_extract] セクションが、選択した最後のキーワードの下に表示されます。

ルール キーワードを使用したアクティブ応答の開始

ライセンス：Protection

システムは、トリガーとして使用された TCP ルールに応答して TCP 接続を閉じるために、またはトリガーとして使用された UDP ルールに応答して UDP セッションを閉じるために、アクティブ応答を開始できます。2 つのキーワードにより、別々の方法でアクティブ応答を開始できます。どちらかのキーワードを含むルールをパケットがトリガーとして使用すると、システムは 1 つのアクティブ応答を開始します。`config response` コマンドを使用して、使用するアクティブ応答インターフェイス、およびパッシブ展開で試行する TCP リセットの回数を設定することもできます。

リセットは接続やセッションに影響を与えるのに間に合うまでに到着する可能性が高いため、アクティブ応答はインライン展開で最も効果を発揮します。たとえば、インライン展開での `react` キーワードに応答して、システムは接続の両端用のトラフィックに TCP リセット（RST）パケットを直接挿入し、通常はこれによって接続が閉じます。

（パッシブ展開ではシステムがパケットを挿入できない、攻撃者がアクティブ応答を無視または回避するよう選択する可能性があるなど）さまざまな理由で、アクティブ応答はファイアウォールの代わりとして想定されていません。

アクティブ応答は戻って来ることがあるため、システムは TCP リセットによる TCP リセットの開始を許可しません。これにより、アクティブ応答が無限に続くことを防止できます。また、システムは、標準的な慣行に従って ICMP 到達不能パケットによる ICMP 到達不能パケットの開始を許可しません。

侵入ルールがアクティブ応答をトリガーとして使用した後、接続またはセッションで追加のトラフィックを検出するよう、TCP ストリーム プリプロセッサを設定できます。追加のトラフィックが検出されると、プリプロセッサは、指定された最大値まで、追加のアクティブ応答を接続またはセッションの両端に送信します。詳細については、侵入廃棄ルールでのアクティブ応答の開始（29-3 ページ）を参照してください。

アクティブ応答を開始するために使用できるキーワードに固有の情報については、以下の項を参照してください。

- タイプ別、方向別のアクティブ応答の開始（36-96 ページ）
- TCP リセット前の HTML ページの送信（36-98 ページ）
- アクティブ応答のリセット試行とインターフェイスの設定（36-98 ページ）

タイプ別、方向別のアクティブ応答の開始

ライセンス：Protection

`resp` キーワードを使用すると、ルール ヘッダーで TCP プロトコルと UDP プロトコルのどちらが指定されているかに基づいて、TCP 接続または UDP セッションにアクティブに（能動的に）応答できます。詳細については、プロトコルの指定（36-5 ページ）を参照してください。

キーワード引数を使用すると、パケットの方向、および TCP リセット（RST）パケットと ICMP 到達不能パケットのどちらをアクティブ応答として使用するかを指定できます。

任意の TCP リセット引数または ICMP 到達不能引数を使用して、TCP 接続を閉じることができます。UDP セッションを閉じるには、ICMP 到達不能引数だけを使用する必要があります。

また、さまざまな TCP リセット引数を使用することで、パケットの送信元、宛先、またはその両方にアクティブ応答を送ることができます。すべての ICMP 到達不能引数はパケット送信元に送られます。ICMP ネットワーク、ホスト、またはポートのどの到達不能パケットを使用するか（または 3 つすべてを使用するか）を指定できます。

ルールがトリガーとして使用されたときに FireSIGHT システムで実行されるアクションを正確に指定するために、resp キーワードで使用できる引数を次の表に列挙します。

表 36-53　resp の引数

引数	説明
reset_source	ルールをトリガーとして使用したパケットを送信したエンドポイントに TCP リセット パケットを送ります。この代わりに、下位互換性のためにサポートされている rst_snd を指定することもできます。
reset_dest	ルールをトリガーとして使用したパケットの宛先であるエンドポイントに TCP リセット パケットを送ります。この代わりに、下位互換性のためにサポートされている rst_rcv を指定することもできます。
reset_both	送信側エンドポイントと受信側エンドポイントの両方に TCP リセット パケットを送ります。この代わりに、下位互換性のためにサポートされている rst_all を指定することもできます。
icmp_net	送信側に ICMP ネットワーク到達不能メッセージを送ります。
icmp_host	送信側に ICMP ホスト到達不能メッセージを送ります。
icmp_port	送信側に ICMP ポート到達不能メッセージを送ります。この引数は、UDP トラフィックを終了するために使われます。
icmp_all	送信側に次の ICMP メッセージを転送します。 • ネットワーク到達不能 • ホスト到達不能 • ポート到達不能

たとえば、ルールがトリガーとして使用されたときに接続の両側をリセットするようルールを設定するには、resp キーワードの値として reset_both を使用します。

次のように、カンマ区切りリストを使用して複数の引数を指定できます。

argument,*argument*,*argument*

使用するアクティブ応答インターフェイスおよびパッシブ展開での TCP リセット試行回数を設定するために config response コマンドを使用する方法については、アクティブ応答のリセット試行とインターフェイスの設定（36-98 ページ）を参照してください。

アクティブ応答を指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [resp] を選択して、[オプションの追加（Add option）] をクリックします。

resp キーワードが表示されます。

手順 2　[resp] フィールドで、resp の引数の表にある引数を指定します。複数の引数を指定する場合は、カンマ区切りのリストを使用します。

TCP リセット前の HTML ページの送信

ライセンス：Protection

react キーワードを使用すると、パケットがルールをトリガーとして使用した時点でデフォルト HTML ページを TCP 接続クライアントに送信できます。HTML ページの送信後に、システムは TCP リセット パケットを使って接続の両端へのアクティブ応答を開始します。react キーワードは UDP トラフィックのアクティブ応答をトリガーとして使用しません。

オプションで、次の引数を指定できます。

msg

msg 引数を使用する react ルールがパケットによってトリガーとして使用されると、HTML ページにルール イベント メッセージが表示されます。イベント メッセージのフィールドについては、ルール構造について（36-2 ページ）を参照してください。

msg 引数を指定しない場合、HTML ページには次のメッセージが含まれます。

```
You are attempting to access a forbidden site.
Consult your system administrator for details.
```

（注）　アクティブ応答は戻って来ることがあるため、HTML 応答ページによって react ルールがトリガーとして使用されないようにしてください（結果としてアクティブ応答が無限に続く可能性があります）。シスコ では、react ルールを十分にテストしてから実稼動環境でアクティブにするよう推奨しています。

使用するアクティブ応答インターフェイスおよびパッシブ展開での TCP リセット試行回数を設定するために config response コマンドを使用する方法については、アクティブ応答のリセット試行とインターフェイスの設定（36-98 ページ）を参照してください。

アクティブ応答を開始する前に HTML ページを送信するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [react] を選択して、[オプションの追加（Add option）] をクリックします。

react キーワードが表示されます。

手順 2　次の 2 つの選択肢があります。

- 接続を閉じる前に、ルール用に設定されたイベント メッセージを含む HTML ページをクライアントに送信するには、[react] フィールドに「msg」と入力します。
- 接続を閉じる前に、次のデフォルト メッセージを含む HTML ページをクライアントに送信するには、[react] フィールドを空白のままにします。

```
You are attempting to access a forbidden site.
Consult your system administrator for details
```

アクティブ応答のリセット試行とインターフェイスの設定

ライセンス：Protection

config response コマンドを使用すると、resp ルールと react ルールによって開始される TCP リセットの動作を詳細に設定できます。また、このコマンドは、廃棄ルールによって開始されるアクティブ応答の動作にも影響を与えます（詳細については、侵入廃棄ルールでのアクティブ応答の開始（29-3 ページ）を参照してください）。

config response コマンドを使用するには、高度な USER_CONF 変数内の別個の 1 行にこれを挿入します。USER_CONF 変数の使用方法については、拡張変数について（3-37 ページ）を参照してください。

注意　機能の説明またはサポート担当の指示に従う場合を除き、侵入ポリシー機能を設定するために高度な USER_CONF 変数を**使用しない**でください。競合または重複する設定が存在すると、システムが停止します。

アクティブ応答リセット試行、アクティブ応答インターフェイス、またはその両方を指定するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　アクティブ応答の回数のみを指定するのか、アクティブ応答インターフェイスのみを指定するのか、またはその両方を指定するのかに応じて、高度な USER_CONF 変数内の別個の 1 行に config response コマンドの 1 つの形式を挿入します。次の選択肢があります。

- アクティブ応答の試行回数のみを指定するには、次のコマンドを挿入します。
  `config response: attempts att`

  例：`config response: attempts 10`
- アクティブ応答インターフェイスのみを指定するには、次のコマンドを挿入します。
  `config response: device dev`

  例：`config response: device eth0`
- アクティブ応答の試行回数とアクティブ応答インターフェイスの両方を指定するには、次のコマンドを挿入します。
  `config response: attempts att, device dev`

  例：`config response: attempts 10, device eth0`

  引数の説明

  att は、受信側ホストにパケットを受け入れさせるために、現在の接続枠で各 TCP リセット パケットを挿入する試行回数（1 ～ 20）です。この*連続試行*はパッシブ展開でのみ効果があります。インライン展開の場合、システムはトリガー パケットの代わりにリセット パケットをストリームに直接挿入します。ICMP 到着可能な 1 つのアクティブ応答のみが送信されます。

  dev は、パッシブ展開でシステムからアクティブ応答を送信したり、インライン展開でアクティブ応答を挿入したりするための代替インターフェイスです。

イベントのフィルタリング

ライセンス：Protection

detection_filter キーワードを使用すると、指定された時間内に指定された数のパケットがルールをトリガーとして使用しない限り、ルールでイベントが生成されないようにすることができます。これにより、早すぎるタイミングでルールがイベントを生成することを回避できます。たとえば、数秒間にログイン試行が 2 ～ 3 回失敗することは想定の範囲内ですが、同じ時間内に多数の試行が発生した場合はブルートフォース アタックを示唆している可能性があります。

detection_filter キーワードの必須の引数は、送信元/宛先のどちらの IP アドレスをシステムで追跡するか、イベントをトリガーする前に検出基準が満たされるべき回数、およびカウントの継続時間を定義します。

イベントのトリガーを遅らせるには、次の構文を使用します。

```
track by_src/by_dst, count count, seconds number_of_seconds
```

track 引数は、ルールの検出基準を満たすパケット数をカウントするときに、パケットの送信元 IP アドレスと宛先 IP アドレスのどちらを使用するかを指定します。システムでイベント インスタンスを追跡する方法を指定するには、次の表の中から引数値を選択します。

表 36-54　detection_filter の追跡引数

引数	説明
by_src	送信元 IP アドレスによる検出基準カウント。
by_dst	宛先 IP アドレスによる検出基準カウント。

count 引数は、ルールでイベントを生成する前に、指定された時間内に指定された IP アドレスのルールをトリガーすべきパケットの数を指定します。

seconds 引数は、ルールでイベントを生成する前に、指定された数のパケットがルールをトリガーすべき時間枠を秒数で指定します。

パケット内でコンテンツ foo を検索するルールが、次の引数を含む detection_filter キーワードを使用するとします。

```
track by_src, count 10, seconds 20
```

この例のルールは、特定の送信元 IP アドレスから 20 秒以内に 10 個のパケットで foo を検出するまでは、イベントを生成しません。システムが最初の 20 秒以内に foo を含むパケットを 7 つしか検出しなかった場合は、イベントが生成されません。しかし、最初の 20 秒間で foo が 40 回出現した場合は、ルールで 30 個のイベントが生成され、20 秒が経過するとカウントが再開されます。

しきい値と detection_filter キーワードの比較

detection_filter キーワードは、非推奨の threshold キーワードに代わるものです。threshold キーワードは、下位互換性を維持するために引き続きサポートされていますが、侵入ポリシー内で設定されるしきい値と同じ機能です。

detection_filter キーワードは、パケットがルールをトリガーとして使用する前に適用される検出機能です。ルールは、指定されたパケット カウントの前に検出されたトリガー パケットに関してイベントを生成しません。また、インライン展開では、パケットを破棄するようルールで設定されていても、そのようなパケットを破棄しません。逆に、指定されたパケット カウントの後に出現する、ルールをトリガーとして使用するパケットに関してルールはイベントを生成します。また、インライン展開でパケットを破棄するよう設定されている場合は、そのようなパケットを破棄します。

しきい値は、検出アクションを発生させないイベント通知機能です。これは、パケットがイベントをトリガーとして使用した後に適用されます。インライン展開において、パケットを破棄するよう設定されたルールは、ルールしきい値とは無関係に、ルールをトリガーとして使用するすべてのパケットを破棄します。

侵入ポリシー内で detection_filter キーワードを侵入イベントしきい値、侵入イベント抑制、およびレート ベースの攻撃防御機能と任意に組み合わせて使用できることに注意してください。また、侵入ポリシー内の侵入イベントしきい値機能と組み合わせて非推奨の threshold キーワードを使用するインポートされたローカル ルールを有効にした場合、ポリシー検証が失敗するこ

とに注意してください。詳細については、イベントしきい値の設定（32-26 ページ）、侵入ポリシー単位の抑制の設定（32-31 ページ）、動的ルール状態の設定（32-36 ページ）、およびローカル ルール ファイルのインポート（66-22 ページ）を参照してください。

攻撃後トラフィックの評価

ライセンス：Protection

ホストまたはセッションに関する追加のトラフィックをログに記録するようシステムに指示するには、tag キーワードを使用します。tag キーワードを使って検出するトラフィックのタイプと量を指定するときには、次の構文を使用します。

tagging_type, count, metric, optional_direction

次の 3 つの表に、その他の使用可能な引数について説明します。

2 つのタイプのタグ機能から選択できます。次の表に、これらのタグ機能の説明を示します。侵入ルールでルール ヘッダー オプションのみを設定した場合、session タグ引数タイプによって、同じセッションからのパケットが別のセッションからのパケットのように記録されることに注意してください。同じセッションからのパケットをまとめてグループ化するには、同じ侵入ルール内で 1 つ以上のルール オプション（flag キーワードや content キーワードなど）を設定します。

表 36-55　tag の引数

引数	説明
session	ルールをトリガーとして使用したセッション内のパケットをログに記録します。
ホスト	ルールをトリガーとして使用したパケットを送信したホストからのパケットをログに記録します。ホストからのトラフィックのみ（src）、またはホストへのトラフィックのみ（dst）を記録する方向修飾子を追加できます。

ログに記録するトラフィック量を指定するには、次の引数を使用します。

表 36-56　count 引数

引数	説明
count	ルールがトリガーとして使用された後にログに記録するパケット数または秒数。 この単位を指定するには、count 引数の後に測定基準引数を使用します。

次の表の中から、トラフィックの時間または量ごとにログで使用する測定基準を選択してください。

注意　高帯域ネットワークでは 1 秒あたり数千パケットが発生する可能性があり、大量のパケットにタグを付けるとパフォーマンスに重大な影響が及ぶ可能性があるため、必ずネットワーク環境に合わせてこの設定を調整してください。

表 36-57　ログの測定基準引数

引数	説明
packets	ルールのトリガー後に、カウントで指定されるパケット数をログに記録します。
秒	ルールのトリガー後に、カウントで指定される秒数の間、トラフィックを記録します。

たとえば、次の tag キーワード値を使用するルールがトリガーとして使用された場合、

```
host, 30, seconds, dst
```

次の 30 秒間にクライアントからホストに送信されるすべてのパケットがログに記録されます。

複数のパケットに及ぶ攻撃の検出

ライセンス：Protection

状態名をセッションに割り当てるには、flowbits キーワードを使用します。すでに名前が付けられた状態に基づいてセッション内の後続パケットを分析することにより、システムは単一セッション内で複数のパケットに及ぶエクスプロイトを検出して警告を出すことができます。

flowbits 状態名は、セッションの特定部分でパケットに割り当てられるユーザ定義のラベルです。パケットの内容に基づいてパケットに状態名を付けると、警告の必要のないパケットと有害なパケットを区別しやすくなります。管理対象デバイスごとに最大 1024 個の状態名を定義できます。たとえば、ログイン成功後にのみ発生することがわかっている有害パケットについて警告するには、flowbits キーワードを使用して、初期ログイン試行を構成するパケットを除去することにより、有害パケットに焦点を絞ることができます。このような機能を実装するには、まず、セッション内のすべてのログイン確立済みパケットに logged_in 状態のラベルを付けるルールを作成した後、2 番目のルールを作成し、最初のルールで設定された状態を持つパケットを検査してそのようなパケットだけを処理する flowbits をそのルールに含めます。ユーザがログイン済みかどうかを判断するために flowbits を使用する例については、state_name を使用した flowbits の例（36-104 ページ）を参照してください。

オプションの *group name* を使用すると、状態のグループに状態名を含めることができます。1 つの状態名は複数のグループに属することができます。グループに関連付けられていない状態は相互排他的ではないため、トリガーとして使用されたルールがグループに関連付けられていない状態を設定した場合、現在設定されている他の状態には影響がありません。グループに状態名を含めて、同じグループ内の別の状態を解除することで誤検出を防止する方法については、誤検出を発生させる flowbits の例（36-105 ページ）の例を参照してください。

次の表は、flowbits キーワードで使用できる演算子、状態、およびグループのさまざまな組み合わせを示しています。なお、状態名には、英数字、ピリオド（.）、アンダースコア（_）、およびダッシュ（-）を含めることができます。

表 36-58　flowbits のオプション

演算子	状態オプション	グループ	説明
設定	state_name	オプション	パケットに関する指定された状態を設定します。グループが定義されている場合は、指定されたグループ内で状態を設定します。
	state_name&state_name	オプション	パケットに関する指定された状態を設定します。グループが定義されている場合は、指定されたグループ内で状態を設定します。
setx	state_name	入力必須	指定されたグループ内でパケットに関して指定された状態を設定し、グループ内の他のすべての状態を解除します。
	state_name&state_name	入力必須	指定されたグループ内でパケットに関して指定された状態を設定し、グループ内の他のすべての状態を解除します。

表 36-58　flowbits のオプション(続き)

演算子	状態オプション	グループ	説明
unset	state_name	グループなし	パケットに関する指定された状態を解除します。
	state_name&state_name	グループなし	パケットに関する指定された状態を解除します。
	すべて	入力必須	指定されたグループ内のすべての状態を解除します。
toggle	state_name	グループなし	指定された状態が設定されている場合はそれを解除し、指定された状態が解除されている場合にはそれを設定します。
	state_name&state_name	グループなし	指定された複数の状態が設定されている場合はそれらを解除し、指定された複数の状態が解除されている場合はそれらを設定します。
	すべて	入力必須	指定されたグループ内で設定されているすべての状態を解除し、指定されたグループ内で解除されているすべての状態を設定します。
isset	state_name	グループなし	指定された状態がパケット内で設定されているかどうかを判別します。
	state_name&state_name	グループなし	指定された複数の状態がパケット内で設定されているかどうかを判別します。
	state_name|state_name	グループなし	指定されたいずれかの状態がパケット内で設定されているかどうかを判別します。
	任意	入力必須	指定されたグループ内で、いずれかの状態が設定されているかどうかを判別します。
	すべて	入力必須	指定されたグループ内で、すべての状態が設定されているかどうかを判別します。
isnotset	state_name	グループなし	指定された状態がパケット内で設定されていないかどうかを判別します。
	state_name&state_name	グループなし	指定された複数の状態がパケット内で設定されていないかどうかを判別します。
	state_name|state_name	グループなし	指定されたいずれかの状態が、パケット内で設定されていないかどうかを判別します。
	任意	入力必須	パケット内でいずれかの状態が設定されていないかどうかを判別します。
	すべて	入力必須	パケット内ですべての状態が設定されていないかどうかを判別します。
リセット	(状態なし)	オプション	すべてのパケットのすべての状態を解除します。グループが指定されている場合、グループ内のすべての状態を解除します。
noalert	(状態なし)	グループなし	イベント生成を抑制するには、これを他の演算子と組み合わせて使用します。

flowbits キーワードを使用するときには、次の点に注意してください。

- setx 演算子を使用する場合、指定した状態は、指定したグループ以外のグループに属することができません。
- setx 演算子を複数回定義して、それぞれのインスタンスで別々の状態と同じグループを指定できます。
- setx 演算子を使用してグループを指定する場合、そのグループに対して set、toggle、unset 演算子を使用することはできません。
- isset 演算子と isnotset 演算子は、指定された状態がグループに含まれるかどうかに関係なく、その状態を評価します。
- 侵入ポリシーの保存時、侵入ポリシーの再適用時、および（アクセス コントロール ポリシーで参照される侵入ポリシー数に関係なく）アクセス コントロール ポリシーの適用時には、グループ指定の**ない** isset または isnotset 演算子を含むルールを有効にした場合、対応する状態名とプロトコルに関する flowbits 割り当て（set、setx、unset、toggle）に影響する 1 つ以上のルールを有効にしないと、対応する状態名の flowbits 割り当てに影響するすべてのルールが有効になります。
- 侵入ポリシーの保存時、侵入ポリシーの再適用時、および（アクセス コントロール ポリシーで参照される侵入ポリシー数に関係なく）アクセス コントロール ポリシーの適用時には、グループを指定**した** isset 演算子または isnotset 演算子を含むルールを有効にした場合、flowbits 割り当て（set、setx、unset、toggle）に影響し、対応するグループ名を定義するすべてのルールもまた有効になります。

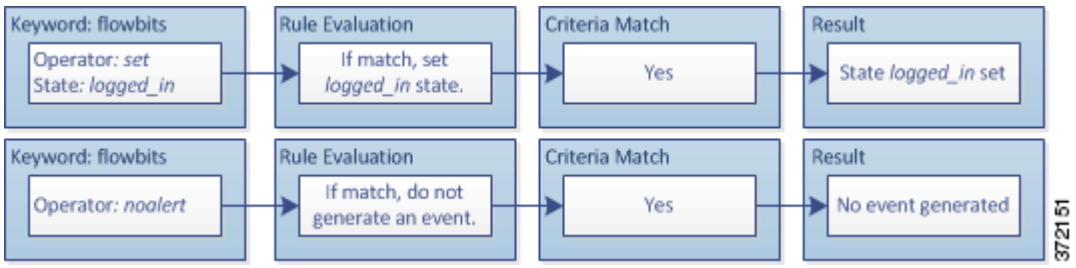
state_name を使用した flowbits の例

Bugtraq ID #1110 に記述されている IMAP 脆弱性について考えてみます。この脆弱性は、IMAP の実装（具体的には LIST、LSUB、RENAME、FIND、および COPY コマンド）で見られます。ただし、攻撃者がこの脆弱性を悪用するには、IMAP サーバにログインする必要があります。IMAP サーバからの LOGIN 確認とそれに続くエクスプロイトは必然的に別々のパケットに存在するため、このエクスプロイトを検出する非フロー ベースのルールを作成するのは困難です。flowbits キーワードを使って一連のルールを作成すると、ユーザが IMAP サーバにログイン済みかどうかを追跡し、ログイン済みの場合は、いずれかの攻撃が検出された時点でイベントを生成することができます。ユーザがログイン済みでない場合、攻撃によって脆弱性が悪用されることはないため、イベントが生成されません。

下記の 2 つのルール フラグメントはこの例を示しています。最初のルール フラグメントは IMAP サーバからの IMAP ログイン確認を検索します。

```
alert tcp any 143 -> any any (msg:"IMAP login"; content:"OK
LOGIN"; flowbits:set,logged_in; flowbits:noalert;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。

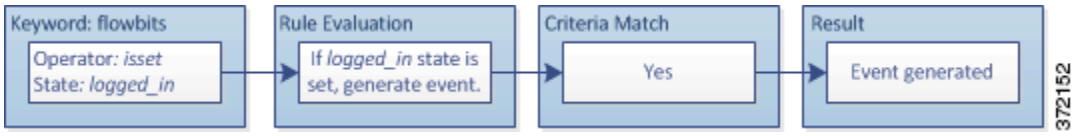


flowbits:set は logged_in 状態を設定しますが、flowbits:noalert がアラートを抑制することに注意してください。これは、IMAP サーバ上で多数の無害なログイン セッションが見つかる可能性があるためです。

次のルール フラグメントは LIST 文字列を検索しますが、セッション内の先行パケットの結果として logged_in 状態が設定済みでない限り、イベントを生成しません。

```
alert tcp any any -> any 143 (msg:"IMAP LIST";
content:"LIST"; flowbits:isset,logged_in;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。



この場合、最初のフラグメントを含むルールが先行パケットによってトリガーとして使用した場合、2 番目のフラグメントを含むルールがトリガーとして使用し、イベントを生成します。

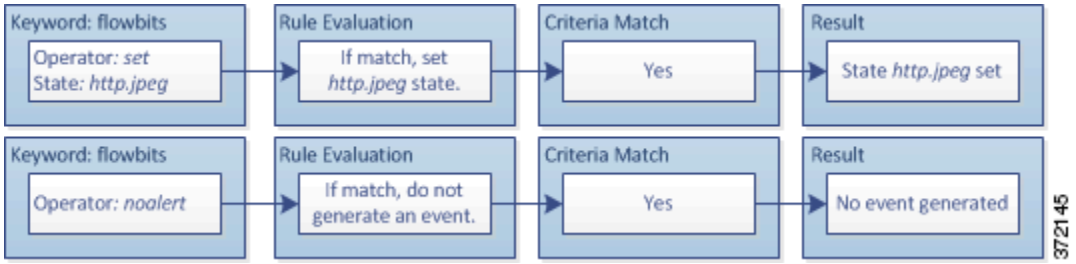
誤検出を発生させる flowbits の例

後続パケット内コンテンツが、効力を失った状態を持つルールに一致することによって誤検出イベントが発生する可能性があります。複数のルールで設定された複数の状態名をグループに含めることでこれを回避できます。次の例は、複数の状態名をグループに含めない場合に誤検出が発生する可能性があることを示しています。

1 つのセッションで次の 3 つのルール フラグメントがこの順序でトリガーとして使用される場合を考えてみます。

```
(msg:"JPEG transfer"; content:"image/";pcre:"/^Content-
Type\x3a(\s*|\s*\r?\n\s+)image\x2fp?jpe?g/smi";
flowbits:set,http.jpeg; flowbits:noalert;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。

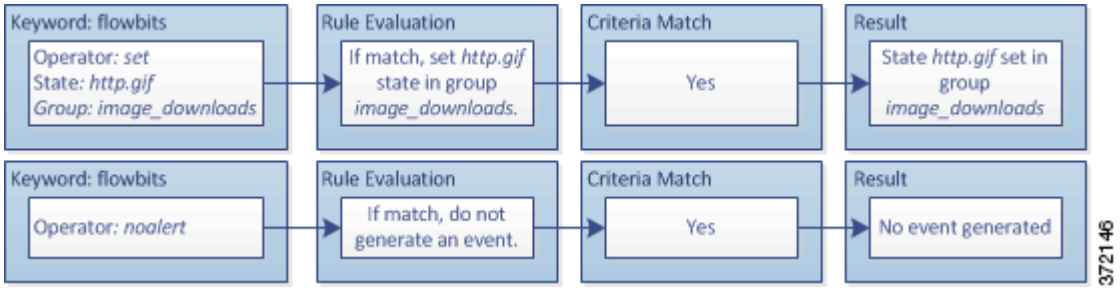


最初のルール フラグメント内の content キーワードと pcre キーワードが JPEG ファイル ダウンロードに一致し、flowbits:set,http.jpeg が http.jpeg flowbits 状態を設定し、flowbits:noalert はルールでのイベント生成を抑制します。イベントが生成されない理由は、このルールの目的がファイル ダウンロードを検出して flowbits 状態を設定することだからです。これにより、1 つ以上のコンパニオン ルールで状態名を検査して有害コンテンツを探し、有害コンテンツが検出された時点でイベントを生成できます。

次のルール フラグメントは、上記の JPEG ファイル ダウンロードに続く GIF ファイル ダウンロードを検出します。

```
(msg:"GIF transfer"; content:"image/"; pcre:"/^Content-
Type\x3a(\s*|\s*\r?\n\s+)image\x2fgif/smi";
flowbits:set,http.tif,image_downloads; flowbits:noalert;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。

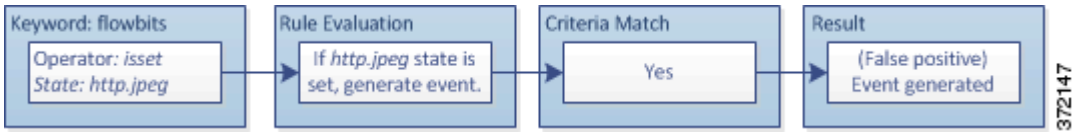


2 番目のルール内の content キーワードと pcre キーワードは GIF ファイル ダウンロードを照合し、flowbits:set,http.tif は http.tif flowbit ステートを設定し、flowbits:noalert はルールでのイベント生成を抑制します。最初のルール フラグメントで設定された http.jpeg 状態が不要になっても引き続き設定されていることに注意してください。これは、後続の GIF ダウンロードが検出されたときに JPEG ダウンロードが既に終了しているはずであるためです。

次に示す 3 番目のルール フラグメントは最初のルール フラグメントのコンパニオンです。

```
(msg:"JPEG exploit";
flowbits:isset,http.jpeg;content:"|FF|"; pcre:"
/\xFF[\xE1\xE2\xED\xFE]\x00[\x00\x01]/";)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。



3 番目のルール フラグメントでは、もはや無意味になった http.jpeg ステートが設定されていることを flowbits:isset,http.jpeg が判別し、content と pcre は（GIF ファイルでは無害でも）JPEG ファイル内では有害とみなされるコンテンツを照合します。3 番目のルール フラグメントによって、JPEG ファイル内に存在しないエクスプロイトに関する誤検出イベントが生成されます。

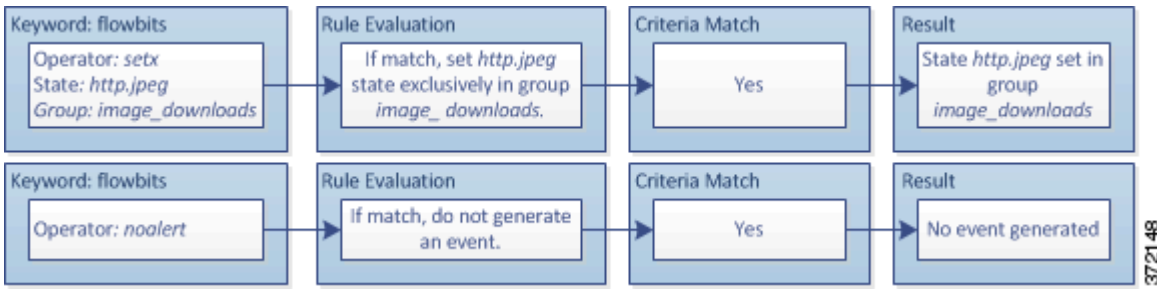
誤検出を防止するための flowbits の例

次の例は、状態名をグループに含めて setx 演算子を使用することで、どのように誤検出を防止できるかを示しています。

前の例とほぼ同じケースを考えます。ただし、最初の 2 つのルールで、同じ状態グループに 2 つの異なる状態名が含まれるようになった点が異なります。

```
(msg:"JPEG transfer"; content:"image/";pcre:"/^Content-
Type\x3a(\s*|\s*\r?\n\s+)image\x2fp?jpe?g/smi";
flowbits:setx,http.jpeg,image_downloads; flowbits:noalert;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。

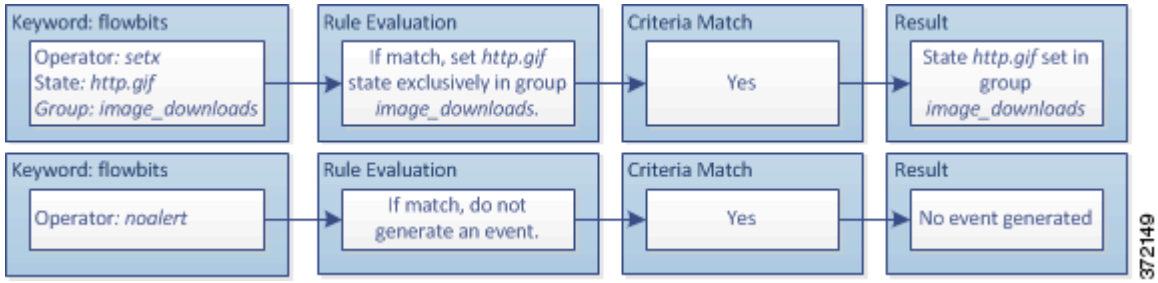


最初のルール フラグメントが JPEG ファイル ダウンロードを検出すると、flowbits:setx,http.jpeg,image_downloads キーワードが flowbits 状態を http.jpeg に設定し、その状態を image_downloads グループに含めます。

その後、次のルールが後続の GIF ファイル ダウンロードを検出します。

```
(msg:"GIF transfer"; content:"image/"; pcre:"/^Content-
Type\x3a(\s*|\s*\r?\n\s+)image\x2fgif/smi";
flowbits:setx,http.tif,image_downloads; flowbits:noalert;)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。

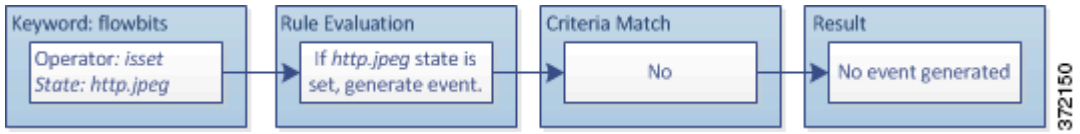


2 番目のルール フラグメントが GIF ダウンロードに一致すると、flowbits:setx,http.tif,image_downloads キーワードが http.tif flowbits 状態を設定し、グループ内の他の状態である http.jpeg を設定解除します。

次に示す 3 番目のルール フラグメントで誤検出は発生しません。

```
(msg:"JPEG exploit";
flowbits:isset,http.jpeg;content:"|FF|"; pcre:"/
\xFF[\xE1\xE2\xED\xFE]\x00[\x00\x01]/";)
```

次の図は、上記のルール フラグメントにおける flowbits キーワードの効果を示しています。



flowbits:isset,http.jpeg が false であるため、ルール エンジンはルールの処理を停止し、イベントは生成されません。こうして、GIF ファイル内のコンテンツが JPEG ファイルに関するエクスプロイト コンテンツと一致した場合でも誤検出が回避されます。

HTTP エンコードのタイプと位置によるイベントの生成

ライセンス：Protection

http_encode キーワードを使用すると、HTTP URI、HTTP ヘッダーの非 cookie データ、HTTP 要求ヘッダーの cookie、HTTP 応答の set-cookie データのいずれかにおいて、正規化前の HTTP 要求または応答内のエンコード タイプに基づいてイベントを生成できます。

HTTP 応答と HTTP cookie を検査し、http_encode キーワードを使用しているルールに一致したものを返すように、HTTP Inspect プリプロセッサを設定する必要があります。詳細については、HTTP トラフィックのデコード（27-34 ページ）とサーバレベル HTTP 正規化オプションの選択（27-36 ページ）を参照してください。

また、侵入ルール内の http_encode キーワードで特定のエンコード タイプによってイベントがトリガーとして使用されるようにするには、HTTP Inspect プリプロセッサ設定で個々の特定のエンコード タイプのデコード オプションとアラート オプションの両方を有効にする必要があります。詳細については、サーバレベル HTTP 正規化エンコード オプションの選択（27-45 ページ）を参照してください。

なお、base36 エンコード タイプは非推奨になりました。下位互換性を維持するために、既存のルールでは base36 引数を使用できますが、これによってルール エンジンが base36 トラフィックを検査することはありません。

次の表は、このオプションでイベントを生成できる、HTTP URI、ヘッダー、cookie、および set-cookie のエンコード タイプを示しています。

表 36-59　http_encode エンコード タイプ

エンコード タイプ	説明
utf8	HTTP Inspect プリプロセッサによる UTF8 エンコード タイプのデコードが有効になっている場合、指定された場所で UTF-8 エンコードを検出します。
double_encode	HTTP Inspect プリプロセッサによるデコードで二重エンコード タイプが有効になっている場合、指定された場所で二重エンコードを検出します。
non_ascii	非 ASCII 文字が検出されても、検出されたエンコード タイプが有効になっていない場合に、指定された場所で非 ASCII 文字を検出します。
uencode	HTTP Inspect プリプロセッサによるデコードで Microsoft %u エンコード タイプが有効になっている場合、指定された場所で Microsoft %u エンコードを検出します。
bare_byte	HTTP Inspect プリプロセッサによるデコードで空白バイト エンコード タイプが有効になっている場合、指定された場所で空白バイト エンコードを検出します。

侵入ルール内で HTTP エンコード タイプと位置を識別するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　http_encode キーワードをルールに追加します。

手順 2　[エンコードする場所（Encoding Location）] ドロップダウン リストで、指定したエンコード タイプを HTTP URI、ヘッダー、または cookie（set-cookie を含む）のいずれで検索するかを選択します。

手順 3　次のいずれかの形式を使用して、1 つ以上のエンコード タイプを指定します。

```
encode_type
encode_type|encode_type|encode_type...
!encode_type
```

ここで、*encode_type* は次のいずれかです。

```
utf8, double_encode, non_ascii, uencode, bare_byte
```

除外（!）演算子と OR（|）演算子を一緒に使用できないことに注意してください。

手順 4　オプションで、複数の http_encode キーワードを同じルールに追加すると、それぞれの条件が AND 結合されます。たとえば、次の条件を含む 2 つのキーワードを入力します。

最初のキーワード http_encode では：

- **エンコード ロケーション**：HTTP URI
- **エンコード タイプ**：utf8

追加のキーワード http_encode では:

- **エンコード ロケーション**:HTTP URI
- **エンコード タイプ**:uencode

この設定例では、HTTP URI で UTF8 および Microsoft IIS %u エンコードを検索します。

ファイル タイプとバージョンの検出

ライセンス:Protection

file_type と file_group キーワードを使用すると、タイプとバージョンに基づいて、FTP、HTTP、SMTP、IMAP、POP3、NetBIOS-ssn（SMB）を介して伝送されるファイルを検出できます。1 つの侵入ルール内で複数の file_type キーワードや file_group キーワードを使用**しない**でください。

ヒント　脆弱性データベース（VDB）を更新すると、最新のファイル タイプ、バージョン、およびグループがルール エディタに表示されます。詳細については、脆弱性データベースの更新（66-14 ページ）を参照してください。

file_type または file_group キーワードに一致するトラフィックに対し侵入イベントを生成するには、特定のプリプロセッサを有効にする**必要があります**。

表 36-60　file_type および file_group の侵入イベントの生成

伝送プロトコル	必要なプリプロセッサまたはプリプロセッサ オプション
FTP	FTP/Telnet のプリプロセッサおよび [Normalize TCP Payload] インライン正規化プリプロセッサオプション。FTP および Telnet トラフィックのデコード（27-20 ページ）およびインライン トラフィックの正規化（29-7 ページ）を参照してください。
HTTP	HTTP Inspect プリプロセッサ。HTTP トラフィックのデコード（27-34 ページ）を参照してください。
SMTP	SMTP プリプロセッサ。SMTP トラフィックのデコード（27-65 ページ）を参照してください。
IMAP	IMAP プリプロセッサ。IMAP トラフィックのデコード（27-58 ページ）を参照してください。
POP3	POP プリプロセッサ。POP トラフィックのデコード（27-62 ページ）を参照してください。
NetBIOS-ssn（SMB）	[SMB File Inspection] DCE/RPC プリプロセッサ オプション。DCE/RPC トラフィックのデコード（27-2 ページ）を参照してください。

詳細については、次の項を参照してください。

- file_type（36-110 ページ）
- file_group（36-111 ページ）

file_type

`file_type` キーワードを使用すると、トラフィック内で検出対象となるファイルのタイプとバージョンを指定できます。ファイル タイプ引数（**JPEG** や **PDF** など）は、トラフィックで検出するファイルの形式を識別します。

(注)　同じ侵入ルール内で file_type キーワードを別の file_type キーワードまたは file_group キーワードと一緒に使用**しない**でください。

デフォルトでは [任意のバージョン(Any Version)] が選択されますが、一部のファイル タイプではバージョン オプション(たとえば PDF バージョン **1.7**)を選択することにより、トラフィックで検出対象となる特定のファイル タイプ バージョンを識別できます。

最新のファイル タイプとバージョンを表示して設定するには、VDB を更新してください。詳細については、脆弱性データベースの更新(66-14 ページ)を参照してください。

侵入ルール内でファイル タイプとバージョンを選択するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ルールの作成(Create Rule)] ページで、ドロップダウン リストから [file_type] を選択して、[オプションの追加(Add option)] をクリックします。

file_type キーワードが表示されます。

手順 2　ドロップダウンリストから 1 つ以上のファイル タイプを選択します。ファイル タイプを選択すると、引数が自動的にルールに追加されます。

ルールからファイル タイプ引数を削除するには、削除するファイル タイプの横にある削除アイコン( )をクリックします。

手順 3　オプションで、各ファイル タイプのターゲット バージョンをカスタマイズします。デフォルトでは [任意のバージョン(Any Version)] が選択されますが、いくつかのファイル タイプでは、個別のターゲット バージョンを選択できます。

(注)　VDB を更新すると、最新のファイル タイプとバージョンがルール エディタに表示されます。[任意のバージョン(Any Version)] を選択した場合、新しいバージョンが今後の VDB 更新に追加されたときにそのバージョンを含めるよう、システムによってルールが設定されます。

file_group

file_group キーワードを使用すると、シスコにより定義された類似のファイル タイプから成るグループを選択して、トラフィック内で検出できます(**マルチメディア**、**オーディオ** など)。また、ファイル グループには、グループ内の各ファイル タイプに関するシスコ定義のバージョンも含まれています。

(注)　同じ侵入ルール内で file_group キーワードを別の file_group キーワードまたは file_type キーワードと一緒に使用**しない**でください。

最新のファイル グループを表示して設定するには、VDB を更新してください。詳細については、脆弱性データベースの更新(66-14 ページ)を参照してください。

侵入ルール内でファイル グループを選択するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [file_group] を選択して、[オプションの追加（Add option）] をクリックします。

`file_group` キーワードが表示されます。

手順 2　オプションで、グループ内のファイル タイプのバージョン情報を表示するには、ファイル グループの上にカーソルを移動し、[(Show Version Info)] をクリックします。

ファイル グループ情報が展開されて、バージョンが表示されます。

手順 3　ルールに追加するファイル グループを選択します。

特定のペイロード タイプを指し示す

ライセンス：Protection

`file_data` キーワードは、`content`、`byte_jump`、`byte_test`、`pcre` などの他のキーワードで使用可能な位置引数の参照として機能するポインタです。`file_data` キーワードが指し示すデータのタイプは、検出されるトラフィックによって決まります。`file_data` キーワードを使用すると、次のペイロード タイプの先頭を指し示すことができます。

- HTTP 応答本文

  HTTP 応答パケットを検査するには、HTTP Inspect プリプロセッサを有効にして、HTTP 応答を検査するようプリプロセッサを設定する必要があります。詳細については、HTTP トラフィックのデコード（27-34 ページ）、およびサーバレベル HTTP 正規化オプションの選択（27-36 ページ）の「**Inspect HTTP Responses**」を参照してください。HTTP Inspect プリプロセッサが HTTP 応答本文データを検出した場合に、`file_data` キーワードが一致します。

- 非圧縮 gzip ファイル データ

  HTTP 応答本文内の非圧縮 gzip ファイルを検査するには、HTTP Inspect プリプロセッサを有効にする必要があります。さらに、HTTP 応答を検査して HTTP 応答本文内の gzip 圧縮ファイルを復元するように、プリプロセッサを設定する必要があります。詳細については、HTTP トラフィックのデコード（27-34 ページ）、およびサーバレベル HTTP 正規化オプションの選択（27-36 ページ）の「**Inspect HTTP Responses**」と「**Inspect Compressed Data**」の各オプションを参照してください。`file_data` キーワードは、HTTP Inspect プリプロセッサが HTTP 応答本文内で非圧縮 gzip データを検出した場合に一致します。

- 正規化された JavaScript

  正規化された JavaScript データを検査するには、HTTP Inspect プリプロセッサを有効にして、HTTP 応答を検査するようプリプロセッサを設定する必要があります。詳細については、HTTP トラフィックのデコード（27-34 ページ）、およびサーバレベル HTTP 正規化オプションの選択（27-36 ページ）の「**Inspect HTTP Responses**」を参照してください。`file_data` キーワードは、HTTP Inspect プリプロセッサが応答本文データ内で JavaScript を検出した場合に一致します。

- SMTP ペイロード

  SMTP ペイロードを検査するには、SMTP プリプロセッサを有効にする必要があります。詳細については、SMTP デコードの設定（27-70 ページ）を参照してください。`file_data` キーワードは、SMTP プリプロセッサが SMTP データを検出した場合に一致します。

- SMTP、POP、または IMAP トラフィック内のエンコードされた電子メール添付ファイル

  SMTP、POP、または IMAP トラフィック内の電子メール添付ファイルを検査するには、それぞれ SMTP、POP、または IMAP プリプロセッサを単独で、または任意に組み合わせて有効にする必要があります。その後、有効にしたプリプロセッサごとに、デコード対象のそれぞれの添付ファイル エンコード タイプをデコードするようプリプロセッサが設定されていることを確認する必要があります。プリプロセッサごとに設定可能な添付ファイル デコード オプションは、[Base64 デコード深さ（Base64 Decoding Depth）]、[7 ビット/8 ビット/バイナリ デコード深さ（7-Bit/8-Bit/Binary Decoding Depth）]、[Quoted Printable デコード深さ（QuotedPrintable Decoding Depth）]、および [Unix-to-Unix デコード深さ（Unix-to-Unix Decoding Depth）] です。詳細については、IMAP トラフィックのデコード（27-58 ページ）、POP トラフィックのデコード（27-62 ページ）、および SMTP トラフィックのデコード（27-65 ページ）を参照してください。

  1 つのルール内で複数の `file_data` キーワードを使用できます。

特定のペイロード タイプの先頭を指し示すには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [file_data] を選択して、[オプションの追加（Add Option）] をクリックします。

`file_data` キーワードが表示されます。

`file_data` キーワードには引数がありません。

パケット ペイロードの先頭を指し示す

ライセンス：Protection

`pkt_data` キーワードは、`content`、`byte_jump`、`byte_test`、`pcre` などの他のキーワードで使用可能な位置引数の参照として機能するポインタです。

正規化された FTP、Telnet、または SMTP トラフィックが検出された場合、`pkt_data` キーワードは、正規化されたパケット ペイロードの先頭を指します。その他のトラフィックが検出された場合、`pkt_data` キーワードは、未加工の TCP または UDP ペイロードの先頭を指します。

侵入ルールで検査するために、該当するトラフィックをシステムで正規化するには、次の正規化オプションを有効にする必要があります。

- FTP トラフィックを検査用に正規化するには、FTP および Telnet プリプロセッサの [FTP コマンドでの Telnet エスケープ コードの検出（Detect Telnet Escape codes within FTP commands）] オプションを有効にする必要があります（サーバレベルの FTP オプションの設定（27-27 ページ）を参照）。
- Telnet トラフィックを検査用に正規化するには、FTP & Telnet プリプロセッサの [正規化（Normalize）] Telnet オプションを有効にする必要があります（Telnet オプションについて（27-22 ページ）を参照）。
- SMTP トラフィックを検査用に正規化するには、SMTP プリプロセッサの [正規化（Normalize）] オプションを有効にする必要があります（SMTP デコードについて（27-65 ページ）を参照）。

1 つのルール内で複数の `pkt_data` キーワードを使用できます。

パケット ペイロードの先頭を指し示すには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [pkt_data] を選択して、[オプションの追加（Add Option）] をクリックします。

`pkt_data` キーワードが表示されます。

`pkt_data` キーワードには引数がありません。

Base64 データのデコードと検査

ライセンス：Protection

`base64_decode` キーワードと `base64_data` キーワードを組み合わせて使用すると、指定したデータを Base64 データとしてデコードおよび検査するようルール エンジンに指示できます。この組み合わせは、HTTP PUT 要求や POST 要求内で Base64 エンコード HTTP 認証要求ヘッダーと Base64 エンコード データを検査する場合などに役立ちます。

これらのキーワードは特に、HTTP 要求内の Base64 データをデコードして検査するうえで役立ちます。また、長いヘッダー行を複数行に拡張するために HTTP で使われるのと同じ方法でスペース文字やタブ文字を使用する SMTP などのプロトコルでも、これらを使用できます。この行拡張（折り返しとも言う）を使用するプロトコル内に行拡張が存在しない場合、後続スペース/タブを伴わない復帰または改行が出現した箇所で検査が終了します。

詳細については、次の各項を参照してください。

- base64_decode（36-114 ページ）
- base64_data（36-115 ページ）

base64_decode

ライセンス：Protection

`base64_decode` キーワードは、パケット データを Base64 データとしてデコードするようルール エンジンに指示します。オプションの引数を使用すると、デコードするバイト数と、デコードを開始するデータ内の位置を指定できます。

`base64_decode` キーワードは 1 つのルール内で 1 回だけ使用可能です。また、少なくとも 1 つの `base64_data` キーワードのインスタンスの前にこれを配置する必要があります。詳細については、base64_data（36-115 ページ）を参照してください。

Base64 データをデコードする前に、ルール エンジンは、複数行にわたって折り返された長いヘッダーを元どおりに広げます。ルール エンジンが次のいずれかに遭遇するとデコードが終了します。

- ヘッダー行の末尾
- デコード対象として指定されたバイト数
- パケットの末尾

次の表に、`base64_decode` キーワードで使用可能な引数の説明を示します。

表 36-61　base64_decode のオプション引数

引数	説明
Bytes	デコードするバイト数を指定します。これを指定しない場合、ヘッダー行の末尾またはパケット ペイロード末尾のどちらかが先に出現するまでデコードが続行されます。ゼロ以外の正の値を指定できます。
Offset	パケット ペイロードの先頭を基準にしたオフセットを決定します。さらに **Relative** も指定した場合は、現在の検査位置を基準にしたオフセットを決定します。ゼロ以外の正の値を指定できます。
Relative	現在の検査位置を基準にして検査することを指定します。

Base64 データをデコードするには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [base64_decode] を選択して、[オプションの追加（Add Option）] をクリックします。

`base64_decode` キーワードが表示されます。

手順 2　オプションで、base64_decode のオプション引数の表に示す引数のいずれかを選択します。

base64_data

ライセンス：Protection

`base64_data` キーワードは、`base64_decode` キーワードを使ってデコードされた Base64 データを検査するための参照を提供します。`base64_data` キーワードは、デコードされた Base64 データの先頭から検査を開始するよう設定します。オプションで、`content` や `byte_test` などの他のキーワードで使用可能な位置引数を使用して、検査位置をさらに指定することもできます。

`base64_decode` キーワードを使用した後に `base64_data` キーワードを少なくとも 1 回使用する必要があります。オプションで、`base64_data` を複数回使用して、デコードされた Base64 データの先頭に戻ることができます。

Base64 データを検査するときには、次の点に注意してください。

- 高速パターン マッチ機能を使用できません（詳細については、高速パターン マッチ機能を使用（Use Fast Pattern Matcher）（36-30 ページ）を参照してください）。
- 中間的な HTTP コンテンツ引数を使ってルール内で Base64 検査を中断する場合は、Base64 データをさらに検査する前に、別の `base64_data` キーワードをルールに挿入する必要があります（詳細については、HTTP コンテンツ オプション（36-26 ページ）を参照してください）。

デコードされた Base64 データを検査するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ルールの作成（Create Rule）] ページで、ドロップダウン リストから [base64_data] を選択して、[オプションの追加（Add Option）] をクリックします。

`base64_data` キーワードが表示されます。

ルールの構築

ライセンス：Protection

独自のカスタム 標準テキスト ルール を作成することもできますが、シスコ 提供の既存の 標準テキスト ルール や 共有オブジェクトのルール を変更して、それを新しいルールとして保存することもできます。シスコ 提供の 共有オブジェクトのルール では、送信元/宛先ポートおよび IP アドレスなどのルール ヘッダー情報だけを変更できることに注意してください。共有オブジェクトのルール 内のルール キーワードとルール引数を変更することはできません。

詳細については、次の各項を参照してください。

- 新しいルールの作成（36-116 ページ）
- 既存のルールの変更（36-118 ページ）
- ルールへのコメントの追加（36-119 ページ）
- カスタム ルールの削除（36-120 ページ）

新しいルールの作成

ライセンス：Protection

独自の 標準テキスト ルール を作成できます。

カスタム 標準テキスト ルール では、ルール ヘッダー設定、ルール キーワード、およびルール引数を設定できます。オプションで、特定のプロトコルを使用する、特定の IP アドレスまたはポートを行き来するトラフィックだけをルールで照合するよう、ルール ヘッダーを設定できます。

新しいルールを作成した後、`GID:SID:Rev` という形式のルール番号を使用することで、そのルールをすばやく見つけることができます。すべての 標準テキスト ルール のルール番号は 1 から始まります。ルール番号の 2 番目の部分である Snort ID（SID）番号は、それがローカル ルールまたは シスコ 提供のルールのどちらであるかを示します。新しいルールを作成すると、システムは、ローカル ルールとして次に使用可能な Snort ID 番号をそのルールに割り当て、ローカル ルール カテゴリ内にルールを保存します。ローカル ルールの Snort ID 番号は 1,000,000 から始まり（ただし、ハイ アベイラビリティ ペアのセカンダリ防御センター上で作成された侵入ルールは 1,000,000,000 から）、新しいローカル ルールが作成されるたびに SID が 1 ずつ増えます。ルール番号の最後の部分はリビジョン番号です。新しいルールのリビジョン番号は 1 です。カスタム ルールを変更するたびに、リビジョン番号が 1 ずつ増えます。

（注） システムは、インポートされた侵入ポリシー内のカスタム ルールに新しい SID を割り当てます。詳細については、設定のインポートおよびエクスポート（A-1 ページ）を参照してください。

ルール エディタを使用してカスタム 標準テキスト ルール を作成するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ポリシー（Policies）] > [侵入（Intrusion）] > [ルール エディタ（Rule Editor）] の順に選択します。

[ルール エディタ（Rule Editor）] ページが表示されます。

手順 2　[ルールの作成（Create Rule）] をクリックします。

[ルールの作成（Create Rule）] ページが表示されます。

手順 3　[Message] フィールドに、イベントと一緒に表示するメッセージを入力します。

イベント メッセージの詳細については、イベント メッセージの定義（36-13 ページ）を参照してください。

ヒント　ルール メッセージの指定は必須です。また、空白文字のみ、1 つ以上の引用符のみ、1 つ以上のアポストロフィのみ、あるいは空白文字/引用符/アポストロフィだけの組み合わせでメッセージを構成することはできません。

手順 4　[分類（Classification）] リストから、イベントのタイプを表す分類を選択します。

使用可能な分類の詳細については、侵入イベント分類の定義（36-13 ページ）を参照してください。

手順 5　[アクション（Action）] リストから、作成するルールのタイプを選択します。次のいずれかを使用できます。

- トラフィックがルールをトリガーとして使用したときにイベントを生成するルールを作成するには、[alert] を選択します。
- ルールをトリガーとして使用したトラフィックを無視するルールを作成するには、[pass] を選択します。

手順 6　[プロトコル（Protocol）] リストから、ルールで検査するパケットのトラフィック プロトコル（**tcp**、**udp**、**icmp**、または **ip**）を選択します。

プロトコル タイプの選択方法については、プロトコルの指定（36-5 ページ）を参照してください。

手順 7　[送信元 IP（Source IPs）] フィールドで、ルールをトリガーとして使用するトラフィックの送信元 IP アドレスまたはアドレス ブロックを入力します。[Destination IPs] フィールドで、ルールをトリガーとして使用するトラフィックの宛先 IP アドレスまたはアドレス ブロックを入力します。

ルール エディタで指定できる IP アドレス構文の詳細については、侵入ルールでの IP アドレスの指定（36-5 ページ）を参照してください。

手順 8　[送信元ポート（Source Port）] フィールドで、ルールをトリガーとして使用するトラフィックの送信元ポート番号を入力します。[Destination Port] フィールドで、ルールをトリガーとして使用するトラフィックの受信側ポート番号を入力します。

（注）　プロトコルが `ip` に設定されている場合、システムは侵入ルール ヘッダー内のポート定義を無視します。

ルール エディタで指定できるポート構文の詳細については、侵入ルールでのポートの定義（36-9 ページ）を参照してください。

手順 9　[方向（Direction）] リストから、ルールをトリガーとして使用するトラフィックの方向を示す演算子を選択します。次のいずれかを使用できます。

- [指向性（Directional）]：送信元 IP アドレスから宛先 IP アドレスに移動するトラフィックを照合します
- [双方向（Bidirectional）]：双方向に移動するトラフィックを照合します

手順 10　[検出オプション（Detection Options）] リストから、使用するキーワードを選択します。

手順 11　[オプションの追加（Add option）] をクリックします。

手順 12　追加したキーワードで指定する引数を入力します。ルール キーワードとその使用方法については、ルールでのキーワードと引数について（36-11 ページ）を参照してください。

キーワードと引数を追加するときには、次の操作を実行することもできます。

- 追加した後のキーワードを並べ替えるには、移動するキーワードの横にある上矢印または下矢印をクリックします。
- キーワードを削除するには、そのキーワードの横にある [X] をクリックします。

追加するキーワード オプションごとに、ステップ 10 ～ 12 を繰り返します。

手順 13　ルールを保存するには、[新規保存（Save As New）] をクリックします。

システムは、ルール番号シーケンスの中でローカル ルールとして次に使用可能な Snort ID（SID）番号をルールに割り当て、ローカル ルール カテゴリ内にルールを保存します。

新しい（または変更された）ルールを適切な侵入ポリシー内で有効にして、侵入ポリシーをアクセス コントロール ポリシーの一部として適用するまでは、そのルールに照らしたトラフィックの評価が開始しません。詳細については、アクセス コントロール ポリシーの適用（12-17 ページ）を参照してください。

既存のルールの変更

ライセンス：Protection

カスタム 標準テキスト ルール を変更できます。シスコ 提供の 標準テキスト ルール または 共有オブジェクトのルール を変更して保存すると、そのルールの 1 つ以上の新しいインスタンスが作成されます。

ルールを作成したり、シスコ のルールを変更したりすると、新しいルールまたはリビジョンがローカル ルール カテゴリにコピーされ、100000 より大きい次に使用可能な Snort ID（SID）がそのルールに割り当てられます。

共有オブジェクトのルール では、ヘッダー情報だけを変更することができます。共有オブジェクトのルール 内で使用されるルール キーワードやその引数を変更することはできません。共有オブジェクトのルール のヘッダー情報を変更して変更内容を保存すると、ルールの新しいインスタンスが作成され、ジェネレータ ID（GID）3、およびカスタム ルールとして次に使用可能な SID が割り当てられます。ルール エディタは、共有オブジェクトのルールの新しいインスタンスを予約済み `soid` キーワードにリンクします。これにより、作成したルールが VRT 作成のルールにマップされます。作成した共有オブジェクトのルールのインスタンスを削除できますが、シスコ 提供の共有オブジェクトのルールは削除できません。詳細については、「ルール ヘッダーについて（36-3 ページ）」と「カスタム ルールの削除（36-120 ページ）」を参照してください。

（注）　共有オブジェクトのルール のプロトコルを変更しないでください。変更した場合、ルールの効果がなくなる可能性があります。

ルールを変更するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ポリシー（Policies）] > [侵入（Intrusion）] > [ルール エディタ（Rule Editor）] の順に選択します。

[ルール エディタ（Rule Editor）] ページが表示されます。

手順 2　変更する 1 つ以上のルールを探します。次の選択肢があります。

- ルール カテゴリを参照することによってルールを探すには、フォルダを通して該当するルールまで移動し、そのルールの横にある編集アイコン( )をクリックします。
- 検索機能によってルールを探すには、該当するルールの検索基準(最も単純なものは SID)を入力して [検索(Search)] をクリックします。検索によって返された、該当するルールをクリックします。詳細については、ルールの検索(36-121 ページ)を参照してください。
- ページに表示されるルールを絞り込むことによってルールを探すには、ルール リストの左上にあるフィルタ アイコン( )で示されるテキスト ボックスにルール フィルタを入力します。該当するルールまで移動して、そのルールの横にある編集アイコン( )をクリックします。詳細については、[ルール エディタ(Rule Editor)] ページでのルールのフィルタリング(36-123 ページ)を参照してください。

ルール エディタが開いて、選択したルールが表示されます。

共有オブジェクトのルール を選択した場合は、ルール ヘッダー情報だけがルール エディタに表示されることに注意してください。[ルール エディタ(Rule Editor)] ページで 共有オブジェクトのルール を識別するには、リストの中で数字の 3(GID)で始まる項目を探します(たとえば 3:1000004)。

手順 3　ルールを変更して(ルール オプションの詳細については 新しいルールの作成(36-116 ページ)を参照)、[新規保存(Save As New)] をクリックします。

ルールがローカル ルール カテゴリに保存されます。

ヒント　システム ルールの代わりに、ローカルで変更したルールを使用するには、ルール状態の設定(32-23 ページ)の手順に従ってシステム ルールを非アクティブにした後、ローカル ルールをアクティブにします。

手順 4　変更を適用するには、アクセス コントロール ポリシーの適用(12-17 ページ)の説明に従って侵入ポリシーをアクセス コントロール ポリシーの一部として適用し、アクティブにします。

ルールへのコメントの追加

ライセンス: Protection

任意の侵入ルールにコメントを追加できます。これにより、ルールや、特定されたエクスプロイトまたはポリシー違反に関するコンテキストおよび情報を提供できます。

コメントをルールに追加するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ポリシー(Policies)] > [侵入(Intrusion)] > [ルール エディタ(Rule Editor)] の順に選択します。

[ルール エディタ(Rule Editor)] ページが表示されます。

手順 2　注釈を付けるルールを探します。次の選択肢があります。

- ルール カテゴリを参照することによってルールを探すには、フォルダを通して該当するルールまで移動し、そのルールの横にある編集アイコン( )をクリックします。
- 検索機能によってルールを探すには、該当するルールの検索基準(最も単純な基準は SID)を入力して [検索(Search)] をクリックします。検索で返された、該当するルールをクリックします。詳細については、ルールの検索(36-121 ページ)を参照してください。

- ページに表示されるルールを絞り込むことによってルールを探すには、ルール リストの左上にあるフィルタ アイコン( )で示されるテキスト ボックスでルール フィルタを入力します。該当するルールまで移動して、そのルールの横にある編集アイコン( )をクリックします。詳細については、[ルール エディタ（Rule Editor）] ページでのルールのフィルタリング（36-123 ページ）を参照してください。

ルール エディタが表示されます。

手順 3　[ルール コメント（Rule Comment）] をクリックします。

[ルール コメント（Rule Comment）] ページが表示されます。

手順 4　テキスト ボックスにコメントを入力し、[コメントの追加（Add Comment）] をクリックします。

コメント テキスト ボックスにコメントが保存されます。

ヒント　また、侵入イベントのパケット ビューで、ルール コメントを追加して表示することもできます。詳細については、イベント情報の表示（41-27 ページ）を参照してください。

カスタム ルールの削除

ライセンス：Protection

侵入ポリシーで現在有効になっていないカスタム ルールを削除することができます。シスコ 提供の 標準テキスト ルール や 共有オブジェクトのルール ルールは削除できません。

削除されたルールは削除済みカテゴリに保存されます。削除済みのルールを、新しいルールの基準として使用することができます。ルールの編集方法については、既存のルールの変更（36-118 ページ）を参照してください。

侵入ポリシーの [ルール（Rules）] ページには削除済みカテゴリが表示されないため、削除したカスタム ルールを有効にすることはできません。

なお、[ルールのアップデート（Rule Updates）] ページですべてのローカル ルールを削除することもできます。たとえば、ワンタイム ルール更新の使用（66-18 ページ）を参照してください。

詳細については、次の各項を参照してください。

- カスタム ルールの作成方法については、新しいルールの作成（36-116 ページ）を参照してください。
- ローカル ルールのインポート方法については、ルールの更新とローカル ルール ファイルのインポート（66-16 ページ）を参照してください。
- ルール状態の設定方法については、ルール状態の設定（32-23 ページ）を参照してください。

カスタム ルールを削除するには、次の手順を実行します。

アクセス：Admin/Intrusion Admin

手順 1　[ポリシー（Policies）] > [侵入（Intrusion）] > [ルール エディタ（Rule Editor）] の順に選択します。

[ルール エディタ（Rule Editor）] ページが表示されます。

手順 2　次の 2 つの選択肢があります。

- [ローカル ルールの削除(Delete Local Rules)] をクリックしてから、[OK] をクリックします。

  変更内容が保存された侵入ポリシー内で現在有効になっていないすべてのルールは、ローカル ルール カテゴリから削除され、削除済みカテゴリに移動されます。
- フォルダを通してローカル ルール カテゴリまで移動します。ローカル ルール カテゴリをクリックして展開してから、削除するルールの横にある削除アイコン( )をクリックします。

  ルールがローカル ルール カテゴリから削除され、削除済みカテゴリに移動されます。

  カスタム 標準テキスト ルール にはジェネレータ ID(GID)1 が割り当てられ(たとえば 1:1000012)、カスタム 共有オブジェクトのルール には GID として 3 が割り当てられる(たとえば 3:1000005)ことに注意してください。

ヒント　また、ヘッダー情報を変更して保存した 共有オブジェクトのルール もローカル ルール カテゴリに保管され、それらは GID 3 で列挙されます。独自に変更した 共有オブジェクトのルール を削除できますが、元の 共有オブジェクトのルール は削除できません。

ルールの検索

ライセンス:Protection

FireSIGHT システムには何千もの標準テキスト ルールが含まれています。シスコ脆弱性調査チームは新しい脆弱性やエクスプロイトが発見されるたびにルールを追加し続けています。特定のルールを簡単に検索して、それをアクティブ化、非アクティブ化、または編集することができます。

次の表は、使用可能な検索オプションについて示しています。

表 36-62　ルール検索基準

オプション	説明
シグニチャ ID(Signature ID)	Snort ID(シグネチャ ID とも呼ばれる)に基づいて 1 つのルールを検索するには、Snort ID 番号を入力します。複数のルールを検索するには、複数の Snort ID 番号をカンマで区切ったリストを入力します。このフィールドには 80 文字の制限があります。
ジェネレータ ID(Generator ID)	標準テキスト ルールを検索するには、**1** を選択します。共有オブジェクトのルールを検索するには、**3** を選択します。
メッセージ	特定のメッセージを含むルールを検索するには、ルール メッセージの 1 つの単語を [メッセージ(Message)] フィールドに入力します。たとえば、DNS exploit を検索するには「`DNS`」と入力し、バッファ オーバーフローエクスプロイトを検索するには「`overflow`」と入力します。
プロトコル	特定のプロトコルのトラフィックを評価するルールを検索するには、プロトコルを選択します。プロトコルを選択しない場合、検索結果にはすべてのプロトコルのルールが含まれます。
送信元ポート	指定したポートからの発信パケットを検査するルールを検索するには、送信元ポート番号またはポート関連の変数を入力します。

表 36-62　ルール検索基準（続き）

オプション	説明
[接続先ポート（Destination Port）]	特定のポート宛てのパケットを検査するルールを検索するには、宛先ポート番号またはポート関連の変数を入力します。
ソース IP	指定した IP アドレスからの発信パケットを検査するルールを検索するには、送信元 IP アドレスまたは IP アドレス関連の変数を入力します。
宛先 IP（Destination IP）	指定した IP アドレス宛てのパケットを検査するルールを検索するには、宛先 IP アドレスまたは IP アドレス関連の変数を入力します。
キーワード	特定のキーワードを検索するには、キーワード検索オプションを使用できます。検索対象のキーワードとキーワード値を選択します。また、キーワード値の前に感嘆符（!）を付けると、指定した値以外の値を照合できます。
カテゴリ（Category）	特定のカテゴリ内のルールを検索するには、[カテゴリ（Category）] リストからカテゴリを選択します。
分類（Classification）	特定の分類が設定されたルールを検索するには、[分類（Classification）] リストから分類名を選択します。
ルール状態（Rule State）	特定のポリシー内のルールおよび特定のルール状態を検索するには、最初の [ルール状態（Rule State）] リストからポリシーを選択し、2 番目のリストから状態を選択して、[イベントの生成（Generate Events）]、[ドロップしてイベントを生成（Drop and Generate Events）]、または [無効（Disabled）] に設定されたルールを検索します。

特定のルールを検索する方法：

アクセス：Admin/Intrusion Admin

手順 1　[ポリシー（Policies）] > [侵入（Intrusion）] > [ルール エディタ（Rule Editor）] の順に選択します。

[ルール エディタ（Rule Editor）] ページが表示されます。

手順 2　ツールバーで [検索（Search）] をクリックします。

[検索（Search）] ページが表示されます。

手順 3　ルール検索基準の表に示されるフィールドを使用して、検索基準を追加します。

（注）　ルールを検索するには、少なくとも 1 つの検索基準を指定する必要があります。

手順 4　特定のキーワードを含むルールを検索するには、次の手順に従います。

- [キーワード（Keyword）] セクションのドロップダウン リストから、検索するキーワードを選択します。

  使用可能なキーワードのリストについては、ルールでのキーワードと引数について（36-11 ページ）を参照してください。
- [キーワード（Keyword）] フィールドに、検索する引数を入力します。

手順 5　[検索（Search）] をクリックします。

ページがリロードされ、検索基準に一致するルールのリストが表示されます。

手順 6　ルール（システム ルールの場合はルールのコピー）を表示または編集するには、ハイパーリンクが付いたルール メッセージをクリックします。ルールの編集方法の詳細については、既存のルールの変更（36-118 ページ）を参照してください。

[ルール エディタ（Rule Editor）] ページでのルールのフィルタリング

ライセンス：Protection

[ルール エディタ（Rule Editor）] ページ上でルールをフィルタ処理して、ルールのサブセットを表示させることができます。たとえば、あるルールまたはその状態を変更したいが、数千ものルールの中からそれを見つけるのが困難な場合に、この機能が役立つことがあります。

フィルタを入力すると、1 つ以上の一致するルールを含むフォルダがページに表示され、一致するルールがない場合はメッセージが表示されます。フィルタには、特殊なキーワードとその引数、文字列、引用符で囲んだリテラル文字列、さらに複数のフィルタ条件を区切るスペースを含めることができます。ただし、正規表現、ワイルドカード文字、および否定文字（!）、「大なり」記号（>）、「小なり」記号（<）などの特殊な演算子をフィルタに含めることはできません。

すべてのキーワード、キーワード引数、および文字列では大文字と小文字が区別されません。`gid` キーワードと `sid` キーワードを除くすべての引数と文字列が部分文字列として扱われます。`gid` と `sid` の引数は、完全一致のみを返します。

オプションで、フィルタ処理前の元のページで 1 つのフォルダを展開すると、その後のフィルタ処理でそのフォルダ内の一致が返されるときにフォルダが展開したままになります。探しているルールが多数のルールを含むフォルダ内に存在する場合には、これが役立つことがあります。

1 つのフィルタを後続の別のフィルタで制約することはできません。入力されたフィルタは、ルール データベース全体を検索して、一致するすべてのルールを返します。前回のフィルタ結果がページに表示されている状態でフィルタを入力すると、そのページの内容が消去され、代わりに新しいフィルタの結果が返されます。

フィルタ処理されている場合もされていない場合も、リスト内のルールで同じ機能を使用できます。たとえば、[ルール エディタ（Rule Editor）] ページでは、リストがフィルタ処理されているかどうかに関わらず、リスト内のルールを編集できます。また、ページのコンテキスト メニューの任意のオプションを使用することもできます。

詳細については、次の各項を参照してください。

- ルール フィルタでのキーワードの使用（36-123 ページ）
- ルール フィルタでの文字列の使用（36-125 ページ）
- ルール フィルタでのキーワードと文字列の組み合わせ（36-125 ページ）
- ルールのフィルタリング（36-126 ページ）

ルール フィルタでのキーワードの使用

ライセンス：Protection

各ルール フィルタに、次の形式で 1 つ以上のキーワードを含めることができます。

`keyword:argument`

ここで、*keyword* はルール フィルタ キーワードの表のいずれかのキーワード、*argument* はキーワードに関連する特定のフィールドで検索される単一の、大文字と小文字を区別しない英数字文字列です。

gid と sid を除くすべてのキーワードの引数が部分文字列として扱われます。たとえば、引数 123 によって "12345"、"41235"、"45123" などが返されます。gid と sid の引数は完全一致のみを返します。たとえば、sid:3080 によって SID 3080 のみが返されます。

ヒント　部分的な SID を検索するには、1 つ以上の文字列を使ってフィルタ処理できます。詳細については、ルール フィルタでの文字列の使用(36-125 ページ)を参照してください。

次の表に、ルールのフィルタ処理に使用できる特定のフィルタリング キーワードと引数を示します。

表 36-63　ルール フィルタ キーワード

キーワード	説明	例
arachnids	ルール参照内の Arachnids ID 全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	arachnids:181
bugtraq	ルール参照内の Bugtraq ID 全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	bugtraq:2120
cve	ルール参照内の CVE 番号全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	cve:2003-0109
gid	引数 1 は標準テキスト ルールを返します。引数 3 は共有オブジェクトのルールを返します。詳細については、プリプロセッサ ジェネレータ ID の読み取り(41-44 ページ)および表 32-1(32-2 ページ)を参照してください。	gid:3
mcafee	ルール参照内の McAfee ID 全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	mcafee:10566
msg	ルールの [メッセージ(Message)] フィールド(イベント メッセージとも呼ばれる)の全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント メッセージの定義(36-13 ページ)を参照してください。	msg:chat
nessus	ルール参照内の Nessus ID 全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	nessus:10737
ref	ルール参照内またはルールの [メッセージ(Message)] フィールド内の単一の英数字文字列の全体または一部分に基づいて、1 つ以上のルールを返します。詳細については、「イベント参照の定義(36-15 ページ)」と「イベント メッセージの定義(36-13 ページ)」を参照してください。	ref:MS03-039

表 36-63　ルール フィルタ キーワード(続き)

キーワード	説明	例
SID	完全に一致するシグニチャ ID を持つルールを返します。詳細については、プリプロセッサ ジェネレータ ID の読み取り(41-44 ページ)を参照してください。	sid:235
URL	ルール参照内の URL 全体またはその一部分に基づいて 1 つ以上のルールを返します。詳細については、イベント参照の定義(36-15 ページ)を参照してください。	url:faqs.org

ルール フィルタでの文字列の使用

ライセンス: Protection

各ルール フィルタに 1 つ以上の英数字文字列を含めることができます。文字列はルールの [メッセージ(Message)] フィールド、シグニチャ ID、およびジェネレータ ID を検索します。たとえば、文字列 `123` を指定するとルール メッセージ内の文字列「`Lotus123`」や「`123mania`」などが返され、さらに SID 6123、SID 12375 なども返されます。ルールの [メッセージ(Message)] フィールドの詳細については、イベント メッセージの定義(36-13 ページ)を参照してください。ルール SID と GID の詳細については、プリプロセッサ ジェネレータ ID の読み取り(41-44 ページ)を参照してください。

すべての文字列では大文字と小文字が区別されず、部分的な文字列として扱われます。たとえば、文字列 `ADMIN`、`admin`、または `Admin` はすべて、`"admin"`、`"CFADMIN"`、`"Administrator"` などを返します。

文字列を引用符で囲むと、完全一致を返すことができます。たとえば、引用符付きのリテラル文字列 `"overflow attempt"` は完全一致のみを返しますが、引用符なしの 2 つの文字列 `overflow` と `attempt` で構成されるフィルタは `"overflow attempt"`、`"overflow multipacket attempt"`、`"overflow with evasion attempt"` などを返します。

ルール フィルタでのキーワードと文字列の組み合わせ

ライセンス: Protection

複数のキーワード、文字列、またはその両方をスペースで区切って任意に組み合わせて入力することで、フィルタ結果を絞り込むことができます。結果には、すべてのフィルタ条件に一致するルールが含まれます。

複数のフィルタ条件を任意の順序で入力できます。たとえば、次のフィルタはそれぞれ同じルールを返します。

- `url:at login attempt cve:200`
- `login attempt cve:200 url:at`
- `login cve:200 attempt url:at`

ルールのフィルタリング

ライセンス: Protection

[ルール エディタ(Rule Editor)] ページ上でルールをフィルタ処理して、ルールのサブセットを表示させると、特定のルールを見つけやすくなります。その後で、いずれかのページ機能を使用できます。これには、コンテキスト メニューで使用可能な機能の選択も含まれます。

特定のルールをフィルタ処理するには、次の手順を実行します。

アクセス: Admin/Intrusion Admin

手順 1　[ポリシー(Policies)] > [侵入(Intrusion)] > [ルール エディタ(Rule Editor)] の順に選択します。

[ルール エディタ(Rule Editor)] ページが表示されます。

ルール フィルタ機能は、[ルール エディタ(Rule Editor)] ページで編集するルールを見つけるときに特に役立つことがあります。詳細については、既存のルールの変更(36-118 ページ)を参照してください。

手順 2　オプションで、[ルールのグループ化基準(Group Rules By)] リストで別のグループ化方法を選択します。

ヒント　すべてのサブグループ内のルールの総計が多い場合は、フィルタリングに時間がかかることがあります。これは、ルール自体の数が少なくても、1 つのルールが複数のカテゴリに属していることがあるためです。

手順 3　オプションで、展開するグループの横にあるフォルダをクリックします。

フォルダが展開されて、そのグループ内のルールが表示されます。ルール グループによっては、さらに展開可能なサブグループが存在します。

また、ルールがどのグループに含まれているか予想できる場合は、フィルタ処理前の元のページでそのグループを展開しておくと便利なことがあります。その後のフィルタ処理でそのフォルダ内の一致が返されるとき、およびフィルタ消去アイコン( )をクリックしてフィルタ処理前のページに戻ったときに、グループが展開されたままになります。

手順 4　フィルタ テキスト ボックスをアクティブにするには、ルール リストの左上にあるテキスト ボックス内のフィルタ アイコン( )の右側をクリックします。

手順 5　フィルタ制約を入力し、Enter キーを押します。

フィルタには、キーワードと引数、引用符付きまたは引用符なしの文字列、および複数の条件を区切るスペースを含めることができます。詳細については、[ルール エディタ(Rule Editor)] ページでのルールのフィルタリング(36-123 ページ)を参照してください。

ページが更新されて、一致するルールを少なくとも 1 つ含むグループが表示されます。

手順 6　オプションで、まだ開いていないフォルダを開くと、一致するルールが表示されます。次のフィルタリング選択肢があります。

- 新しいフィルタを入力するには、フィルタ テキスト ボックス内にカーソルを移動してクリックし、そのボックスをアクティブにしてから、フィルタを入力して Enter キーを押します。
- フィルタ処理された現在のリストを消去してフィルタ処理されていないの元のページに戻すには、フィルタ消去アイコン( )をクリックします。

手順 7　オプションで、ページに表示されているルールを通常の方法で変更します。既存のルールの変更(36-118 ページ)を参照してください。

変更内容を有効にするには、アクセス コントロール ポリシーの適用（12-17 ページ）の説明に従って、アクセス コントロール ポリシーの侵入ポリシー部分を適用します。