APPENDIX **B**

シグニチャ エンジンの概要

（注） 現在、Cisco IPS 7.1 をサポートしているプラットフォームは、IPS SSP を搭載した Cisco ASA 5585-X のみです。それ以外の Cisco IPS センサーは、IPS 7.1 を現在サポートしていません。

（注） IPS SSP を搭載した Cisco ASA 5585-X は、ASA 8.2(4.4) 以降および ASA 8.4(2) 以降でサポートされています。ASA 8.3(*x*) ではサポートされていません。

この付録では、IPS シグニチャ エンジンについて説明します。次のような構成になっています。

- 「シグニチャ エンジンの概要」（P.B-2）
- 「Master エンジン」（P.B-4）
- 「正規表現の構文」（P.B-10）
- 「AIC エンジン」（P.B-11）
- 「Atomic エンジン」（P.B-12）
- 「Fixed エンジン」（P.B-27）
- 「Flood エンジン」（P.B-30）
- 「Meta エンジン」（P.B-31）
- 「Multi String エンジン」（P.B-32）
- 「ノーマライザ エンジン」（P.B-33）
- 「Service エンジン」（P.B-36）
- 「State エンジン」（P.B-54）
- 「String エンジン」（P.B-56）
- 「String XL エンジン」（P.B-59）
- 「Sweep エンジン」（P.B-62）
- 「Traffic Anomaly エンジン」（P.B-65）
- 「Traffic ICMP エンジン」（P.B-67）
- 「Trojan エンジン」（P.B-68）

シグニチャ エンジンの概要

シグニチャ エンジンは、特定のカテゴリで多数のシグニチャをサポートすることを目的とした Cisco IPS のコンポーネントです。エンジンは、パーサーとインスペクタで構成されています。各エンジンにはパラメータのセットがあり、パラメータには使用可能な範囲や値のセットがあります。

（注）　Cisco IPS エンジンは、標準化された 正規表現をサポートします。

Cisco IPS には、次のシグニチャ エンジンが含まれます。

- AIC：Web トラフィックの綿密な分析を行います。AIC エンジンは、HTTP セッションに対してより細かな制御を実行して、HTTP プロトコルの不正利用を防ぎます。また、指定されたポート上でトンネルを作成しようとするインスタント メッセージングや gotomypc などのアプリケーションを管理制御できます。また、AIC を使用して、FTP トラフィックを検査し、発行されるコマンドを制御することもできます。AIC エンジンには、AIC FTP と AIC HTTP の 2 つがあります。
- Atomic：Atomic エンジンは、マルチレベルの選択肢を持つ 4 つのエンジンに結合されます。IP + TCP など、レイヤ 3 属性とレイヤ 4 属性を 1 つのシグニチャ内で結合できます。Atomic エンジンでは、標準化された正規表現のサポートが使用されます。Atomic エンジンは、次のタイプで構成されます。
  - Atomic ARP：レイヤ 2 ARP プロトコルを検査します。Atomic ARP エンジンが異なるのは、大半のエンジンはレイヤ 3 IP プロトコルに基づいているためです。
  - Atomic IP Advanced：IPv6 レイヤ 3 および ICMPv6 レイヤ 4 トラフィックを検査します。
  - Atomic IP：IP プロトコル パケット、および関連付けられているレイヤ 4 トランスポート プロトコルを検査します。

    このエンジンを使用することで、IP ヘッダーおよびレイヤ 4 ヘッダー内のフィールドに照合する値を指定でき、正規表現を使用したレイヤ 4 ペイロードの検査が可能になります。

    （注）　すべての IP パケットは、Atomic IP エンジンよって検査されます。このエンジンは、4.x Atomic ICMP、Atomic IP Options、Atomic L3 IP、Atomic TCP、および Atomic UDP の各エンジンに取って代わりました。

  - Atomic IPv6：不正な形式の IPv6 トラフィックによって誘導される 2 つの IOS 脆弱性を検出します。
- Fixed：固定レベルまで並列正規表現照合を実行してから、1 つの正規表現テーブルを使用する検査を停止します。Fixed エンジンには、ICMP、TCP、および UDP の 3 つがあります。
- Flood：ホストおよびネットワークに向けられた ICMP および UDP フラッドを検出します。Flood エンジンには、Flood Host と Flood Net の 2 つがあります。
- Meta：スライディング時間間隔内に、関連した方法で発生するイベントを定義します。このエンジンは、パケットではなくイベントを処理します。
- Multi String：1 つのシグニチャに対する複数の文字列を照合することにより、レイヤ 4 トランスポート プロトコルおよびペイロードを検査します。このエンジンは、ストリームベースの TCP と単一の UDP、および ICMP パケットを検査します。
- Normalizer：IP および TCP ノーマライザが機能する方法を設定し、IP および TCP ノーマライザに関連するシグニチャ イベントに設定を提供します。RFC 準拠を強制できます。

- Service：特定のプロトコルを処理します。Service エンジンは、次のプロトコル タイプに分けられます。
  - DNS：DNS（TCP および UDP）トラフィックを検査します。
  - FTP：FTP トラフィックを検査します。
  - FTP V2：IOS IPS をサポートします。

    このシグニチャ エンジンは、IOS IPS 向けにチューニング済みのプロトコル デコード エンジンを提供します。このエンジンを使用しようとすると、エラー メッセージを受け取ります。
  - Generic：カスタム サービスおよびペイロードをデコードし、ネットワーク プロトコルを一般的に分析します。
  - H225：VoIP トラフィックを検査します。VoIP ネットワークに到達する SETUP メッセージが有効であり、ポリシーで定義されている範囲内にあることを、ネットワーク管理者が確認するのに役立ちます。また、アドレスや、url-id、email-id、および表示情報などの Q.931 文字列フィールドが特定の長さに従っており、そこに潜在的な攻撃パターンが含まれていないことを確認するのにも役立ちます。
  - HTTP：HTTP トラフィックを検査します。WEBPORTS 変数では、HTTP トラフィックの検査ポートを定義します。
  - HTTP V2：IOS IPS をサポートします。

    このシグニチャ エンジンは、IOS IPS 向けにチューニング済みのプロトコル デコード エンジンを提供します。このエンジンを使用しようとすると、エラー メッセージを受け取ります。
  - IDENT：IDENT（クライアントおよびサーバ）トラフィックを検査します。
  - MSRPC：MSRPC トラフィックを検査します。
  - MSSQL：Microsoft SQL トラフィックを検査します。
  - NTP：NTP トラフィックを検査します。
  - P2P：P2P トラフィックを検査します。
  - RPC：RPC トラフィックを検査します。
  - SMB Advanced：Microsoft SMB パケットと Microsoft DCE/RPC（MSRPC）over SMB パケットを処理します。

    （注）　SMB エンジンは、SMB Advanced エンジンに置き換えられました。SMB エンジンがまだ IDM、IME、および CLI に表示される場合でも、このシグニチャは廃止されています。つまり、新規のシグニチャには、対応する古いシグニチャの ID を持つ obsoletes パラメータ セットがあります。新規の SMB Advanced エンジンを使用して、SMB エンジンにあったカスタム シグニチャを書き換えてください。
  - SMPT V1：IOS IPS をサポートします。

    このシグニチャ エンジンは、IOS IPS 向けにチューニング済みのプロトコル デコード エンジンを提供します。このエンジンを使用しようとすると、エラー メッセージを受け取ります。
  - SNMP：SNMP トラフィックを検査します。
  - SSH：SSH トラフィックを検査します。
  - TNS：TNS トラフィックを検査します。
- State：SMTP などのプロトコル内の文字列のステートフル検索を実施します。State エンジンには非表示のコンフィギュレーション ファイルがあります。このファイルは、新規の状態定義をシグニチャ アップデートに配信できるよう、状態の移行を定義するために使用されます。

- String：ICMP、TCP、または UDP プロトコルに基づいた正規表現文字列を検索します。String エンジンには、String ICMP、String TCP、および String UDP の 3 つがあります。
- String XL：ICMP、TCP、または UDP プロトコルに基づいた正規表現文字列を検索します。String XL エンジンは、正規表現アクセラレータ カードの最適な動作を実現します。String エンジンには、String ICMP XL、String TCP XL、および String UDP XL の 3 つがあります。

（注）　現時点で、String XL エンジンと正規表現アクセラレータ カードをサポートしているのは、Cisco ASA 5585-X のみです。

（注）　正規表現アクセラレータ カードは、標準の String エンジンと新規の String XL エンジンの両方に使用されます。ほとんどの標準 String エンジン シグニチャは、変更することなく、正規表現アクセラレータ カードによってコンパイルおよび分析できます。ただし、標準の String エンジン シグニチャを正規表現アクセラレータ カード用にコンパイルできない特別な状況もあります。そのような状況では、正規表現アクセラレータ カードでコンパイルする String XL エンジンの特定のパラメータを使用して、String XL エンジンに新規のシグニチャが作成されます。String XL エンジンの新規シグニチャは、標準の String エンジンにある元のシグニチャに代わるものです。

- Sweep：1 つのホスト（ICMP と TCP）、宛先ポート（TCP と UDP）、および 2 つのノード間で RPC 要求を送受信する複数のポートからのスイープを分析します。Sweep エンジンには、Sweep と Sweep Other TCP の 2 つがあります。
- Traffic Anomaly：ワームについて TCP、UDP、およびその他のトラフィックを検査します。
- Traffic ICMP：TFN2K、LOKI、DDOS などの非標準プロトコルを分析します。パラメータを設定できるのは 2 つのシグニチャだけです。
- Trojan：BO2K および TFN2K など、非標準プロトコルからのトラフィックを分析します。Trojan エンジンには、Bo2k、Tfn2k、および UDP の 3 つがあります。これらのエンジンには、ユーザが設定できるパラメータはありません。

詳細情報

シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Master エンジン

Master エンジンは、他のエンジンに構造体およびメソッドを提供し、コンフィギュレーションからの入力とアラート出力を処理します。ここでは、Master エンジンについて説明します。次のような構成になっています。

- 「一般的なパラメータ」（P.B-5）
- 「アラート頻度」（P.B-7）
- 「イベント アクション」（P.B-8）

一般的なパラメータ

次のパラメータは、Master エンジンの一部であり、すべてのシグニチャに適用されます（そのシグニチャ エンジンにとって意味がある場合）。

表 B-1 に、一般的な Master エンジンのパラメータを示します。

表 B-1　MASTER エンジンのパラメータ

パラメータ	説明	値
signature-id	このシグニチャの ID を指定します。	*number*
sub-signature-id	このシグニチャのサブ ID を指定します。	*number*
alert-severity	次に示すアラートの重大度を指定します。 • 危険アラート • 中間レベル アラート • 低レベル アラート • 情報アラート	• high • medium • low • informational（デフォルト）
sig-fidelity-rating	このシグニチャの忠実度レーティングを指定します。	0 ～ 100 （デフォルト = 100）
promisc-delta	アラートの重大度の決定に使用するデルタ値を指定します。	0 ～ 30 （デフォルト = 5）
sig-name	シグニチャの名前を指定します。	*sig-name*
alert-notes	アラート メッセージに含まれる、このシグニチャに関する追加情報を示します。	*alert-notes*
user-comments	このシグニチャに関するコメントを示します。	*comments*
alert-traits	このシグニチャについて文書化する特性を指定します。	0 ～ 65335
release	シグニチャが最後に更新されたリリースを示します。	*release*
signature-creation-date	シグニチャが作成された日付を指定します。	—
signature-type	シグニチャのカテゴリを指定します。	• anomaly • component • exploit • other vulnerability
engine	シグニチャが属するエンジンを指定します。 **（注）** エンジン固有のパラメータは、engine カテゴリの下に表示されます。	—
event-count	アラートを生成するまでのイベントの発生回数を指定します。	1 ～ 65535 （デフォルト = 1）

表 B-1　MASTER エンジンのパラメータ （続き）

パラメータ	説明	値
event-count-key	このシグニチャに関するイベントをカウントするストレージ タイプを指定します。 • 攻撃者のアドレス • 攻撃者と攻撃対象のアドレス • 攻撃者のアドレスと攻撃対象のポート • 攻撃対象のアドレス • 攻撃者と攻撃対象のアドレスおよびポート	• Axxx • AxBx • Axxb • xxBx • AaBb
specify-alert-interval {yes | no}	アラート間隔をイネーブルにします。 • alert-interval：イベント カウントがリセットされるまでの秒数を指定します。	2 ～ 1000
status	シグニチャが、イネーブルとディセーブルのどちらであるか、アクティブと廃棄のどちらであるかを指定します。	enabled | retired {yes | no}
obsoletes	新しいシグニチャによって古いシグニチャがディセーブルになったことを示します。	—
vulnerable-os-list	パッシブ OS フィンガープリントと組み合わせると、IPS は、所定の攻撃がターゲット システムに関連している可能性があるかどうかを判断できるようになります。	aix bsd general-os hp-ux ios irix linus mac-os netware other solaris unix windows windows-ut windows-nt-2k-xp
mars-category {yes | no}	MARS 攻撃カテゴリにシグニチャをマッピングします。[1]	—

1. これは、コンフィギュレーションに設定でき、アラートに表示できる静的情報カテゴリです。詳細については、MARS のマニュアルを参照してください。

Promiscuous Delta

無差別デルタによって、無差別モードでの特定のアラートのリスク レーティングが下がります。センサーはターゲット システムの属性を認識せず、無差別モードではパケットを拒否できないため、管理者がリスク レーティングの高いアラートの調査に集中できるよう、（リスク レーティングの低さに基づいて）無差別アラートの優先順位を下げるのに役立ちます。インライン モードでは、攻撃パケットがターゲット ホストに到達することのないよう、センサーがそれらのパケットを拒否できるので、ターゲットが脆弱であるかどうかは問題になりません。ネットワーク上で攻撃は拒否されるので、IPS はリスク レーティング値を下げません。サービス、OS、またはアプリケーションに固有ではないシグニチャは、無差別デルタが 0 になります。シグニチャが OS、サービス、またはアプリケーションに固有の場合は、カテゴリごとに 5 つのポイントで 5、10、または 15 の無差別デルタが計算されます。

注意　シグニチャの promisc-delta 設定は変更しないことを推奨します。

Obsoletes

シスコのシグニチャ チームでは、改善された最新のシグニチャに置き換えられた古い廃止シグニチャを示すため、また、エンジンでより適切なインスタンスを使用できる場合にそのエンジンでディセーブル化されたシグニチャを示すために、obsoletes フィールドを使用します。たとえば、現在、いくつかの String XL ハードウェア高速化シグニチャが、String エンジンに定義された同等のシグニチャに取って代わりました。

Vulnerable OS List

シグニチャの脆弱な OS の設定とパッシブ OS フィンガープリントを組み合わせると、IPS は、特定の攻撃がターゲット システムに関連している可能性があるかどうかを判断できます。攻撃が関連していることがわかると、表示されるアラートのリスク レーティング値が増大します。関連性が不明である場合は、通常、パッシブ OS フィンガープリント リストにエントリが存在しないため、リスク レーティングに対する変更は行われません。パッシブ OS フィンガープリントのエントリが存在し、シグニチャの脆弱な OS の設定に一致しない場合、リスク レーティング値は減少します。リスク レーティングは、デフォルトで +/- 10 ポイント単位で増減されます。

詳細情報

- 無差別モードの詳細については、「無差別モード」（P.5-3）を参照してください。
- パッシブ OS フィンガープリントの詳細については、「OS ID の設定」（P.8-26）を参照してください。

アラート頻度

アラート頻度のパラメータの目的は、stick などの IDS DoS ツールに対抗するために、イベント ストアに書き込まれるアラートの量を削減することです。fire-all、fire-once、summarize、および global-summarize という 4 つのモードがあります。サマリー モードは、現在のアラート量に応じて動的に変わります。たとえば、シグニチャを fire-all に設定できますが、一定のしきい値に達するとサマライズが開始されます。

表 B-2 に、アラート頻度のパラメータを示します。

表 B-2　Master エンジンのアラート頻度のパラメータ

パラメータ	説明	値
alert-frequency	アラートをグループ化するためのサマリー オプション。	—
summary-mode	サマライズに使用されるモード。	—
fire-all	すべてのイベントについてアラートを起動します。	—
fire-once	1 回だけアラートを起動します。	—
global-summarize	攻撃者や攻撃対象の数に関係なく 1 回だけアラートが起動されるようにアラートをサマライズします。	—
summarize	アラートをサマライズします。	—
summary-threshold	アラート数のしきい値。この値を超えるとシグニチャはサマリー モードに送られます。	0 ～ 65535
global-summary-threshold	イベント数のしきい値。この値を超えるとアラートはグローバル サマリーにサマライズされます。	1 ～ 65535

表 B-2　Master エンジンのアラート頻度のパラメータ （続き）

パラメータ	説明	値
summary-interval	各サマリー アラートで使用される時間（秒数）。	1 ～ 1000
summary-key	シグニチャをサマライズするストレージ タイプ：	Axxx
	• 攻撃者のアドレス	AxBx
	• 攻撃者と攻撃対象のアドレス	Axxb
	• 攻撃者のアドレスと攻撃対象のポート	xxBx
	• 攻撃対象のアドレス	AaBb
	• 攻撃者と攻撃対象のアドレスおよびポート	

イベント アクション

（注）　次のイベント アクションのほとんどは、その特定のエンジンに適していない場合を除き、各シグニチャ エンジンに属しています。

次のイベント アクション パラメータは、各シグニチャ エンジンに属しています（そのシグニチャ エンジンにとって意味がある場合）。

- アラートおよびログ アクション
  - produce-alert：evIdsAlert をイベント ストアに書き込みます。
  - produce-verbose-alert：evIdsAlert に、違反パケットの符号化ダンプ（切り捨てられている可能性あり）を組み込みます。
  - log-attacker-packets：攻撃者のアドレスを含むパケットの IP ロギングを開始し、アラートを送信します。
  - log-victim-packets：攻撃対象のアドレスを含むパケットの IP ロギングを開始し、アラートを送信します。
  - log-pair-packets：（インライン モードのみ）攻撃者のアドレスと攻撃対象のアドレスのペアを含むパケットの IP ロギングを開始します。
  - request-snmp-trap：SNMP 通知を実行する要求を NotificationApp に送信します。
- 拒否アクション
  - deny-packet-inline：（インライン モードのみ）このパケットを送信しません。

    （注）　deny-packet-inline に対するイベント アクション オーバーライドは保護されているため、削除できません。このオーバーライドを使用しない場合は、このエントリに対する override-item-status をディセーブルに設定してください。

  - deny-connection-inline：（インライン モードのみ）TCP フローでこのパケットおよび将来のパケットを送信しません。
  - deny-attacker-victim-pair-inline：（インライン モードのみ）指定された期間、この攻撃者と攻撃対象のアドレスのペアについては、このパケットおよび将来のパケットを送信しません。
  - deny-attacker-service-pair-inline：（インライン モードのみ）指定された期間、攻撃者のアドレスと攻撃対象のポートのペアについては、このパケットと将来のパケットを送信しません。

- deny-attacker-inline：（インライン モードのみ）指定された期間、この攻撃者のアドレスから発生したこのパケットおよび将来のパケットを送信しません。

（注）　これは最も厳しい拒否アクションです。単一の攻撃者アドレスからの現在および将来のパケットが拒否されます。各拒否アドレスは、拒否が開始された最初のイベントから *X* 秒でタイムアウトになります。X は、設定した秒数です。ネットワーク上でアドレスを再び許可する **clear denied-attackers** コマンドを使用して、拒否されたすべての攻撃者エントリをクリアできます。

- modify-packet-inline：（インライン モードのみ）エンド ポイントにおけるパケットの使用目的についてあいまいさを取り除くため、パケット データを変更します。

（注）　modify-packet-inline は、ノーマライザ エンジンの一部です。このパラメータは、パケットを停止し、不良チェックサム、範囲外の値、その他の RFC 違反などの不規則な問題を修正します。

- その他のアクション

（注）　IPv6 は、request-block-host、request-block-connection、または request-rate-limit の各イベント アクションをサポートしません。

  - request-block-connection：この接続をブロックするよう、ARC に要求します。
  - request-block-host：この攻撃者ホストをブロックするよう、ARC に要求します。
  - request-rate-limit：レート制限を実行するよう、ARC に要求します。
  - reset-tcp-connection：TCP リセットを送信して、TCP フローを乗っ取って終了します。

パケットのインライン拒否について

deny-packet-inline がアクションとして設定されているシグニチャの場合、または deny-packet-inline をアクションとして追加するイベント アクション オーバーライドの場合、次のアクションが実行される場合があります。

- droppedPacket
- deniedFlow
- tcpOneWayResetSent

パケットのインライン拒否アクションは、アラート内でドロップ パケット アクションとして表示されます。パケットのインライン拒否が TCP 接続に対して発生すると、自動的に接続のインライン拒否アクションにアップグレードされ、アラート内で拒否フローとして表示されます。IPS がパケットを 1 つだけ拒否しても、TCP は同じパケットの送信を繰り返し試みます。そのため、IPS で接続全体を拒否して、再送信が必ず失敗するようにします。

接続のインライン拒否が発生すると同時に、IPS は自動的に TCP 単方向リセットを送信します。これは、アラート内で、送信された TCP 単方向リセットとして表示されます。IPS は、接続を拒否するとき、開いている接続をクライアント（一般に攻撃者）とサーバ（一般に攻撃対象）の両方にそのまま残します。開いている接続が多くなりすぎると、攻撃対象にリソースの問題が発生する可能性があります。そのため、IPS は TCP リセットを攻撃対象に送信して、攻撃対象（通常はサーバ）側の接続を閉じます。これにより、攻撃対象のリソースが保護されます。さらに、フェールオーバーも阻止されます。それにより、接続が別のネットワーク パスにフェールオーバーして、攻撃対象に到達するようなことはなくなります。IPS は、攻撃者側を開いたままにし、攻撃者側からのすべてのトラフィックを拒否します。

正規表現の構文

正規表現（Regex）は、テキストを記述する手段として、強力で柔軟性のある表記言語です。パターンマッチングでは、正規表現によりあらゆる任意のパターンを簡潔に表記できます。

表 B-3 に、IPS シグニチャの正規表現の構文を示します。

表 B-3　シグニチャの正規表現の構文

メタ文字	名前	説明
?	疑問符	0 回または 1 回の繰り返し。
*	星印（アスタリスク）	0 回以上の繰り返し。
+	プラス	1 回以上の繰り返し。
{x}	量指定子	ちょうど *X* 回の繰り返し。
{x,}	最小量指定子	少なくとも *X* 回の繰り返し。
.	ドット	改行（0x0A）以外の任意の 1 文字。
[abc]	文字クラス	リスト内の任意の 1 文字。
[^abc]	否定文字クラス	リストにない任意の 1 文字。
[a-z]	文字範囲クラス	範囲内（両端も含む）の任意の 1 文字。
( )	カッコ	他のメタ文字の適用範囲を制限する際に使用する。
|	論理和（OR）	このメタ文字によって区切られている複数の表現のいずれかと一致します。
^	キャレット	行の先頭。
char	エスケープ文字。	*char* がメタ文字である場合も含めて、*char* そのものと一致する。
char	文字	char がメタ文字でない場合は、char そのものと一致する。
\r	復帰	復帰文字（0x0D）と一致する。
\n	改行	改行文字（0x0A）と一致する。
\t	Tab	タブ文字（0x09）と一致する。
\f	フォーム フィード	フォーム フィード文字（0x0C）と一致する。
\xNN	エスケープされた 16 進数文字	16 進コード 0xNN（0<=N<=F）を持つ文字と一致する。
\NNN	エスケープされた 8 進数文字	8 進コード NNN（0<=N<=8）を持つ文字と一致する。

繰り返し演算子ではいずれの場合も、該当する文字列のうち最も短いものが一致対象となります。一方、それ以外の演算子では、その適用範囲に最大限多くの文字が取り込まれるため、該当する文字列のうち最も長いものが一致対象となります。

表 B-4 は、正規表現のパターンの例を示したものです。

表 B-4　正規表現のパターン

一致対象	正規表現
Hacker	Hacker
Hacker または hacker	[Hh]acker
bananas、banananas、bananananananas など、一定の規則で構成されたすべての文字列	ba(na)+s
同じ行の中にある foo と bar の間に改行以外の文字が 0 個以上ある文字列	foo.*bar
foo または bar	foo|bar
moon または soon	(m|s)oon

AIC エンジン

Application Inspection and Control（AIC）エンジンは、HTTP Web トラフィックを検査し、FTP コマンドを施行します。ここでは、AIC エンジンとそのパラメータについて説明します。次のような構成になっています。

- 「AIC エンジンについて」（P.B-11）
- 「AIC エンジンとセンサーのパフォーマンス」（P.B-11）
- 「AIC エンジンのパラメータ」（P.B-12）

AIC エンジンについて

AIC は、Web トラフィックを徹底的に分析できます。HTTP セッションに対してより細かな制御を実行して、HTTP プロトコルの悪用を防ぎます。また、指定されたポート上でトンネルを作成しようとするインスタント メッセージングや gotomypc などのアプリケーションを管理制御できます。P2P およびインスタント メッセージングが HTTP で動作している場合は、それらのアプリケーションの検査チェックおよびポリシー チェックが可能です。AIC は、FTP トラフィックを検査し、実行されているコマンドを制御する方法も提供します。事前に定義されているシグニチャをイネーブルまたはディセーブルにしたり、カスタム シグニチャによってポリシーを作成したりできます。

（注） AIC エンジンは、HTTP トラフィックが AIC Web ポート上で受信されたときに動作します。トラフィックが Web トラフィックでも、AIC Web ポート上で受信されなければ、Service HTTP エンジンが実行されます。AIC 検査は、AIC Web ポートとして設定されたポートで、検査対象のトラフィックが HTTP トラフィックであれば、任意のポート上に存在できます。

AIC エンジンとセンサーのパフォーマンス

アプリケーション ポリシーの適用は、独自のセンサー機能です。AIC ポリシーの適用は、悪用、脆弱性、および異常を検査する従来の IPS テクノロジーをベースにしたものではなく、HTTP サービス ポリシーと FTP サービス ポリシーを適用するように設計されています。このポリシー適用に必要な検査動作は、従来の IPS 検査動作と比べて負荷が非常に高くなります。この機能の使用には、パフォーマンスの大幅な低下が伴います。AIC がイネーブルの場合、センサーの全体的な帯域幅キャパシティが減少します。

AIC ポリシーの適用は、IPS デフォルト設定ではディセーブルになっています。AIC ポリシーの適用を有効にする場合は、必要なポリシーだけを慎重に選択し、必要のないポリシーはディセーブルにすることを強く推奨します。また、センサーの検査負荷が最大容量に近くなっている場合は、センサーがオーバーサブスクライブされる可能性があるので、この機能の使用は推奨されません。このタイプのポリシー適用を処理する場合は、適応型セキュリティ アプライアンス ファイアウォールを使用することを推奨します。

AIC エンジンのパラメータ

AIC エンジンは、Web トラフィックのディープ インスペクション用のシグニチャを定義します。また、FTP コマンドを許可し、強制するシグニチャを定義します。AIC エンジンには、AIC HTTP と AIC FTP の 2 つがあります。

AIC エンジンには、次の機能があります。

- Web トラフィック：
  - RFC コンプライアンスの強制
  - HTTP 要求メソッドの許可と強制
  - 応答メッセージの検証
  - MIME タイプの強制
  - 転送符号化タイプの検証
  - 転送されるメッセージ コンテンツとデータ タイプに基づいたコンテンツ制御
  - URI の長さの強制
  - 設定されているポリシーとヘッダーに応じたメッセージ サイズの強制
  - トンネリング、P2P、およびインスタント メッセージングの強制

    この強制は、正規表現を使用して実行されます。事前定義されたシグニチャが存在しますが、リストを拡張できます。
- FTP トラフィック：
  - FTP コマンドの許可と強制

詳細情報

- AIC エンジン シグニチャの設定手順については、「AIC シグニチャの設定」（P.7-18）を参照してください。
- カスタム AIC シグニチャの例については、「AIC シグニチャの作成」（P.7-27）を参照してください。

Atomic エンジン

Atomic エンジンには、アラートの起動原因となる単純な単一パケットの条件に関するシグニチャが含まれます。ここでは、Atomic エンジンについて説明します。次のような構成になっています。

- 「Atomic ARP エンジン」（P.B-13）
- 「Atomic IP Advanced エンジン」（P.B-14）
- 「Atomic IP エンジン」（P.B-23）
- 「Atomic IPv6 エンジン」（P.B-26）

Atomic ARP エンジン

Atomic ARP エンジンは、レイヤ 2 の基本的な ARP シグニチャを定義し、ARP スプーフィング ツールである dsniff と ettercap に対して高度な検出を実行します。

表 B-5 に、Atomic ARP エンジンに固有のパラメータを示します。

表 B-5　Atomic ARP エンジンのパラメータ

パラメータ	説明	値
specify-arp-operation	（任意）ARP 動作をイネーブルにします。 • arp-operation：検査する ARP 動作のタイプ。	0 ～ 65535
specify-mac-flip	（任意）MAC アドレスの置換回数をイネーブルにします。 • mac-flip：アラート内の MAC アドレスを置換する回数を指定します。	0 ～ 65535
specify-request-inbalance	（任意）要求のアンバランスをイネーブルにします。 • request-inbalance：特定の IP アドレスに対して応答よりも要求の方が多い場合に、アラートを起動します。	0 ～ 65535
specify-type-of-arp-sig	（任意）ARP シグニチャのタイプをイネーブルにします。 • type-of-arp-sig：起動する ARP シグニチャのタイプを指定します。 – 宛先ブロードキャスト：255.255.255.255 の ARP 宛先アドレスを検出した場合に、このシグニチャのアラートを起動します。 – 同一の送信元と宛先：送信元と宛先の MAC アドレスが同じである ARP 宛先アドレスを検出した場合に、このシグニチャのアラートを起動します。 – 送信元ブロードキャスト（デフォルト）：255.255.255.255 の ARP 送信元アドレスを検出した場合に、このシグニチャのアラートを起動します。 – 送信元マルチキャスト：ARP 送信元 MAC アドレス 01:00:5e:（00-7f）を検出した場合に、このシグニチャのアラートを起動します。	dst-broadcast same-src-dst src-broadcast src-multicast
storage-key	固定データを保存するために使用するアドレス キーのタイプ。 • 攻撃者のアドレス • 攻撃者と攻撃対象のアドレス • 攻撃対象のアドレス • グローバル	Axxx AxBx xxBx xxxx

Atomic IP Advanced エンジン

ここでは、Atomic IP Advanced エンジンについて説明します。次のような構成になっています。

- 「Atomic IP Advanced エンジンの概要」（P.B-14）
- 「Atomic IP Advanced エンジンに関する制約事項」（P.B-15）
- 「Atomic IP Advanced エンジンのパラメータ」（P.B-15）

Atomic IP Advanced エンジンの概要

Atomic IP Advanced エンジンは、IPv6 ヘッダーとその拡張、IPv4 ヘッダーとそのオプション、ICMP、ICMPv6、TCP、および UDP の解析および解釈を行い、異常なアクティビティを探し出します。

Atomic IP Advanced エンジン シグニチャは、次のことを行います。

- 偽造されたアドレスなど、IP アドレスの異常を検査します。
- パケットの長さフィールドに不良情報が含まれていないか検査します。
- パケットに関する情報アラートを起動します。
- 限定された既知の脆弱性に対して、重大度の高いアラートを起動します。
- IPv6 にも適用できる Engine Atomic IP 内の IPv6 固有のシグニチャを複製します。
- パケット データからの IP アドレス、ポート、プロトコル、および制限付きの情報に基づいてトンネル化されたトラフィックを識別する、デフォルトのシグニチャを提供します。

一番外側の IP トンネルだけが識別されます。IPv6 トンネル、または IPv4 トンネル内の IPv6 トラフィックが検出されると、シグニチャはアラートを起動します。埋め込まれたトンネル内のその他の IPv6 トラフィックがすべて検査されるわけではありません。次のトンネリング メソッドがサポートされていますが、個別に検出されません。たとえば、ISATAP、6to4、および手動 IPv6 RFC 4213 トンネルはすべて IPv4 内の IPv6 として表示され、シグニチャ 1007 によって検出されます。

- ISATAP
- 6to4（RFC 3056）
- 手動で設定されたトンネル（RFC 4213）
- IPv6 over GRE
- UDP 内の Teredo（IPv6）
- MPLS（非暗号化）
- IPv6 over IPv6

IPv6 は次の動作をサポートしています。

- 送信元 IP アドレス、宛先 IP アドレス、または IP アドレス ペアによる拒否
- アラート
- TCP 接続のリセット
- ロギング

Atomic IP Advanced エンジンに関する制約事項

Atomic IP Advanced エンジンには、次の制限事項があります。

- レイヤ 4 識別子が最初のパケットに表示されないようにパケットがフラグメント化されている場合、パケットのレイヤ 4 フィールドは検出できません。
- フラグメントの再構成は存在しないため、IPv6 によってフラグメント化されたパケットのフローから、レイヤ 4 の攻撃は検出できません。
- トンネル化されたフローを使用した攻撃は検出できません。
- フラグメンテーション ヘッダーに対する確認は制限されています。
- IPv6 機能は、ASA 8.2(4) が導入されている場合に IPS SSP でサポートされます。
- 不正な重複ヘッダーが存在する場合、シグニチャは起動しますが、各ヘッダーを個別に検査することはできません。
- 異常検出は、IPv6 トラフィックをサポートしません。IPv4 トラフィックのみが異常検出プロセッサに送信されます。
- IPv6 トラフィックに対するレート制限およびブロッキングはサポートされていません。シグニチャにブロックまたはレート制限イベント アクションが設定されている場合、IPv6 トラフィックによってそのシグニチャがトリガーされると、アラートは生成されますがアクションは実行されません。

Atomic IP Advanced エンジンのパラメータ

（注）　範囲内の 2 番目の数は、最初の数以上である必要があります。

表 B-6 に、Atomic IP Advanced エンジンに固有のパラメータを示します。

表 B-6　Atomic IP Advanced エンジンのパラメータ

パラメータ	説明	値
グローバル		
fragment-status	フラグメントが必要かどうかを指定します。	any | no-fragments | want-fragments
specify-encapsulation	（任意）パケットの L3 の始まりの前にあるカプセル化を指定します。[1] • encapsulation：検査するカプセル化のタイプ。	none | mpls | gre | ipv4-in-ipv6 | ipip| any
specify-ip-version	（任意）IP プロトコル バージョンを指定します。 • version：IPv4 または IPv6。	ipv4 | ipv6
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]
Regex		
specify-regex-inspection	（任意）正規表現の検査をイネーブルにします。	[yes] | [no]

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
regex-scope	検索の開始ポイントと終了ポイントを指定します。	• ipv6-doh-only • ipv6-doh-plus • ipv6-hoh-only • ipv6-hoh-plus • ipv6-rh-only • ipv6-rh-plus • layer3-only • layer3-plus • layer4
regex-string	単一の TCP パケット内で検索する正規表現を指定します。	string
specify-exact-match-offset	完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために regex-string が報告する必要のある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	最小一致長をイネーブルにします。 • min-match-length：regex-string が一致する必要のある最小バイト数を指定します。	0 ～ 65535
specify-min-match-offset	最小一致オフセットをイネーブルにします。 • min-match-offset：一致を有効にするために regex-string が報告する必要のある最小ストリーム オフセットを指定します。	0 ～ 65535
specify-max-match-offset	最大一致オフセットをイネーブルにします。 • max-match-offset：一致を有効にするために regex-string が報告する必要のある最大ストリーム オフセットを指定します。	0 ～ 65535
IPv6		
specify-authentication-header	（任意）認証ヘッダーの検査をイネーブルにします。 • ah-present：認証ヘッダーが存在することを指定します。 – ah-length：認証ヘッダーの長さを指定します。 – ah-next-header：認証ヘッダーの値を指定します。	have-ah | no-ah 0 ～ 1028 0 ～ 255

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-dest-options-header	（任意）宛先オプション ヘッダーの検査をイネーブルにします。 • doh-present：宛先オプション ヘッダーが存在することを指定します。 – doh-count：検査する宛先オプション ヘッダーの数を指定します。 – doh-length：検査する宛先オプション ヘッダーの長さを指定します。 – doh-next-header：検査する次の宛先オプション ヘッダーの数を指定します。 – doh-option-type：検査する宛先オプション ヘッダーのタイプを指定します。 – doh-option-length：検査する宛先オプション ヘッダーの長さを指定します。	have-doh | no-doh 0 ～ 2 8 ～ 2048 0 ～ 255 0 ～ 255 0 ～ 255
specify-esp-header	（任意）ESP ヘッダーの検査をイネーブルにします。 • esp-present ：ESP ヘッダーが存在することを指定します。	have-esp | no-esp
specify-first-next-header	（任意）最初のネクスト ヘッダーの検査をイネーブルにします。 • first-next-header : 検査する最初のネクスト ヘッダーの値を指定します。	0 ～ 255
specify-flow-label	（任意）フロー ラベルの検査をイネーブルにします。 • flow-label：検査するフロー ラベルの値を指定します。	0 ～ 1048575
specify-headers-out-of-order	（任意）順序が不正なヘッダーの検査をイネーブルにします。 • headers-out-of-order：検査するヘッダー順序を指定します。	[true] | [false]
specify-headers-repeated	（任意）繰り返されたヘッダーの検査をイネーブルにします。 • headers-repeated：検査するヘッダーの繰り返しを指定します。	[true] | [false]
specify-hop-limit	（任意）ホップ制限をイネーブルにします。 • hop-limit：検査するホップ制限の値を指定します。	0 ～ 255
specify-hop-options-header	（任意）ホップバイホップ オプション ヘッダーの検査をイネーブルにします。 • hoh-present : ホップバイホップ オプション ヘッダーが存在することを指定します。	have-hoh | no-hoh

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-ipv6-addr-options	（任意）IPv6 アドレス オプションをイネーブルにします。 • ipv6-addr-options : IPv6 アドレス オプションを指定します。 – address-with-localhostt : ::1 を持つ IP アドレス。 – documentation-address : 2001:db8::/32 プレフィクスを持つ IP アドレス。 – ipv6-addr : IP アドレス。 – link-local-address : IPv6 リンク ローカル アドレスを検査します。 – multicast-dst : 宛先マルチキャスト アドレスを検査します。 – multicast-src : 送信元マルチキャスト アドレスを検査します。 – not-link-local-address : リンクローカルではないアドレスを検査します。 – not-valid-address : リンクローカル、グローバル、またはマルチキャストに予約されていないアドレスを検査します。 – src-ip-eq-dst-ip : 送信元アドレスと宛先アドレスは同じです。	
specify-ipv6-data-length	（任意）IPv6 データ長の検査をイネーブルにします。 • ipv6-data-length : 検査する IPv6 データ長を指定します。	0 ～ 65535
specify-ipv6-header-length	（任意）IPv6 ヘッダー長の検査をイネーブルにします。 • ipv6-header-length : 検査する IPv6 ヘッダー長を指定します。	0 ～ 65535
specify-ipv6-total-length	（任意）IPv6 の合計長の検査をイネーブルにします。 • ipv6-total-length : 検査する IPv6 の合計長を指定します。	0 ～ 65535
specify-ipv6-payload-length	（任意）IPv6 ペイロード長の検査をイネーブルにします。 • ipv6-payload-length : 検査する IPv6 ペイロード長を指定します。	0 ～ 65535
specify-routing-header	（任意）ルーティング ヘッダーの検査をイネーブルにします。 • rh-present : ルーティング ヘッダーが存在することを指定します。	have-rh | no-rh

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-traffic-class	（任意）トラフィック クラスの検査をイネーブルにします。 • traffic-class：検査するトラフィック クラスの値を指定します。	0 ～ 255
IPv4		
specify-ip-addr-options	（任意）IP アドレス オプションをイネーブルにします。 • ip-addr-options：IP アドレス オプションを指定します。	address-with-localhost ip-addr rfc-1918-address src-ip-eq-dst-ip
specify-ip-header-length	（任意）IP ヘッダー長の検査をイネーブルにします。 • ip-header-length 検査する IP ヘッダー長を指定します。	0 ～ 16
specify-ip-id	（任意）IP 識別子の検査をイネーブルにします。 • ip-id：検査する IP ID を指定します。	0 ～ 255
specify-ip-option-inspection	（任意）IP オプションの検査をイネーブルにします。 • ip-option-inspection：IP オプションの値を指定します。 – ip-option：照合する IP OPTION コード。 – ip-option-abnormal：オプションのリストに誤りがあります。	0 ～ 65535 [true] | [false]
specify-ip-payload-length	（任意）IP ペイロード長の検査をイネーブルにします。 • ip-payload-length：検査する IP ペイロード長を指定します。	0 ～ 65535
specify-ip-tos	（任意）サービスの IP タイプを指定します。 • ip-tos：検査するサービスの IP タイプを指定します。	0 ～ 255
specify-ip-total-length	（任意）IP 合計長の検査をイネーブルにします。 • ip-total-length：検査する IP パケットの合計長を指定します。	0 ～ 65535
specify-ip-ttl	（任意）IP 存続可能時間の検査をイネーブルにします。 • ip-ttl：IP TTL の検査を指定します。	0 ～ 255
specify-ip-version	（任意）IP バージョンの検査をイネーブルにします。 • ip-version：検査する IP バージョンを指定します。	0 ～ 16

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
L4 プロトコル		
specify-l4-protocol	（任意）L4 プロトコルの検査をイネーブルにします。 • l4-protocol：検査する L4 プロトコルを指定します。	icmp icmpv6 tcp udp other
その他の L4 プロトコル		
other-ip-protocol-id	（任意）その他の L4 プロトコルの検査をイネーブルにします。 • other-ip-protocol-id：アラートを送信する単一 IP プロトコル番号を指定します。	0 ～ 256
L4 プロトコル ICMP		
specify-icmp-code	（任意）L4 ICMP コードの検査をイネーブルにします。 • icmp-code：ICMP ヘッダーの CODE 値を指定します。	0 ～ 65535
specify-icmp-id	（任意）L4 ICMP ID の検査をイネーブルにします。 • icmp-id：ICMP ヘッダーの IDENTIFIER 値を指定します。	0 ～ 65535
specify-icmp-seq	（任意）L4 ICMP シーケンスの検査をイネーブルにします。 • icmp-seq：検査する ICMP シーケンスを指定します。	0 ～ 65535
specify-icmp-type	（任意）ICMP ヘッダー タイプの検査をイネーブルにします。 • icmp-type：ICMP ヘッダーの TYPE 値を指定します。	0 ～ 65535
L4 プロトコル ICMPv6		
specify-icmpv6-code	（任意）L4 ICMPv6 コードの検査をイネーブルにします。 • icmpv6-code：ICMPv6 ヘッダーの CODE 値を指定します。	0 ～ 255
specify-icmpv6-id	（任意）L4 ICMPv6 識別子の検査をイネーブルにします。 • icmpv6-id：ICMPv6 ヘッダーの IDENTIFIER 値を指定します。	0 ～ 65535
specify-icmpv6-length	（任意）L4 ICMPv6 長の検査をイネーブルにします。 • icmpv6-length：ICMPv6 ヘッダーの LENGTH 値。	0 ～ 65535

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-icmpv6-mtu-field	（任意）L4 ICMPv6 MTU フィールドの検査をイネーブルにします。 • icmpv6-mtu-field：ICMPv6 ヘッダーの MTU フィールド値。	4,294,967,295
specify-icmpv6-option-type	（任意）L4 ICMPv6 タイプの検査をイネーブルにします。 • icmpv6-option-type：検査する ICMPv6 オプション タイプを指定します	0 ～ 255
icmpv6-option-length	（任意）L4 ICMPv6 オプション タイプの検査をイネーブルにします。 • icmpv6-option-length：検査する ICMPv6 オプション タイプを指定します	0 ～ 255
specify-icmpv6-seq	（任意）L4 ICMPv6 シーケンスの検査をイネーブルにします。 • icmpv6-seq：ICMPv6 ヘッダーの SEQUENCE 値。	0 ～ 65535
specify-icmpv6-type	（任意）L4 ICMPv6 タイプの検査をイネーブルにします。 • icmpv6-type : ICMPv6 ヘッダーの TYPE 値。	0 ～ 255
L4 プロトコル TCP および UDP		
specify-dst-port	（任意）使用する宛先ポートをイネーブルにします。 • dst-port：このシグニチャの該当宛先ポート。	0 ～ 65535
specify-src-port	（任意）使用する送信元ポートをイネーブルにします。 • src-port : このシグニチャの該当送信元ポート。	0 ～ 65535
specify-tcp-mask	（任意）使用する TCP マスクをイネーブルにします。 • tcp-mask：TCP フラグの比較に使用するマスク。 – URG ビット – ACK ビット – PSH ビット – RST ビット – SYN ビット – FIN ビット	• urg • ack • psh • rst • syn • fin

表 B-6　Atomic IP Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-tcp-flags	（任意）使用する TCP フラグをイネーブルにします。 • tcp-flags：マスクによってマスクされた場合に照合する TCP フラグ。 – URG ビット – ACK ビット – PSH ビット – RST ビット – SYN ビット – FIN ビット	• urg • ack • psh • rst • syn • fin
specify-tcp-reserved	（任意）使用のために予約されている TCP をイネーブルにします。 • tcp-reserved：予約されている TCP。	0 ～ 63
specify-tcp-header-length	（任意）L4 TCP ヘッダー長の検査をイネーブルにします。 • tcp-header-length：検査に使用する TCP ヘッダーの長さを指定します。	0 ～ 60
specify-tcp-payload-length	（任意）L4 TCP ペイロード長の検査をイネーブルにします。 • tcp-payload-length：TCP ペイロードの長さを指定します。	0 ～ 65535
specify-tcp-urg-pointer	（任意）L4 TCP URG ポインタの検査をイネーブルにします。 • tcp-urg-pointer：TCP URG フラグ検査を指定します。	0 ～ 65535
specify-tcp-window-size	（任意）L4 TCP ウィンドウ サイズの検査をイネーブルにします。 • tcp-window-size：TCP パケットのウィンドウ サイズを指定します。	0 ～ 65535
specify-udp-valid-length	（任意）L4 UDP の有効な長さの検査をイネーブルにします。 • udp-valid-length：有効と見なされ、検査の必要のない UDP パケット長を指定します。	0 ～ 65535
specify-udp-length-mismatch	（任意）L4 UDP 長の不一致の検査をイネーブルにします。 • udp-length-mismatch：IP データ長が UDP ヘッダー長よりも短い場合はアラートを起動します。	[true] | [false]

1. パケットが GRE、IPIP、IPv4inIPv6、または MPL の場合、センサーは L3 カプセル化ヘッダーおよびカプセル化ヘッダーをスキップし、すべての検査は 2 番目の L3 から開始されます。カプセル化の列挙型によって、エンジンは当該の L3 の前にカプセル化ヘッダーが存在するかどうかを確認できます。

詳細情報

- カスタム IPv6 シグニチャの例については、「Atomic IP Advanced エンジン シグニチャの例」（P.7-53）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Atomic IP エンジン

Atomic IP エンジンは、IP プロトコル ヘッダーと、関連するレイヤ 4 トランスポート プロトコル（TCP、UDP、および ICMP）およびペイロードを検査するシグニチャを定義します。Atomic エンジンでは、複数のパケットにまたがる固定データは保存されません。その代わりに、1 つのパケットの分析を基にしてアラームを起動できます。

表 B-7 に、Atomic IP エンジンに固有のパラメータを示します。

表 B-7　Atomic IP エンジンのパラメータ

パラメータ	説明	値
specify-ip-addr-options	（任意）IP アドレス オプションをイネーブルにします。 • ip-addr-options：IP アドレス オプションを指定します。	address-with-localhost ip-addr rfc-1918-address src-ip-eq-dst-ip
specify-ip-header-length	（任意）IP ヘッダー長の検査をイネーブルにします。 • ip-header-length：検査する IP ヘッダー長を指定します。	0 ～ 16
specify-ip-id	（任意）IP 識別子の検査をイネーブルにします。 • ip-id：検査する IP ID を指定します。	0 ～ 255
specify-ip-option-inspection	（任意）IP オプションの検査をイネーブルにします。 • ip-option-inspection：IP オプションの値を指定します。 – ip-option：照合する IP OPTION コード。 – ip-option-abnormal：オプションのリストに誤りがあります。	0 ～ 65535 [true] | [false]
specify-ip-payload-length	（任意）IP ペイロード長の検査をイネーブルにします。 • ip-payload-length：検査する IP ペイロード長を指定します。	0 ～ 65535
specify-ip-tos	（任意）サービスの IP タイプを指定します。 • ip-tos：検査するサービスの IP タイプを指定します。	0 ～ 6 255

表 B-7　Atomic IP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-ip-total-length	（任意）IP 合計長の検査をイネーブルにします。 • ip-total-length：検査する IP パケットの合計長を指定します。	0 ～ 65535
specify-ip-ttl	（任意）IP 存続可能時間の検査をイネーブルにします。 • ip-ttl：IP TTL の検査を指定します。	0 ～ 255
specify-ip-version	（任意）IP バージョンの検査をイネーブルにします。 • ip-version：検査する IP バージョンを指定します。	0 ～ 16
specify-l4-protocol	（任意）L4 プロトコルの検査をイネーブルにします。 • l4-protocol：検査する L4 プロトコルを指定します。	icmp tcp udp other-protocol
specify-icmp-code	（任意）L4 ICMP コードの検査をイネーブルにします。 • icmp-code：ICMP ヘッダーの CODE 値を指定します。	0 ～ 65535
specify-icmp-id	（任意）L4 ICMP ID の検査をイネーブルにします。 • icmp-id：ICMP ヘッダーの IDENTIFIER 値を指定します。	0 ～ 65535
specify-icmp-seq	（任意）L4 ICMP シーケンスの検査をイネーブルにします。 • icmp-seq：検査する ICMP シーケンスを指定します。	0 ～ 65535
specify-icmp-type	（任意）ICMP ヘッダー タイプの検査をイネーブルにします。 • icmp-type：ICMP ヘッダーの TYPE 値を指定します。	0 ～ 65535
specify-icmp-total-length	（任意）L4 ICMP 合計ヘッダー長の検査をイネーブルにします。 • icmp-total-length：検査する ICMP 合計長の値を指定します。	0 ～ 65535
other-ip-protocol-id	（任意）その他の L4 プロトコルの検査をイネーブルにします。 • other-ip-protocol-id：アラートを送信する単一 IP プロトコル番号を指定します。	0 ～ 256

表 B-7　Atomic IP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-dst-port	（任意）使用する宛先ポートをイネーブルにします。 • dst-port：このシグニチャの該当宛先ポート。	0 ～ 65535
specify-src-port	（任意）使用する送信元ポートをイネーブルにします。 • src-port：このシグニチャの該当送信元ポート。	0 ～ 65535
specify-tcp-mask	（任意）使用する TCP マスクをイネーブルにします。 • tcp-mask：TCP フラグの比較に使用するマスク。 – URG ビット – ACK ビット – PSH ビット – RST ビット – SYN ビット – FIN ビット	• urg • ack • psh • rst • syn • fin
specify-tcp-flags	（任意）使用する TCP フラグをイネーブルにします。 • tcp-flags：マスクによってマスクされた場合に照合する TCP フラグ。 – URG ビット – ACK ビット – PSH ビット – RST ビット – SYN ビット – FIN ビット	• urg • ack • psh • rst • syn • fin
specify-tcp-reserved	（任意）使用のために予約されている TCP をイネーブルにします。 • tcp-reserved：予約されている TCP。	0 ～ 63
specify-tcp-header-length	（任意）L4 TCP ヘッダー長の検査をイネーブルにします。 • tcp-header-length：検査に使用する TCP ヘッダーの長さを指定します。	0 ～ 60

表 B-7　Atomic IP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-tcp-payload-length	（任意）L4 TCP ペイロード長の検査をイネーブルにします。 • tcp-payload-length：TCP ペイロードの長さを指定します。	0 ～ 65535
specify-tcp-urg-pointer	（任意）L4 TCP URG ポインタの検査をイネーブルにします。 • tcp-urg-pointer：TCP URG フラグ検査を指定します。	0 ～ 65535
specify-tcp-window-size	（任意）L4 TCP ウィンドウ サイズの検査をイネーブルにします。 • tcp-window-size：TCP パケットのウィンドウ サイズを指定します。	0 ～ 65535
specify-udp-length	（任意）L4 UDP 長の検査をイネーブルにします。 • udp-length：IP データ長が UDP ヘッダー長よりも短い場合はアラートを起動します。	0 ～ 65535
specify-udp-valid-length	（任意）L4 UDP の有効な長さの検査をイネーブルにします。 • udp-valid-length：有効と見なされ、検査の必要のない UDP パケット長を指定します。	0 ～ 65535
specify-udp-length-mismatch	（任意）L4 UDP 長の不一致の検査をイネーブルにします。 • udp-length-mismatch：IP データ長が UDP ヘッダー長よりも短い場合はアラートを起動します。	[true] | [false]

Atomic IPv6 エンジン

Atomic IPv6 エンジンは、異常のある IPv6 トラフィックによって誘導される 2 つの IOS 脆弱性を検出します。これらの脆弱性は、ルータ クラッシュやその他のセキュリティ問題を引き起こすおそれがあります。IOS 脆弱性の 1 つは、最初のフラグメントを複数処理し、バッファ オーバーフローを引き起こします。もう 1 つは、不正な ICMPv6 ネイバー探索オプションを処理し、同様にバッファ オーバーフローを引き起こします。

（注）　IPv6 は、IP アドレスを 32 ビットから 128 ビットに拡大し、より多くのアドレッシング階層レベルとアドレス指定可能ノード、およびアドレスの自動設定をサポートします。

8 つの Atomic IPv6 シグニチャがあります。Atomic IPv6 は、次のタイプのネイバー探索プロトコルを検査します。

- タイプ 133：ルータ送信要求
- タイプ 134：ルータ アドバタイズメント

- タイプ 135：ネイバー送信要求
- タイプ 136：ネイバー アドバタイズメント
- タイプ 137：リダイレクト

（注）　ホストおよびルータはネイバー探索を使用して、添付されたリンクに常駐し、無効になったキャッシュ値を素早くパージすることがわかっているネイバーのリンク層アドレスを判断します。また、ホストはネイバー探索を使用して、ホストに代わってパケットを転送する隣接ルータを検出します。

各ネイバー探索タイプには、1 つまたは複数のネイバー探索オプションを設定することができます。Atomic IPv6 エンジンは、各オプションの長さに対して RFC 2461 に明記されている有効値への準拠性を検査します。オプションの長さに違反があると、その長さが検出されたオプション タイプに対応するアラートが生成されます（シグニチャ 1601 ～ 1605）。

（注）　Atomic IPv6 シグニチャには、特に設定が必要なパラメータはありません。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Fixed エンジン

Fixed エンジンは、複数の正規表現パターンを 1 つのパターン マッチング テーブルにまとめ、データ内の検索を一度に実行できるようにします。このエンジンは、ICMP、TCP、および UDP プロトコルをサポートします。最小の検査深さに達すると（1 ～ 100 バイト）、検査は停止します。Fixed エンジンには、Fixed ICMP、Fixed TCP、および Fixed UDP の 3 つがあります。

（注）　Fixed TCP および Fixed UDP は、除外ポートとして service-ports パラメータを使用します。Fixed ICMP は、除外される ICMP タイプとして service-ports パラメータを使用します。

表 B-8 に、Fixed ICMP エンジンに固有のパラメータを示します。

表 B-8　Fixed ICMP エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
max-payload-inspect-length	シグニチャの最大検査深さを指定します。	1 ～ 250
regex-string	単一のパケット内で検索する正規表現を指定します。	string

表 B-8　Fixed ICMP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために regex-string が報告する必要のある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：regex-string が一致する必要のある最小バイト数を指定します。	0 ～ 65535
specify-icmp-type	（任意）ICMP ヘッダー タイプの検査をイネーブルにします。 • icmp-type：ICMP ヘッダーの TYPE 値を指定します。	0 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

表 B-9 に、Fixed TCP エンジンに固有のパラメータを示します。

表 B-9　Fixed TCP エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
max-payload-inspect-length	シグニチャの最大検査深さを指定します。	1 ～ 250
regex-string	単一のパケット内で検索する正規表現を指定します。	string
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために regex-string が報告する必要のある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：regex-string が一致する必要のある最小バイト数を指定します。	0 ～ 65535

表 B-9　Fixed TCP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-service-ports	使用するサービス ポートをイネーブルにします。 • service-ports：ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

表 B-10 に、Fixed UDP エンジンに固有のパラメータを示します。

表 B-10　Fixed UDP エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
max-payload-inspect-length	シグニチャの最大検査深さを指定します。	1 ～ 250
regex-string	単一のパケット内で検索する正規表現を指定します。	string
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために regex-string が報告する必要のある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：regex-string が一致する必要のある最小バイト数を指定します。	0 ～ 65535
specify-service-ports	使用するサービス ポートをイネーブルにします。 • service-ports：ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Flood エンジン

Flood エンジンは、複数のパケットを単一のホストまたはネットワークに送信しているホストまたはネットワークをモニタするシグニチャを定義します。たとえば、攻撃対象ホストに送信される（特定のタイプの）パケットが 1 秒あたり 150 以上検出されると起動するシグニチャを作成できます。Flood エンジンには、Flood Host と Flood Net の 2 つのタイプがあります。

表 B-11 に、Flood Host エンジンに固有のパラメータを示します。

表 B-11　Flood Host エンジンのパラメータ

パラメータ	説明	値
protocol	検査するトラフィックの種類。	ICMP UDP
rate	1 秒あたりのパケット数のしきい値。	0 ～ 65535[1]
icmp-type	ICMP ヘッダー タイプの値を指定します。	0 ～ 65535
dst-ports	UDP プロトコルを選択した場合の宛先ポートを指定します。	0 ～ 65535[2] a-b[,c-d]
src-ports	UDP プロトコルを選択する場合の送信元ポートを指定します。	0 ～ 65535[3] a-b[,c-d]

1. レートが 1 秒あたりのパケット数よりも大きい場合はアラートが起動します。
2. 範囲の 2 番目の数は、最初の数以上である必要があります。
3. 範囲の 2 番目の数は、最初の数以上である必要があります。

表 B-12 に、Flood Net エンジンに固有のパラメータを示します。

表 B-12　Flood Net エンジンのパラメータ

パラメータ	説明	値
gap	フラッディング シグニチャに許可された時間の差（秒単位）。	0 ～ 65535
peaks	許可されたフラッディング トラフィックのピークの数。	0 ～ 65535
protocol	検査するトラフィックの種類。	ICMP TCP UDP
rate	1 秒あたりのパケット数のしきい値。	0 ～ 65535[1]
sampling-interval	トラフィックをサンプリングする間隔。	1 ～ 3600
icmp-type	ICMP ヘッダー タイプの値を指定します。	0 ～ 65535

1. レートが 1 秒あたりのパケット数よりも大きい場合はアラートが起動します。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Meta エンジン

注意　Meta エンジン シグニチャを大量に使用すると、全体的なセンサー パフォーマンスに悪影響を与える可能性があります。

Meta エンジンでは、スライディング時間間隔内に、関連した方法で発生するイベントを定義します。このエンジンは、パケットではなくイベントを処理します。シグネチャ イベントが生成されると、Meta エンジンはシグネチャ イベントを検査して、1 つ以上の Meta 定義に一致するかどうかを判定します。Meta エンジンは、すべてのイベント要件が満たされるとシグネチャ イベントを生成します。

すべてのシグネチャ イベントは、シグニチャ イベント アクション プロセッサによって Meta エンジンに渡されます。シグニチャ イベント アクション プロセッサは、最小ヒット数オプションを処理してからイベントを渡します。Meta エンジンがコンポーネント イベントを処理してから、サマライズおよびイベント アクションは処理されます。

表 B-13 に、Meta エンジンに固有のパラメータを示します。

表 B-13　Meta エンジン パラメータ

パラメータ	説明	値
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]
meta-reset-interval	Meta シグニチャをリセットする時間（秒単位）。	0 ～ 3600
component-list	Meta コンポーネントのリスト。 • edit：既存のエントリを編集します。 • insert：新しいエントリをリストに挿入します。 – begin：エントリをアクティブ リストの先頭に配置します。 – end：エントリをアクティブ リストの末尾に配置します。 – inactive：エントリを非アクティブ リストに配置します。 – before:エントリを指定エントリの前に配置します。 – after：エントリを指定エントリの後に配置します。 • move：リスト内のエントリを移動します。	*name1*

表 B-13　Meta エンジン パラメータ （続き）

パラメータ	説明	値
meta-key	Meta シグニチャのストレージ タイプ。 • 攻撃者のアドレス • 攻撃者と攻撃対象のアドレス • 攻撃者と攻撃対象のアドレスおよびポート • 攻撃対象のアドレス	AaBb AxBx Axxx xxBx
unique-victim-ports	Meta シグニチャごとに一意の必須攻撃対象ポートの番号。	1 ～ 256
component-list-in-order	コンポーネント リストが順番に起動するかどうか。	[true] | [false]

詳細情報

- カスタム Meta エンジン シグニチャの例については、「Meta エンジン シグニチャの例」（P.7-49）を参照してください。
- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Multi String エンジン

注意　Multi String エンジンは、メモリの使用状況に大きく影響することがあります。

Multi String エンジンでは、レイヤ 4 トランスポート プロトコル（ICMP、TCP、および UDP）のペイロードを検査するシグニチャを定義します。この検査は、1 つのシグニチャに対して複数の文字列を照合して行います。シグニチャを起動するために一致する必要がある一連の正規表現パターンを指定できます。たとえば、UDP サービスで regex 1 とそれに続く regex 2 を検索するシグニチャを定義できます。UDP および TCP の場合は、ポート番号と方向を指定できます。単一の送信元ポート、単一の宛先ポート、または両方のポートを指定できます。文字列の照合は両方向で実行されます。

複数の正規表現パターンを指定する必要がある場合は、Multi String エンジンを使用してください。それ以外の場合は、String ICMP、String TCP、または String UDP エンジンを使用して、これらのプロトコルのいずれかに対応した単一の正規表現パターンを指定できます。

表 B-14 に、Multi String エンジンに固有のパラメータを示します。

表 B-14　Multi String エンジンのパラメータ

パラメータ	説明	値
inspect-length	起動するシグニチャに対して違反するすべての文字列を含める必要があるストリームまたはパケットの長さ。	0 ～ 4294967295
protocol	レイヤ 4 プロトコルの選択。	icmp tcp udp

表 B-14　Multi String エンジンのパラメータ （続き）

パラメータ	説明	値
regex-component	regex コンポーネントのリスト。 • regex-string：検索する文字列。 • spacing-type：前回の一致箇所から、またはそれが一致結果からリストの最初のエントリの場合はストリームまたはパケットの先頭から空ける必要のある間隔のタイプ。	list（1 ～ 16 項目） exact minimum
port-selection	検査する TCP または UDP ポートのタイプ。 • both-ports：送信元ポートと宛先ポートの両方を指定します。 • dest-ports：宛先ポートの範囲を指定します。 • source-ports：送信元ポートの範囲を指定します。[1]	0 ～ 65535 [2]
exact-spacing	この正規表現文字列と直前の正規表現文字列との間、またはそれがリスト内の最初のエントリである場合にはストリームまたはパケットの先頭から空ける必要のある正確なバイト数。	0 ～ 4294967296
min-spacing	この正規表現文字列と直前の正規表現文字列との間、またはストリームやパケットの先頭から（リスト内の最初のエントリである場合）、空ける必要のある最小バイト数。	0 ～ 4294967296
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. ポートの照合は、クライアントからサーバへ、およびサーバからクライアントへ向かうトラフィック フローの両方の方向で実行されます。たとえば、送信元ポート値が 80 である場合、クライアントからサーバへのトラフィック フローの方向では、クライアント ポートがポート 80 の場合に検査が実行されます。サーバからクライアントへのトラフィック フローの方向では、サーバ ポートがポート 80 の場合に検査が実行されます。
2. 有効な値は、a-b[,c-d] 形式で指定された 0 ～ 65535 の範囲内の整数から成るカンマ区切りのリストです。範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

ノーマライザ エンジン

ノーマライザ エンジンは、IP フラグメンテーションと TCP 正規化を処理します。ここでは、ノーマライザ エンジンについて説明します。次のような構成になっています。

- 「ノーマライザ エンジンの概要」（P.B-34）
- 「ノーマライザ エンジンのパラメータ」（P.B-35）

ノーマライザ エンジンの概要

（注）　ノーマライザ エンジンには、カスタム シグニチャは追加できません。既存のシグニチャの調整は可能です。

ノーマライザ エンジンは、IP フラグメント再構成と TCP ストリーム再構成を処理します。ノーマライザ エンジンを使用すると、センサーが同時に追跡しようとするフラグメントの最大数など、システム リソースの使用に対する制限を設定できます。無差別モードのセンサーは、違反に対するアラートを報告します。インライン モードのセンサーは、produce-alert、deny-packet-inline、および modify-packet-inline など、イベント アクション パラメータで指定されたアクションを実行します。

注意　シグニチャ 3050 Half Open SYN Attack でアクションとして modify-packet-inline を選択している場合、保護がアクティブである間は 20 ～ 30 % ほどパフォーマンスが低下する可能性があります。保護は、実際の SYN フラッドの間だけアクティブになります。

IP フラグメンテーションの正規化

IP データグラムの意図した、または意図しないフラグメンテーションによってエクスプロイトが隠れる可能性があるので、検出が困難または不可能になります。フラグメンテーションは、ファイアウォールやルータにあるようなアクセス コントロール ポリシーを回避するために使用される場合もあります。また、さまざまなオペレーティング システムがさまざまな方法を使用して、フラグメント化されたデータグラムをキューに入れたりディスパッチしたりします。センサーがエンド ホストでデータグラムを再構成するために考えられるすべての方法を確認する必要がある場合は、センサーが DoS 攻撃を受ける可能性があります。フラグメント化されたデータグラムをすべてインラインで再構成し、完全なデータグラムだけを転送し、必要に応じてそのデータグラムを再度フラグメント化すれば、これを防ぐことができます。IP フラグメンテーションの正規化の装置は、この機能を実行します。

TCP の正規化

意図的または意図しない TCP セッション セグメンテーションによって、いくつかの攻撃クラスが隠されることがあります。ポリシーが偽陽性や偽陰性なしに実施されるようにするには、2 つの TCP エンドポイントの状態が追跡され、実際のホスト エンドポイントによって処理されたデータだけが渡される必要があります。TCP ストリームで重複が発生する可能性がありますが、TCP セグメントの再転送以外は、非常にまれです。TCP セッションでの上書きは発生してはならないものです。上書きが発生する場合は、誰かがセキュリティ ポリシーを意図的に回避しようとしているか、TCP スタックの実装が壊れています。センサーが TCP プロキシとして動作していない限り、両方のエンドポイントの状態について完全な情報を保持することはできません。センターが TCP プロキシとして動作する代わりに、セグメントが適切に並べられ、ノーマライザ エンジンによって回避および攻撃に関連する異常なパケットが検索されます。

IPv6 フラグメント

ノーマライザ エンジンは IPv6 フラグメントを再構成し、その他のエンジンやプロセッサによる検査およびアクション用に再構成されたバッファを転送できます。IPv4 と IPv6 の相違点は、次のとおりです。

- ノーマライザ エンジン シグニチャの modify-packet-inline は、IPv6 データグラムには効果がありません。
- シグニチャ 1206（IP Fragment Too Small）は、IPv6 データグラムに対して起動しません。Atomic IP Advanced エンジンのシグニチャ 1741 は、小さすぎる IPv6 フラグメントに対して起動します。
- シグニチャ 1202 は IPv6 ヘッダー フィールドが長いため、IPv6 の max-datagram-size を超えてさらに 48 バイトを追加できます。

詳細情報

- ノーマライザ エンジンに IP フラグメントの再構成シグニチャを設定する手順については、「IP フラグメント再構成の設定」（P.7-29）を参照してください。
- ノーマライザ エンジンに TCP ストリームの再構成シグニチャを設定する手順については、「TCP ストリーム再構成の設定」（P.7-33）を参照してください。

ノーマライザ エンジンのパラメータ

表 B-15 に、ノーマライザ エンジンに固有のパラメータを示します。

表 B-15　ノーマライザ エンジンのパラメータ

パラメータ	説明
edit-default-sigs-only	編集可能なシグニチャ。
specify-fragment-reassembly-timeout	（任意）フラグメント再構築タイムアウトをイネーブルにします。
specify-hijack-max-old-ack	（任意）hijack-max-old-ack をイネーブルにします。
specify-max-dgram-size	（任意）最大データグラム サイズをイネーブルにします。
specify-max-fragments	（任意）最大フラグメントをイネーブルにします。
specify-max-fragments-per-dgram	（任意）データグラムあたりの最大フラグメントをイネーブルにします。
specify-max-last-fragments	（任意）直前の最大フラグメントをイネーブルにします。
specify-max-partial-dgrams	（任意）最大部分データグラムをイネーブルにします。
specify-max-small-fragss	（任意）最大スモール フラグメントをイネーブルにします。
specify-min-fragment-size	（任意）最小フラグメント サイズをイネーブルにします。
specify-service-ports	（任意）サービス ポートをイネーブルにします。
specify-syn-flood-max-embryonic	（任意）SYN フラッドの最大初期接続をイネーブルにします。
specify-tcp-closed-timeout	（任意）TCP クローズド タイムアウトをイネーブルにします。
specify-tcp-embryonic-timeout	（任意）TCP 初期接続タイムアウトをイネーブルにします。
specify-tcp-idle-timeout	（任意）TCP アイドル タイムアウトをイネーブルにします。
specify-tcp-max-mss	（任意）TCP 最大 mss（最大セグメント サイズ）をイネーブルにします。
specify-tcp-max-queue	（任意）TCP 最大キューをイネーブルにします。
specify-tcp-min-mss	（任意）TCP 最小 mss をイネーブルにします。
specify-tcp-option-number	（任意）TCP オプション番号をイネーブルにします。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service エンジン

ここでは、Service エンジンについて説明します。次のような構成になっています。

- 「Service エンジンの概要」（P.B-36）
- 「Service DNS エンジン」（P.B-36）
- 「Service FTP エンジン」（P.B-38）
- 「Service Generic エンジン」（P.B-38）
- 「Service H225 エンジン」（P.B-40）
- 「Service HTTP エンジン」（P.B-42）
- 「Service IDENT エンジン」（P.B-44）
- 「Service MSRPC エンジン」（P.B-45）
- 「Service MSSQL エンジン」（P.B-46）
- 「Service NTP エンジン」（P.B-47）
- 「Service P2P エンジン」（P.B-47）
- 「Service RPC エンジン」（P.B-48）
- 「Service SMB Advanced エンジン」（P.B-49）
- 「Service SNMP エンジン」（P.B-51）
- 「Service SSH エンジン」（P.B-52）
- 「Service TNS エンジン」（P.B-53）

Service エンジンの概要

Service エンジンは、2 つのホスト間のレイヤ 5+ トラフィックを分析します。これらは、固定データを追跡する 1 対 1 シグニチャです。エンジンは、ライブ サービスに似た方法でレイヤ 5+ ペイロードを分析します。

Service エンジンには共通した特性がありますが、個々のエンジンには、検査対象のサービスに関する固有の情報が保持されます。文字列エンジンの使用が不適切または望ましくない場合には、Service エンジンによって、アルゴリズムに特化した汎用文字列エンジンの機能が補完されます。

Service DNS エンジン

表 B-16 に、Service DNS エンジンに固有のパラメータを示します。

表 B-16　Service DNS エンジンのパラメータ

パラメータ	説明	値
protocol	このインスペクタの該当プロトコル。	tcp udp
specify-query-chaos-string	（任意）DNS クエリー クラスのカオス文字列をイネーブルにします。	*query-chaos-string*

表 B-16　Service DNS エンジンのパラメータ （続き）

パラメータ	説明	値
specify-query-class	（任意）クエリー クラスをイネーブルにします。 • query-class：DNS クエリー クラスの 2 バイト値	0 ～ 65535
specify-query-invalid-domain-name	（任意）無効なドメイン名のクエリーをイネーブルにします。 • query-invalid-domain-name：255 を超える DNS クエリーの長さ	[true] | [false]
specify-query-jump-count-exceeded	（任意）しきい値を超えたジャンプ カウントのクエリーをイネーブルにします。 • query-jump-count-exceeded：DNS 圧縮カウンタ	[true] | [false]
specify-query-opcode	（任意）クエリー命令コードをイネーブルにします。 • query-opcode：DNS クエリー命令コードの 1 バイト値	0 ～ 65535
specify-query-record-data-invalid	（任意）無効なレコード データのクエリーをイネーブルにします。 • query-record-data-invalid：不完全な DNS レコード データ	[true] | [false]
specify-query-record-data-len	（任意）クエリー レコード データ長をイネーブルにします。 • query-record-data-len：DNS 応答レコードのデータ長	0 ～ 65535
specify-query-src-port-53	（任意）クエリー送信元ポート 53 をイネーブルにします。 • query-src-port-53：DNS パケット送信元ポート 53	[true] | [false]
specify-query-stream-len	（任意）クエリー ストリーム長をイネーブルにします。 • query-stream-len：DNS パケット長	0 ～ 65535
specify-query-type	（任意）クエリー タイプをイネーブルにします。 • query-type：DNS クエリー タイプの 2 バイト値	0 ～ 65535
specify-query-value	（任意）クエリー値をイネーブルにします。 • query-value：クエリー 0 応答 1	[true] | [false]

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service FTP エンジン

Service FTP エンジンは、FTP ポートのコマンド デコード専用です。無効な **port** コマンドや PASV ポート スプーフィングをトラップします。これは、String エンジンが検出に適さない場合のギャップを埋めます。パラメータはブール値であり、**port** コマンドのデコードにおけるさまざまなエラー トラップ条件にマッピングされます。Service FTP エンジンは、TCP ポート 20 および 21 で稼動します。ポート 20 はデータ用であり、Service FTP エンジンはこのポートに対する検査を行いません。Service FTP エンジンは、ポート 21 の制御トランザクションを検査します。

表 B-17 に、Service FTP エンジンに固有のパラメータを示します。

表 B-17　Service FTP エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
ftp-inspection-type	実行する検査のタイプ： • FTP ポート コマンド内の無効なアドレスを検索します。 • FTP ポート コマンド内の無効なポートを検索します。 • PASV ポート スプーフィングを検索します。	bad-port-cmd-address bad-port-cmd-port pasvI
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service Generic エンジン

Service Generic エンジンを使用すると、コンフィギュレーション ファイルでシグニチャを更新するだけで、プログラム シグニチャを発行できます。このエンジンには、コンフィギュレーション ファイルで定義されている簡易マシンおよびアセンブリ言語が含まれています。このエンジンは、仮想マシンを介して（アセンブリ言語から導出された）マシン コードを実行します。仮想マシンは、命令を処理し、パケットから重要な情報を引き出して、マシン コードに指定されている比較および演算を実行します。このエンジンは、String エンジンと State エンジンを補足する迅速なシグニチャ応答エンジンとして設計されています。

新しい機能は、正規表現パラメータを Service Generic エンジンと拡張命令に追加します。Service Generic エンジンは、パケットを解析するために作成されたミニプログラムに基づいてトラフィックを分析できます。これらのミニプログラムは、パケットを分析し、特定の条件を検索するコマンドで構成されます。

（注） カスタム シグニチャを作成するために、Service Generic エンジンを使用することはできません。

注意 この複雑な言語に特有の性質上、重大度およびイベント アクション以外に Service Generic エンジンのシグニチャ パラメータを編集することは推奨しません。

表 B-18 に、Service Generic エンジンに固有のパラメータを示します。

表 B-18　Service Generic エンジンのパラメータ

パラメータ	説明	値
specify-dst-port	（任意）宛先ポートをイネーブルにします。 • dst-port：このシグニチャの該当宛先ポート。	0 ～ 65535
specify-ip-protocol	（任意）IP プロトコルをイネーブルにします。 • ip-protocol：このインスペクタが検査する IP プロトコル。	0 ～ 255
specify-payload-source	（任意）ペイロード送信元検査をイネーブルにします。 • payload-source：次のタイプのペイロード送信元検査。 – ICMP データの検査 – レイヤ 2 ヘッダーの検査 – レイヤ 3 ヘッダーの検査 – レイヤ 4 ヘッダーの検査 – TCP データの検査 – UDP データの検査	icmp-data l2-header l3-header l4-header tcp-data udp-dataI
specify-src-port	（任意）送信元ポートをイネーブルにします。 • src-port：このシグニチャの該当送信元ポート。	0 ～ 65535
specify-regex-string	ポリシー タイプが regex の場合に検索する正規表現。 • 単独の TCP パケット内での検索に使用する正規表現。 • （任意）使用する最小一致長をイネーブルにします。 一致と見なされるために必要な正規表現の最小一致長です。	regex-string specify-min -match-leng th
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service H225 エンジン

Service H225 エンジンは、H225.0 プロトコルを分析します。このプロトコルは、多数のサブプロトコルで構成され、H.323 スイートの一部です。H.323 は、パケットベースのネットワーク上での会議開催を実現するために連携して動作する、複数のプロトコルとその他の標準の集まりです。

H.225.0 コール シグナリングおよびステータス メッセージは、H.323 コール セットアップの一部です。ゲートキーパーやエンドポイント端末など、ネットワーク内のさまざまな H.323 エンティティが、H.225.0 プロトコル スタックの実装を実行します。Service H225 エンジンは、H.225.0 プロトコルを分析して、複数の H.323 ゲートキーパー、VoIP ゲートウェイ、およびエンドポイント端末に対する攻撃を検出します。また、TCP PDU を介して交換されるコール シグナリング メッセージについて、ディープ パケット インスペクションを提供します。さらに、Service H225 エンジンは、H.225.0 プロトコルを分析することで無効な H.255.0 メッセージ、およびこれらのメッセージのさまざまなプロトコル フィールドの悪用やそれらに対するオーバーフロー攻撃を検出します。

H.225.0 コール シグナリング メッセージは、Q.931 プロトコルに基づいています。発信側エンドポイントは、Q.931 SETUP メッセージを着信側となるエンドポイントに送信します。着信側エンドポイントのアドレスは、許可手順またはいくつかのルックアップ手法を通じて取得します。着信側のエンドポイントは、Q.931 CONNECT メッセージを送信して接続を受け入れるか、または接続を拒否します。H.225.0 接続が確立されると、発信側エンドポイントまたは着信側エンドポイントのどちらかが H.245 アドレスを提供します。このアドレスを使用して、制御プロトコル（H.245）チャネルが確立されます。

SETUP コール シグナリング メッセージは、コール セットアップの一部として H.323 エンティティ間で交換される最初のメッセージであるため、特に重要です。SETUP メッセージはコール シグナリング メッセージでよく見られるフィールドの多くを使用しており、攻撃に晒される可能性のある実装ではほとんど SETUP メッセージのセキュリティ チェックに失敗します。そのため、H.225.0 SETUP メッセージの妥当性を確認し、ネットワーク境界に対してもチェックを実施することが非常に重要となります。

Service H225 エンジンには、H225 SETUP メッセージの TPKT 検証、Q.931 プロトコル検証、および ASN.1PER 検証を実行するためのシグニチャが組み込まれています。ASN.1 は、データ構造を記述するための表記です。PER は、異なる形式の符号化を使用します。PER は、データ タイプに基づいて符号化し、よりコンパクトな表現を生成することに特化しています。

Q.931 および TPKT 長さシグニチャを調整し、より細分化されたシグニチャを特定の H.225 プロトコル フィールドに追加および適用し、Q.931 または H.225 プロトコルの単一のフィールドに複数のパターン検索シグニチャを適用できます。

Service H225 エンジンは、次の機能をサポートします。

- TPKT 検証および長さチェック
- Q.931 情報要素の検証
- Q.931 情報要素内のテキスト フィールドの正規表現シグニチャ
- Q.931 情報要素の長さチェック
- SETUP メッセージの検証
- ASN.1 PER 符号化エラー チェック
- ULR-ID、E-mail-ID、h323-id などのフィールドの正規表現と長さの両方に対応する設定シグニチャ

TPKT および SAN.1 シグニチャの数は決まっています。これらのタイプのカスタム シグニチャは作成できません。TPKT シグニチャの場合は、長さシグニチャの値範囲だけを変更する必要があります。ASN.1 の場合、パラメータは変更しないでください。Q.931 シグニチャについては、テキスト フィールドの新規の正規表現シグニチャを追加できます。SETUP シグニチャについては、各種の SETUP メッセージ フィールドの長さおよび正規表現を確認するためのシグニチャを追加できます。

表 B-19 に、Service H225 エンジンに固有のパラメータを示します。

表 B-19　Service H.225 エンジンのパラメータ

パラメータ	説明	値
message-type	シグニチャを適用する H225 メッセージのタイプ。 • SETUP • ASN.1-PER • Q.931 • TPKT	asn.1-per q.931 setup tpkt
policy-type	シグニチャを適用する H225 ポリシーのタイプ: • フィールド長を検査する。 • 存在を検査する。 特定のフィールドがメッセージ内に存在する場合は、アラートが送信されます。 • 正規表現を検査する。 • フィールドの妥当性を検査する。 • 値を検査する。 TPKT シグニチャの場合、[regex] と [presence] は有効な値ではありません。	length presence regex validate value
specify-field-name	（任意）使用するフィールド名をイネーブルにします。SETUP および Q.931 メッセージ タイプにのみ有効です。このシグニチャを適用するフィールド名のドット付き表記を指定します。 • field-name：検査するフィールドの名前。	1 ～ 512
specify-invalid-packet-index	（任意）ASN と TPKT 固有のエラー、および固定マッピングを持つその他のエラーで使用する無効なパケット インデックスをイネーブルにします。 • invalid-packet-index：無効なパケット インデックスを検査します。	0 ～ 255
specify-regex-string	ポリシー タイプが [regex] の場合に検索する正規表現。TPKT シグニチャには設定しないでください。 • 単独の TCP パケット内での検索に使用する正規表現。 • （任意）使用する最小一致長をイネーブルにします。 一致と見なされるために必要な正規表現の最小一致長です。TPKT シグニチャには設定しないでください。	regex-string specify-min-match-length

表 B-19　Service H.225 エンジンのパラメータ （続き）

パラメータ	説明	値
specify-value-range	長さまたは値ポリシー タイプに有効です（0x00 ～ 6535）。その他のポリシー タイプの場合は無効です。 • value-range：値の範囲。	0 ～ 65535[1] a-b
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service HTTP エンジン

Service HTTP エンジンは、サービスに特化した文字列ベースのパターンマッチング検査エンジンです。HTTP プロトコルは、今日のネットワークで最も一般的に使用されるプロトコルの 1 つです。また、必要な前処理の時間が非常に長く、検査を必要とするシグニチャも非常に多いため、システムの全体的なパフォーマンスを決める要因になっています。

Service HTTP エンジンでは、複数のパターンを 1 つのパターンマッチング テーブルにまとめることでデータ内の検索を一度に実行できる正規表現ライブラリが使用されます。このエンジンは、Web サービスに向かう方向のみのトラフィック、または HTTP 要求を検索します。このエンジンでリターン トラフィックを検査することはできません。このエンジンでは、シグニチャごとに対象の Web ポートを別々に指定できます。

HTTP 解読とは、符号化された文字を ASCII 対応文字に正規化することによって、HTTP メッセージをデコードするプロセスです。このプロセスは、ASCII 正規化と呼ばれることもあります。

HTTP パケットを検査するには、あらかじめそのデータを、ターゲット システムでのデータ処理時に表示されるものと同じデータ表現として解読または正規化しておく必要があります。また、ホスト ターゲット タイプごとにカスタマイズされたデコード方式を用意することが推奨されます。そのためには、ターゲット上で動作しているオペレーティング システムおよび Web サーバのバージョンを確認する必要があります。Service HTTP エンジンのデフォルトの解読動作は Microsoft IIS Web サーバを対象としています。

表 B-20 に、Service HTTP エンジンに固有のパラメータを示します。

表 B-20　Service HTTP エンジンのパラメータ

パラメータ	説明	値
de-obfuscate	検索の前に反回避解読を適用します。	[true] | [false]
max-field-sizes	最大フィールド サイズ グループ。	—

表 B-20　Service HTTP エンジンのパラメータ （続き）

パラメータ	説明	値
specify-max-arg-field-length	（任意）引数フィールドの最大長をイネーブルにします。 • max-arg-field-length：引数フィールドの最大長。	0 ～ 65535
specify-max-header-field-length	（任意）ヘッダー フィールドの最大長をイネーブルにします。 • max-header-field-length：ヘッダー フィールドの最大長。	0 ～ 65535
specify-max-request-length	（任意）要求フィールドの最大長をイネーブルにします。 • max-request-length : 要求フィールドの最大長。	0 ～ 65535
specify-max-uri-field-length	（任意）URI フィールドの最大長をイネーブルにします。 • max-uri-field-length：URI フィールドの最大長。	0 ～ 65535
regex	正規表現グループ。	—
specify-arg-name-regex	（任意）特定の正規表現の引数フィールドの検索をイネーブルにします。 • arg-name-regex：HTTP 引数フィールド（コンテンツ長で定義されているとおりのエンティティ本体の中で、? の後ろ）で検索する正規表現。	—
specify-header-regex	（任意）特定の正規表現のヘッダー フィールドの検索をイネーブルにします。 • header-regexHTTP ヘッダー フィールドで検索する正規表現。 ヘッダーは、最初の CRLF の後ろから定義され、CRLFCRLF まで続きます。	—
specify-request-regex	（任意）特定の正規表現の要求フィールドの検索をイネーブルにします。 • request-regexHTTP URI フィールドと HTTP 引数フィールドの両方で検索する正規表現。 • specify-min-request-match-length：要求の最小一致長の設定をイネーブルにします。	0 ～ 65535
specify-uri-regex	（任意）HTTP URI フィールドで検索する正規表現。URI フィールドは、HTTP メソッド（たとえば、GET）の後ろで、最初の CRLF の前まで定義されます。正規表現は保護されています。つまり、値は変更できません。	[/\\][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][a-zA-Z][.]jpeg

表 B-20　Service HTTP エンジンのパラメータ （続き）

パラメータ	説明	値
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- Service HTTP カスタム シグニチャの例については、「Service HTTP エンジン シグニチャの例」（P.7-46）を参照してください。
- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service IDENT エンジン

Service IDENT エンジンは、TCP ポート 113 のトラフィックを検査します。基本デコードと、長さのオーバーフローを指定するパラメータが用意されています。たとえば、コンピュータ A のユーザまたはプログラムからコンピュータ B の ident 要求が実行された場合は、A と B 間の接続のユーザ ID のみが求められている可能性があります。B の ident サーバは、TCP ポート 113 の接続を受信します。A のクライアントは接続を確立し、その接続で使用している A と B のポートの番号を送信して、ID が必要な接続を指定します。B のサーバは、その接続を使用しているユーザを判断し、そのユーザの名前を示す文字列を A に返します。Service IDENT エンジンは、TCP ポート 113 での ident の不正利用を検査します。

表 B-21 に、Service IDENT エンジンに固有のパラメータを示します。

表 B-21　Service IDENT エンジンのパラメータ

パラメータ	説明	値
inspection-type	実行する検査のタイプ： • has-newline：ペイロードで非終端改行文字を検査します。 • has-bad-port：ペイロードで不良ポートを検査します。 • size：これよりも長いペイロード長を検査します。	—
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
direction	トラフィックの方向： • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service MSRPC エンジン

Service MSRPC エンジンは、MSRPC パケットを検査します。MSRPC によって、ネットワーク環境内の複数のコンピュータとそれらのアプリケーション ソフトウェアとの間の連携処理が可能になります。これは、トランザクションベースのプロトコルです。つまり、チャネルを確立し、処理要求と応答を渡す一連の通信が行われます。

MSRPC は、ISO レイヤ 5 ～ 6 プロトコルで、UDP、TCP、および SMB などのその他のトランスポート プロトコルの上層にあたります。MSRPC エンジンには、MSRPC PDU のフラグメンテーションと再構成を可能にする機能が組み込まれています。

この通信チャネルが、最近の Windows NT、Windows 2000、および Window XP セキュリティの脆弱性の原因です。Service MSRPC エンジンは、最も一般的なトランザクション タイプに対応する DCE および RPC プロトコルだけをデコードします。

表 B-22 に、Service MSRPC エンジンに固有のパラメータを示します。

表 B-22　Service MSRPC エンジンのパラメータ

パラメータ	説明	値
protocol	このインスペクタの該当プロトコル。 • type：UDP または TCP	tcp udp
specify-flags	設定するフラグ。 • msrpc-flags • msrpc-tcp-flags-mask	concurrent-execution did-not-execute first-fragment last-fragment maybe-semantics object-uuid reserved
specify-operation	（任意）MSRPC 動作の使用をイネーブルにします。 • operation：要求する MSRPC 動作。 SMB_COM_TRANSACTION コマンドに必要です。完全一致。	0 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

表 B-22　Service MSRPC エンジンのパラメータ （続き）

パラメータ	説明	値
specify-regex-string	（任意）正規表現文字列の使用をイネーブルにします。 • specify-exact-match-offset：完全一致オフセットをイネーブルにします。 – exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。 • specify-min-match-length：最小一致長をイネーブルにします。 – min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
specify-uuid	（任意）UUID をイネーブルにします。 • uuid：MSRPC UUID フィールド。	000001a000000000c000000000000046

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service MSSQL エンジン

Service MSSQL エンジンは、Microsoft SQL サーバによって使用されるプロトコルを検査します。このエンジンには 1 つの MSSQL シグニチャが含まれています。このエンジンは、デフォルトの sa アカウントを使用した MSSQL サーバへのログイン試行を検出すると、アラートを起動します。ログイン ユーザ名や、パスワードが使用されたかどうかなど、MSSQL プロトコル値に基づいてカスタム シグニチャを追加できます。

表 B-23 に、Service MSSQL エンジンに固有のパラメータを示します。

表 B-23　Service MSSQL エンジンのパラメータ

パラメータ	説明	値
password-present	MS SQL ログインでパスワードが使用されたかどうか。	[true] | [false]
specify-sql-username	（任意）SQL ユーザ名の使用をイネーブルにします。 • sql-username：MS SQL サービスにログインするユーザのユーザ名（完全一致）。	sa

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service NTP エンジン

Service NTP エンジンは、NTP プロトコルを検査します。このエンジンには、1 つの NTP シグニチャ（NTP readvar オーバーフロー シグニチャ）が含まれます。このシグニチャは、サイズが大きいため NTP サービスでキャプチャできない NTP データが、readvar コマンドに指定されていることを検出した場合に、アラートを起動します。NTP プロトコルの値（モードや制御パケットのサイズなど）に基づいて、シグニチャを調整したり、カスタム シグニチャを作成したりできます。

表 B-24 に、Service NTP エンジンに固有のパラメータを示します。

表 B-24　Service NTP エンジンのパラメータ

パラメータ	説明	値
inspection-type	実行する検査のタイプ。	
inspect-ntp-packets	NTP パケットを検査します。 • control-opcode：RFC1305 の付録 B に基づく NTP 制御パケットの命令コード番号。 • max-control-data-size：制御パケットで送信されるデータの最大許容量。 • mode：RFC 1305 に基づく NTP パケットの動作モード。	0 ～ 65535
is-invalid-data-packet	無効な NTP データ パケットを検索します。NTP データ パケットの構造を調べ、サイズが正しいことを確認します。	[true] | [false]
is-non-ntp-traffic	NTP ポートの非 NTP パケットをチェックします。	[true] | [false]

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service P2P エンジン

P2P ネットワークでは、ファイル共有のためにクライアントとサーバの両方として同時に機能できるノードを使用します。P2P ネットワークには著作物が含まれていることが多く、企業ネットワークでそれらを使用することは会社のポリシーに違反する可能性があります。Service P2P エンジンはそのようなネットワークをモニタし、最適化された TCP および UDP P2P プロトコル ID を提供します。Service P2P エンジンには、次の特性があります。

- すべての TCP ポートおよび UDP ポートで受信します。
- 正規表現ではなく、ハードコード化されたシグニチャの使用によってパフォーマンスを向上します。
- P2P プロトコルが識別されるか、P2P プロトコルは識別されずに 10 パケットを確認した後、トラフィックを無視します。

P2P シグニチャはハード コード化されているので、編集できるパラメータは Master エンジンのパラメータのみになります。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service RPC エンジン

Service RPC エンジンは、RPC プロトコルに特化しており、反回避の方式として完全なデコードが行われます。これにより、フラグメント化されたメッセージ（複数パケット内の 1 つのメッセージ）およびバッチ メッセージ（1 つのパケット内の複数メッセージ）を処理できます。

RPC ポートマッパーは、ポート 111 上で動作します。通常の RPC メッセージは、550 より上位であれば任意のポートで送受信できます。RPC スイープは、TCP ポート スイープとほぼ同じものです。異なるのは、有効な RPC メッセージが送信された場合に一意のポートだけをカウントするという点です。RPC は、UDP 上でも実行されます。

表 B-25 に、Service RPC エンジンに固有のパラメータを示します。

表 B-25　Service RPC エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
protocol	該当プロトコル。	tcp udp
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
specify-regex-string	（任意）正規表現文字列の使用をイネーブルにします。 • specify-exact-match-offset：完全一致オフセットをイネーブルにします。 – exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。 • specify-min-match-length：最小一致長をイネーブルにします。 – min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
specify-is-spoof-src	（任意）スプーフィングの送信元アドレスをイネーブルにします。 • is-spoof-src：送信元アドレスが 127.0.0.1 の場合にアラートを起動します。	[true] | [false]
specify-port-map-program	（任意）ポートマッパー プログラムをイネーブルにします。 • port-map-program：このシグニチャのポートマッパーに送信されたプログラム番号。	0 ～ 9999999999
specify-rpc-max-length	（任意）RPC 最大長をイネーブルにします。 • rpc-max-length：RPC メッセージ全体の最大許容長。長さが指定した値より長いとアラートを起動します。	0 ～ 65535

表 B-25　Service RPC エンジンのパラメータ （続き）

パラメータ	説明	値
specify-rpc-procedure	（任意）RPC プロシージャをイネーブルにします。 • rpc-procedure：このシグニチャの RPC プロシージャ番号。	0 ～ 1000000
specify-rpc-program	（任意）RPC プログラムをイネーブルにします。 • rpc-program：このシグニチャの RPC プログラム番号。	0 ～ 1000000
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service SMB Advanced エンジン

（注）　SMB エンジンは、SMB Advanced エンジンに置き換えられました。SMB エンジンがまだ IDM、IME、および CLI に表示される場合でも、このシグニチャは廃止されています。つまり、新規のシグニチャには、対応する古いシグニチャの ID を持つ obsoletes パラメータ セットがあります。新規の SMB Advanced エンジンを使用して、SMB エンジンにあったカスタム シグニチャを書き換えてください。

Service SMB Advanced エンジンは、SMB パケットを介して Microsoft SMB および Microsoft RPC を処理します。Service SMB Advanced エンジンは、コネクション型 MSRPC に同じデコード メソッドを使用します。これは、MSRPC エンジンに、MSRPC パケットは SMB プロトコルを経由する必要があるという要件があるためです。Service SMB Advanced エンジンは、TCP ポート 139 および 445 での SMB を経由した MSRPC をサポートしています。また、MSRPC エンジンからのコネクション型 DCS/RPC コードのコピーを使用します。

表 B-26 に、Service SMB Advanced エンジンに固有のパラメータを示します。

表 B-26　Service SMB Advanced エンジンのパラメータ

パラメータ	説明	値
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535 a-b[,c-d] [1]
specify-command	（任意）SMB コマンドをイネーブルにします。 • command：SMB コマンドの値。完全に一致する必要があります。SMB パケットのタイプを定義します。[2]	0 ～ 255

表 B-26　Service SMB Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-direction	（任意）トラフィック方向をイネーブルにします。 • direction：トラフィックの方向を指定できます。 – from-service：サービス ポートからクライアント ポート宛のトラフィック。 – to-service：クライアント ポートからサービス ポート宛のトラフィック。	from service to service
specify-operation	（任意）MSRPC over SMB をイネーブルにします。 • msrpc-over-smb-operation：SMB_COM_TRANSACTION コマンドに使用します。完全に一致する必要があります。	0 ～ 65535
specify-regex-string	（任意）正規表現文字列の検索をイネーブルにします。 • regex-string：単一の TCP パケット内で検索する正規表現。	
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。	
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：正規表現文字列が一致する必要がある最小バイト数。	
specify-payload-source	（任意）ペイロード送信元をイネーブルにします。 • payload-source：ペイロード送信元の検査。[3]	
specify-scan-interval	（任意）スキャン間隔をイネーブルにします。 • scan-interval：アラート率の計算に使用される間隔（秒単位）。	1 ～ 131071
specify-tcp-flags	（任意）TCP フラグをイネーブルにします。 • msrpc-tcp-flags • msrpc-tcp-flags-mask	• concurrent execution • did not execute • first fragment • last fragment • maybe • object UUID • pending cancel • reserved

表 B-26　Service SMB Advanced エンジンのパラメータ （続き）

パラメータ	説明	値
specify-type	（任意）MSRPC over SMB パケットのタイプをイネーブルにします。 • type：SMB パケットを介した MSRPC の Type フィールド。	• [0] = 要求 • [2] = 応答 • [11] = バインド • [12] = バインド応答
specify-uuid	（任意）UUID を経由した MSRPC をイネーブルにします。 • uuid：MSRPC UUID フィールド。	16 進数の 0 ～ 9、a ～ f、A ～ F で構成される 32 文字の文字列。
specify-hit-count	（任意）ヒット カウントをイネーブルにします。 • hit-count：scan-interval 内の発生回数のしきい値。この値を超えるとアラートが起動されます。	1 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	[true] | [false]

1. 範囲の 2 番目の数は、最初の数以上である必要があります。
2. 現在、37 (0x25) SMB_COM_TRANSACTION コマンド \x26amp および 162 (0xA2) SMB COM NT CREATE ANDX コマンドをサポートしています。
3. TCP_Data は、パケット全体に対して正規表現を実行します。SMB_Data は、SMB ペイロードでのみ正規表現を実行します。Resource_DATA は、SMB_Resource に対して正規表現を実行します。

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

Service SNMP エンジン

Service SNMP エンジンは、ポート 161 宛のすべての SNMP パケットを検査します。特定のコミュニティ名とオブジェクト ID に基づいて、SNMP シグニチャを調整したり、カスタム SNMP シグニチャを作成したりできます。

コミュニティ名とオブジェクト ID を照合するために、文字列比較や正規表現演算を使用する代わりに、整数を使用してすべての比較を実行し、プロトコル デコードを高速化しストレージ要件を削減します。

表 B-27 に、Service SNMP エンジンに固有のパラメータを示します。

表 B-27　Service SNMP エンジンのパラメータ

パラメータ	説明	値
inspection-typeI	実行する検査のタイプ。	—
brute-force-inspection	総当たり攻撃の試行を検査します。 • brute-force-count：総当たり攻撃と見なされる一意の SNMP コミュニティ名の数。	0 ～ 65535
invalid-packet-inspection	SNMP プロトコル違反を検査します。	—
non-snmp-traffic-inspection	UDP ポート 161 宛の非 SNMP トラフィックを検査します。	—
snmp-inspection	SNMP トラフィックを検査します。 • specify-community-name [yes | no] – community-name：SNMP コミュニティ名（SNMP パスワード）を検索します。 • specify-object-id [yes | no] – object-id：SNMP オブジェクト ID を検索します。	*community-name* *object-id*

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service SSH エンジン

Service SSH エンジンは、ポート 22 の SSH トラフィックに対して使用します。SSH セッションのセットアップを除いてすべてが暗号化されるため、Service SSH エンジンはセットアップのフィールドだけをモニタします。SSH には 2 つのデフォルト シグニチャがあります。これらのシグニチャを調整することはできますが、カスタム シグニチャは作成できません。

表 B-28 に、Service SSH エンジンに固有のパラメータを示します。

表 B-28　Service SSH エンジンのパラメータ

パラメータ	説明	値
length-type	次の SSH 長さタイプのいずれかを検査します。 • key-length：検査対象の SSH キーの長さ。 – length：キーがこれよりも長い場合は、RSAREF オーバーフローが発生します。 • user-length：ユーザ長の SSH 検査。 – length：キーがこれよりも長い場合は、RSAREF オーバーフローが発生します。	0 ～ 65535

表 B-28　Service SSH エンジンのパラメータ （続き）

パラメータ	説明	値
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
specify-packet-depth	（任意）パケット数をイネーブルにします。 • packet-depth：セッション キーが失われたと判断するまでにモニタするパケット数。	0 ～ 65535

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Service TNS エンジン

Service TNS エンジンは、TNS プロトコルを検査します。TNS は、業界標準のすべてのネットワーク プロトコルに単一の共通インターフェイスを備えたデータベース アプリケーションを提供します。TNS を使用すると、アプリケーションは異なるプロトコルを使用しているネットワークを経由して他のデータベース アプリケーションに接続することができます。デフォルトの TNS 受信ポートは TCP 1521 です。TNS は、クライアントを別のホストまたは他の TCP ポート（またはその両方）にリダイレクトする REDIRECT フレームもサポートしています。REDIRECT パケットをサポートするため、TNS エンジンはすべての TCP ポートで受信します。また、TNS 以外のストリームを無視するための簡単な TNS フレーム ヘッダー検証ルーチンを備えています。

表 B-29 に、Service TNS エンジンに固有のパラメータを示します。

表 B-29　Service TNS エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
type	TNS フレーム値のタイプを指定します。 • [1]：接続 • [2]：受け入れ • [4]：拒否 • [5]：リダイレクト • [6]：データ • [11]：再送信 • [12]：マーカー	1 2 4 5 6 11 12

表 B-29　Service TNS エンジンのパラメータ （続き）

パラメータ	説明	値
specify-regex-string	（任意）正規表現文字列の使用をイネーブルにします。 • specify-exact-match-offset：完全一致オフセットをイネーブルにします。 – exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。 • specify-min-match-length：最小一致長をイネーブルにします。 – min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
specify-regex-payload-src	検査するプロトコルを指定します。 • payload-src： – tcp-data：TCP パケットのデータ部分に対して正規表現を実行します。 – tns-data：すべての空白が削除されている TNS データに対してだけ正規表現を実行します。	tcp tns

詳細情報

- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。
- シグニチャの正規表現構文の一覧については、「正規表現の構文」（P.B-10）を参照してください。

State エンジン

State エンジンは、TCP ストリームに対して状態ベースで、正規表現ベースのパターン検査を行います。State エンジンとは、何らかの状態を保存するデバイスで、特定の時に入力に基づいて 1 つの状態から別の状態に移行したり、処理または出力を発生させたりすることができます。ステート マシンは、出力やアラートの原因となる特定のイベントを記述するために使用します。State エンジンには、SMTP、Cisco Login、および LPR Format String の 3 つのステート マシンがあります。

表 B-30 に、State エンジンに固有のパラメータを示します。

表 B-30　State エンジンのパラメータ

パラメータ	説明	値
state-machine	ステート マシン グループ。	• smpt • lpr-format-string • cisco-login
cisco-login	Cisco ログインのステート マシンを指定します。 • state-name：状態の名前。この状態になると、シグニチャはアラートを起動します。 – シスコ デバイスの状態 – Control-C 状態 – パスワード プロンプト状態 – 開始状態	• cisco-device • control-c • pass-prompt • start
lpr-format-string	LPR フォーマット ストリングの脆弱性を検査するステート マシンを指定します。 • state-name：状態の名前。この状態になると、シグニチャはアラートを起動します。 – LPR フォーマット ストリング検査を終了する中断状態 – フォーマット文字の状態 – 開始状態	• abort • format-char • start
state-name	SMTP プロトコルのステート マシンを指定します。 • state-name：状態の名前。この状態になると、シグニチャはアラートを起動します。 – LPR フォーマット ストリング検査を終了する中断状態 – メール本文の状態 – メール ヘッダーの状態 – SMTP コマンドの状態 – 開始状態	• abort • mail-body • mail-header • smtp-commands • start
direction	トラフィックの方向： • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]

表 B-30　State エンジンのパラメータ （続き）

パラメータ	説明	値
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。	0 ～ 65535
specify-max-match-offset	（任意）最大一致オフセットをイネーブルにします。 • max-match-offset：一致を有効にするために正規表現文字列が報告する必要がある最大ストリーム オフセット。	0 ～ 65535
specify-min-match-offset	（任意）最小一致オフセットをイネーブルにします。 • min-match-offset：一致を有効にするために正規表現文字列が報告する必要がある最小ストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

String エンジン

String エンジンは、ICMP、TCP、および UDP の各プロトコルを対象とした、汎用のパターンマッチング検査エンジンです。String エンジンでは、複数のパターンを 1 つのパターンマッチング テーブルにまとめることでデータ内の検索を一度に実行できる正規表現エンジンが使用されます。String エンジンには、String ICMP、String TCP、および String UDP の 3 種類が存在します。

表 B-31 に、String ICMP エンジンに固有のパラメータを示します。

表 B-31　String ICMP エンジンのパラメータ

パラメータ	説明	値
direction	トラフィックの方向： • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
icmp-type	ICMP ヘッダーの TYPE 値。	0 ～ 18[1] a-b[,c-d]
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

表 B-32 に、String TCP エンジンに固有のパラメータを示します。

表 B-32　String TCP エンジン

パラメータ	説明	値
direction	トラフィックの方向： • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535

表 B-32　String TCP エンジン （続き）

パラメータ	説明	値
strip-telnet-options	パターンを検索する前に、データから Telnet オプション文字を削除します。[2]	[true] | [false]
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

1. 範囲の 2 番目の数は、最初の数以上である必要があります。
2. このパラメータは、主に、IPS 反回避ツールとして使用します。

表 B-33 に、String UDP エンジンに固有のパラメータを示します。

表 B-33　String UDP エンジン

パラメータ	説明	値
direction	トラフィックの方向： • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
service-ports	ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。	0 ～ 65535[1] a-b[,c-d]
specify-exact-match-offset	（任意）完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット。	0 ～ 65535
specify-min-match-length	（任意）最小一致長をイネーブルにします。 • min-match-length：正規表現文字列が一致する必要がある最小バイト数。	0 ～ 65535
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

1. 範囲の 2 番目の数は、最初の数以上である必要があります。

詳細情報

- カスタム String エンジン シグニチャの例については、「String TCP エンジン シグニチャの例」（P.7-43）を参照してください。
- すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

String XL エンジン

String XL エンジンは、その他の String エンジンと同じことを行います（つまり、1 つのシグニチャあたりに 1 つの文字列のマッチング機能を提供します）が、使用する正規表現構文が異なります。String TCP XL エンジンはストリームベースであり、Cross-Packet Inspection（XPI）を使用します。パケットは、順番に並んでいる必要があります。UDP と ICMP はどちらもステートレスであるので、String UDP XL および String ICMP XL シグニチャ エンジンはセッション ステートを割り当てる必要がなく、各パケットは別個の検索になります。

正規表現アクセラレータ カードは、標準の String エンジンと新規の String XL エンジンの両方に使用されます。ほとんどの標準 String エンジン シグニチャは、変更することなく、正規表現アクセラレータ カードによってコンパイルおよび分析できます。ただし、標準の String エンジン シグニチャを正規表現アクセラレータ カード用にコンパイルできない特別な状況もあります。そのような状況では、正規表現アクセラレータ カードでコンパイルする String XL エンジンの特定のパラメータを使用して、String XL エンジンに新規のシグニチャが作成されます。String XL エンジンの新規シグニチャは、標準の String エンジンにある元のシグニチャに代わるものです。

正規表現構文または raw 表現構文を使用できますが、raw 表現構文はエキスパート ユーザ専用です。String XL シグニチャを設定するときは、raw 表現構文を使用していない限り、regex-string パラメータが必要です。

（注）　raw 正規表現とは、raw モード処理に使用される正規表現の構文です。エキスパート モード専用であり、Cisco IPS シグニチャ開発チーム、あるいはその監視下にある作業者のみによる使用を対象としています。String XL シグニチャは、通常の正規表現と raw 正規表現のいずれでも設定できます。

表 B-34 に、String XL エンジン（TCP、ICMP、および UDP）に固有のパラメータを示します。

表 B-34　String XL エンジンのパラメータ

パラメータ	説明	値
direction	（必須）検査するトラフィックの方向。 • サービス ポートからクライアント ポート宛のトラフィック。 • クライアント ポートからサービス ポート宛のトラフィック。	from-service to-service
dot-all	true に設定すると、\n を含む [\x00-\xFF] と一致し、false に設定すると、\n を除く範囲 [\x00-\xFF] 内のすべてと一致します。	true | false（デフォルト）
end-optional	パケットの最後で、その他すべての条件が満たされても最後が表示されない場合、最小限度を超えていれば一致結果が報告されます。	true | false（デフォルト）
icmp-type	ICMP メッセージ ヘッダーのタイプ。シグニチャ エンジンが string-icmp である場合は必須です。	0 ～ 18[1] a-b[,c-d]
no-case	表現内のすべてのアルファベット文字の大文字と小文字を区別します。	true | false（デフォルト）

表 B-34　String XL エンジンのパラメータ （続き）

パラメータ	説明	値
raw-regex	true に設定すると、min-match-length、max-match-length、min-whole-length、max-whole-length、dot-all、utf8、no-case、stingy、および end-optional は正規表現文字列の再フォーマットに使用されません。 （注）　raw-regex を使用すると、正規表現文字列を変換せずに raw 構文に入力できます。	true | false（デフォルト）
regex-string	（必須）検索に使用する正規表現パターン。 （注）　max-stream-length が設定されていない限り、このパラメータは必須です。max-stream-length が設定されている場合は、regex-string を設定しないでください。	string
service-ports	（必須）ターゲット サービスが常駐する、カンマ区切りのポートのリストまたはポート範囲。 （注）　このパラメータは、String XL TCP シグニチャ エンジンおよび String XL UDP シグニチャ エンジンに必須です。String XL ICMP シグニチャ エンジンには、使用できません。	0 ～ 65535[2] a-b[,c-d]
specify-exact-match-offset	完全一致オフセットをイネーブルにします。 • exact-match-offset：一致を有効にするために正規表現文字列が報告する必要がある正確なストリーム オフセット（バイト単位）。	[yes] | [no] 0 ～ 65535
specify-max-match-offset	最大一致オフセットをイネーブルにします。 • maximum-match-offset：一致を有効にするために正規表現文字列が報告する必要がある最大ストリーム オフセット。	yes | no 0 ～ 65535
specify-min-match-offset	最小一致オフセットをイネーブルにします。 • min-match-offset：一致を有効にするために正規表現文字列が報告する必要がある最小ストリーム オフセット（バイト単位）。	yes | no 0 ～ 65535
specify-max-match-length	最大一致長をイネーブルにします。 • max-match-length：パターンがヒットと見なされるために正規表現文字列が一致する必要のある最大バイト数。	yes | no 0 ～ 65535
specify-min-match-length	最小一致長をイネーブルにします。 • min-match-length：パターンがヒットと見なされるために正規表現文字列が一致する必要のある最小バイト数。	yes | no 0 ～ 65535

表 B-34　String XL エンジンのパラメータ （続き）

パラメータ	説明	値
specify-max-stream-length	最大ストリーム長をイネーブルにします。 • max-stream-length：最初に設定されたバイト数に検索を制限します。ストリームの長さが、この値でチェックされます。ストリームのバイト数がこの値よりも多い場合は、アラートがトリガーされます。 （注）　このパラメータを指定すると、raw-regex または regex-string を設定できません。	[yes] | [no] 0 ～ 65535
specify-max-whole-length	最大全長をイネーブルにします。 • max-whole-length：フラグメント化されないパターンの最大長。	[yes] | [no] 0 ～ 65535
specify-min-whole-length	最小全長をイネーブルにします。 • min-whole-length：フラグメント化されないパターンの最小長。	yes | no 0 ～ 65535
stingy	最初に完全に一致した後は、それより大きい一致検索を停止します。 （注）　stingy は、min-match-length を指定した場合のみ使用できます。それ以外では、無視されます。	true | false（デフォルト）
strip-telnet-options	パターンを検索する前に、データから Telnet オプション文字を削除します。[3]	true | false（デフォルト）
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true | false（デフォルト）
utf8	表現内のすべての公正な UTF-8 バイト シーケンスを 1 文字として処理します。	true | false（デフォルト）

1. 範囲の 2 番目の数は、最初の数以上である必要があります。
2. 範囲の 2 番目の数は、最初の数以上である必要があります。
3. このパラメータは、主に、IPS 反回避ツールとして使用します。

サポートされないパラメータ

end-optional パラメータと specify-max-stream-length パラメータは、String XL エンジンに表示されますが、IPS 7.1(1)E4 では無効です。これらのパラメータを設定しようとすると、エラー メッセージを受け取ります。たとえば、specify-max-stream-length を使用してシグニチャを作成し、保存しようとすると、次のエラー メッセージが表示されます。

```
Apply Changes?[yes]: yes
Error: string-xl-tcp 60003.0 : Maximum Stream Length is currently not supported.
Please don't use this option.

The configuration changes failed validation, no changes were applied.
Would you like to return to edit mode to correct the errors? [yes]:
```

Sweep エンジン

ここでは、Sweep エンジンについて説明します。次のような構成になっています。

- 「Sweep エンジン」（P.B-62）
- 「Sweep Other TCP エンジン」（P.B-64）

Sweep エンジン

Sweep エンジンは、2 つのホスト間または 1 つのホストから多数のホストへのトラフィックを分析します。既存のシグニチャを調整することも、カスタム シグニチャを作成することもできます。Sweep エンジンには、ICMP、UDP、および TCP 用のプロトコル固有のパラメータがあります。

Sweep エンジンのアラート条件は、根本的に一意のパラメータのカウントに基づいています。一意のパラメータとは、スイープのタイプに応じて明確に特定されたホスト数またはポート数のしきい値です。一意のパラメータは、一定時間内にアドレス セット上で一意の数を超えるポートまたはホストが検出された場合に、アラームをトリガーします。一意のポートおよびホストのトラッキング処理をカウンティングと言います。

注意　送信元 IP アドレスおよび宛先 IP アドレスに基づくイベント アクション フィルタは、正規のシグニチャとしてフィルタリングしないので、Sweep エンジンには機能しません。スイープ アラートで送信元 IP アドレスと宛先 IP アドレスをフィルタリングするには、Sweep エンジン シグニチャで送信元 IP アドレスおよび宛先 IP アドレス フィルタ パラメータを使用します。

Sweep エンジン内のすべてのシグニチャに、一意のパラメータを指定する必要があります。2 ～ 40 の制限がスイープに対して適用されます。2 は、スイープの絶対最小値であり、それに満たない場合は（1 つのホストまたはポートの）スイープではありません。40 は実際の最大値で、スイープが過剰にメモリを消費しないように適用する必要があります。さらに現実的な一意の範囲の値は 5 ～ 15 です。

TCP スイープには、どのスイープ インスペクタ スロットで特定の接続をカウントするかを決定するために、TCP フラグとマスクを指定する必要があります。ICMP スイープには、さまざまなタイプの ICMP パケットを識別するために、ICMP タイプを指定する必要があります。

DataNode

Sweep エンジン シグニチャに関連するアクティビティが検出されると、IPS は DataNode を使用して、特定のホストに対するモニタリングを停止するタイミングを決定します。DataNode には、ストリームのクロスパケット再構成用に、また、ストリーム単位、送信元単位、および宛先単位の検査状態を追跡するために必要とされる、さまざまな固定カウンタおよび変数が含まれます。スイープを含む DataNode は、スイープの期限を決定します。DataNode は、*x* 秒間（プロトコルによる）トラフィックが検出されなかった場合に、スイープを停止します。

DataNode には、いくつかの適応型タイムアウトがあります。DataNode は、包含されるオブジェクトがすべて削除されてから、アドレス セットに対して 30 秒のアイドル時間経過後に期限切れとなります。包含されている各オブジェクトにはさまざまなタイムアウトが設定されています。たとえば、TCP ストリームには確立された接続に対する 1 時間のタイムアウトがあります。その他のほとんどのオブジェクトに設定されている期限は、5 ～ 60 秒など、はるかに短い時間になります。

表 B-35 に、Sweep エンジンに固有のパラメータを示します。

表 B-35　Sweep エンジンのパラメータ

パラメータ	説明	値
dst-addr-filter	スイープ カウント アルゴリズムから除外する宛先 IP アドレス。	<A.B.C.D>-<A.B.C.D>[,<A.B.C.D>-<A.B.C.D>]
src-addr-filter	スイープ カウント アルゴリズムから除外する送信元 IP アドレス。	<A.B.C.D>-<A.B.C.D>[,<A.B.C.D>-<A.B.C.D>]
protocol	このインスペクタの該当プロトコル。	• icmp • udp • tcp
specify-icmp-type	（任意）ICMP ヘッダー タイプの検査をイネーブルにします。 • icmp-type：ICMP ヘッダーの TYPE 値を指定します。	0 ～ 255
specify-port-range	（任意）検査でのポート範囲の使用をイネーブルにします。 • port-range：検査で使用する UDP ポート範囲。	0 ～ 65535 a-b[,c-d]
fragment-status	フラグメントが必要かどうかを指定します。 • 任意のフラグメント ステータス。 • フラグメントを検査しない。 • フラグメントを検査する。	• any • no-fragments • want-fragments
inverted-sweep	一意のカウントの対象として宛先ポートではなく送信元ポートを使用します。	[true] | [false]
mask	TCP フラグの比較に使用するマスク： • URG ビット • ACK ビット • PSH ビット • RST ビット • SYN ビット • FIN ビット	• urg • ack • psh • rst • syn • fin
storage-key	固定データを保存するために使用するアドレス キーのタイプ。 • 攻撃者のアドレス • 攻撃者と攻撃対象のアドレス • 攻撃者のアドレスと攻撃対象のポート	Axxx AxBx Axxb
suppress-reverse	このアドレス セットで反対方向にスイープが実行されている場合、アラートを起動しません。	true| false
swap-attacker-victim	アラート メッセージおよびアクションで攻撃者および攻撃対象のアドレスとポート（送信元および宛先）がスワップする場合は、true。スワップしない場合は false（デフォルト）。	true| false

表 B-35　Sweep エンジンのパラメータ （続き）

パラメータ	説明	値
tcp-flags	マスクによってマスクされた場合に照合する TCP フラグ。 • URG ビット • ACK ビット • PSH ビット • RST ビット • SYN ビット • FIN ビット	• urg • ack • psh • rst • syn • fin
unique	2 つのホスト間の一意のポート接続数のしきい値。	0 ～ 65535

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Sweep Other TCP エンジン

Sweep Other TCP エンジンは、2 つのホスト間のトラフィックを分析して、一般に攻撃対象のフィンガープリントに使用される異常パケットを検索します。既存のシグニチャを調整することも、カスタム シグニチャを作成することもできます。

TCP スイープには、TCP フラグとマスクを指定する必要があります。TCP フラグのセットに複数のエントリを指定できます。また、オプションのポート範囲を指定して、特定のパケットを除外することもできます。

表 B-36 に、Sweep Other TCP エンジンに固有のパラメータを示します。

表 B-36　Sweep Other TCP エンジンのパラメータ

パラメータ	説明	値
specify-port-range	（任意）検査でのポート範囲の使用をイネーブルにします。 • port-range：検査で使用する UDP ポート範囲。	0 ～ 65535 a-b[,c-d]
set-tcp-flags	照合する TCP フラグを設定します。 • tcp-flags：この検査で使用される TCP フラグ。 – URG ビット – ACK ビット – PSH ビット – RST ビット – SYN ビット – FIN ビット	• urg • ack • psh • rst • syn • fin

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Traffic Anomaly エンジン

（注）　異常検出シグニチャを編集または調整することはできますが、カスタムの異常検出シグニチャを作成することはできません。

Traffic Anomaly エンジンには、3 つのプロトコル（TCP、UDP、およびその他）をカバーする 9 つの異常検出シグニチャが含まれます。各シグニチャには 2 つのサブシグニチャがあります。一方はスキャナ用で、もう一方はワームに感染したホスト（またはワーム攻撃されているスキャナ）用です。異常検出で異常が見つかると、これらのシグニチャに関するアラートがトリガーされます。すべての異常検出シグニチャは、デフォルトでイネーブルになり、各シグニチャのアラート重大度は高く設定されます。

スキャナが検出されても、ヒストグラム異常が発生しない場合、スキャナ シグニチャはその攻撃者（スキャナ）の IP アドレスをファイルに保存します。ヒストグラム シグニチャがトリガーされた場合は、スキャンを行っている攻撃者のアドレスによってそれぞれ（スキャナ シグニチャではなく）ワーム シグニチャがトリガーされます。アラートの詳細は、ヒストグラムがトリガーされたため、ワーム検出に使用されるしきい値を示します。その時点から、すべてのスキャナはワームに感染したホストとして検出されます。

次の異常検出イベント アクションが可能です。

- produce-alert：イベントをイベント ストアに書き込みます。
- deny-attacker-inline：指定された期間、この攻撃者のアドレスから発生したこのパケットおよび将来のパケットを送信しません。
- log-attacker-packets：攻撃者のアドレスが含まれるパケットに対する IP ロギングを開始します。
- log-pair-packets：攻撃者と攻撃対象のアドレス ペアが含まれているパケットの IP ロギングを開始します。
- deny-attacker-service-pair-inline：送信元 IP アドレスと宛先ポートをブロックします。
- request-snmp-trap：SNMP 通知を実行するよう、NotificationApp に要求を送信します。
- request-block-host：このホスト（攻撃者）をブロックするよう、ARC に要求を送信します。

表 B-37 に、異常検出ワーム シグニチャのリストを示します。

表 B-37　異常検出ワーム シグニチャ

シグニチャ ID	サブシグニチャ ID	名前	説明
13000	0	Internal TCP Scanner	内部ゾーンで TCP プロトコルを介して単一スキャナを識別しました。
13000	1	Internal TCP Scanner	内部ゾーンで TCP プロトコル上にワーム攻撃を識別しました。TCP ヒストグラムのしきい値を超え、TCP プロトコル上にスキャナが識別されました。
13001	0	Internal UDP Scanner	内部ゾーンで UDP プロトコル上に単一スキャナを識別しました。
13001	1	Internal UDP Scanner	内部ゾーンで UDP プロトコル上にワーム攻撃を識別しました。UDP ヒストグラムのしきい値を超え、UDP プロトコル上にスキャナが識別されました。

表 B-37　異常検出ワーム シグニチャ （続き）

シグニチャ ID	サブシグニチャ ID	名前	説明
13002	0	Internal Other Scanner	内部ゾーンでほかのプロトコル上に単一スキャナを識別しました。
13002	1	Internal Other Scanner	内部ゾーンでほかのプロトコル上にワーム攻撃を識別しました。ほかのヒストグラムのしきい値を超え、ほかのプロトコル上にスキャナが識別されました。
13003	0	External TCP Scanner	外部ゾーンで TCP プロトコルを介して単一スキャナを識別しました。
13003	1	External TCP Scanner	外部ゾーンで TCP プロトコル上にワーム攻撃を識別しました。TCP ヒストグラムのしきい値を超え、TCP プロトコル上にスキャナが識別されました。
13004	0	External UDP Scanner	外部ゾーンで UDP プロトコル上に単一スキャナを識別しました。
13004	1	External UDP Scanner	外部ゾーンで UDP プロトコル上にワーム攻撃を識別しました。UDP ヒストグラムのしきい値を超え、UDP プロトコル上にスキャナが識別されました。
13005	0	External Other Scanner	外部ゾーンで他のプロトコルを介して単一スキャナを識別しました。
13005	1	External Other Scanner	外部ゾーンでその他のプロトコル上にワーム攻撃を識別しました。その他のヒストグラムのしきい値を超え、その他のプロトコル上にスキャナが識別されました。
13006	0	Illegal TCP Scanner	不正ゾーンで TCP プロトコル上に単一スキャナを識別しました。
13006	1	Illegal TCP Scanner	不正ゾーンで TCP プロトコルを介してワーム攻撃を識別しました。TCP ヒストグラムのしきい値を超え、TCP プロトコルを介してスキャナが識別されました。
13007	0	Illegal UDP Scanner	不正ゾーンで UDP プロトコル上に単一スキャナを識別しました。
13007	1	Illegal UDP Scanner	不正ゾーンで UDP プロトコル上にワーム攻撃を識別しました。UDP ヒストグラムのしきい値を超え、UDP プロトコル上にスキャナが識別されました。
13008	0	Illegal Other Scanner	不正ゾーンでほかのプロトコルを介して単一スキャナを識別しました。
13008	1	Illegal Other Scanner	不正ゾーンでほかのプロトコルを介してワーム攻撃を識別しました。他のヒストグラムのしきい値を超え、他のプロトコルを介してスキャナが識別されました。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Traffic ICMP エンジン

Traffic ICMP エンジンは、TFN2K、LOKI、DDoS などの非標準プロトコルを分析します。このエンジンには、ユーザが設定可能なパラメータを持つ 2 つのシグニチャ（LOKI プロトコルに基づく）だけが含まれます。

TFN2K は、TFN の新しいバージョンです。TFN2K は DDoS エージェントの一種であり、感染した複数のコンピュータ（ゾンビ）による協調した攻撃（何百または何千もの未知の攻撃ホストから 1 つのコンピュータまたはドメインに向けて偽のトラフィック フラッドを送信する攻撃）を制御します。TFN2K はランダムに抽出されたパケット ヘッダー情報を送信しますが、それにはシグニチャの定義に使用できる 2 つの識別子が付いています。1 つは L3 チェックサムが不正かどうかを示し、もう 1 つはペイロードの末尾に文字 64「A」が検出されたかどうかを示します。TFN2K は、任意のポートで実行可能であり、ICMP、TCP、UDP、またはこれらのプロトコルの組み合わせを使用して通信できます。

LOKI は、バック ドア型トロイの木馬タイプです。コンピュータが感染すると、悪意のあるコードにより ICMP トンネルが作成されます。この ICMP トンネルは、ICMP 応答内での小さなペイロードの送信に使用されるおそれがあります（ICMP をブロックするように設定していないと、ICMP 応答はファイアウォールを通過することがあります）。LOKI シグニチャは、ICMP エコーの要求と応答のアンバランス、簡易 ICMP コード、およびペイロード識別子をモニタします。

（TFN2K を除く）DDOS カテゴリは、ICMP ベースの DDOS エージェントを対象とします。ここで使用する主なツールは、TFN と Stacheldraht です。これらは TFN2K と同様に動作しますが、ICMP だけに依存し、固定コマンド（整数および文字列）を備えています。

表 B-38 に、Traffic ICMP エンジンに固有のパラメータを示します。

表 B-38　Traffic ICMP エンジンのパラメータ

パラメータ	説明	値
parameter-tunable-sig	設定可能なパラメータがシグニチャに存在するかどうか。	[yes] | [no]
inspection-type	実行する検査のタイプ： • 最初の LOKI トラフィックを検査する。 • 変更された LOKI トラフィックを検査する。	is-loki is-mod-loki
reply-ratio	要求と応答のアンバランス。要求と比べて、応答が指定した数より多い場合に、アラートを起動します。	0 ～ 65535
want-request	アラートを起動する前に、ECHO REQUEST の検出が必要となります。	[true] | [false]

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。

Trojan エンジン

Trojan エンジンは、BO2K および TFN2K などの非標準プロトコルを分析します。Trojan エンジンには、Trojan BO2K、TrojanTFN2K、および Trojan UDP の 3 つがあります。

BO は、UDP 上のみで実行された最初の Windows のバック ドア型トロイの木馬です。これはまもなく、BO2K に更新されました。BO2K は、基本的な XOR で暗号化された UDP と TCP の両方に対応しています。これらには、特定のクロスパケット特性を持つプレーンな BO ヘッダーがあります。

また、BO2K には、BO ヘッダーを暗号化し、ほぼ認識できないクロスパケット パターンを作成するように設計された隠れた TCP モジュールもあります。UDP モードの BO および BO2K は、Trojan UDP エンジンによって処理されます。TCP モードは、Trojan BO2K エンジンによって処理されます。

（注）　Trojan UDP エンジンの swap-attacker-victim を除き、Trojan エンジンに固有のパラメータはありません。

詳細情報

すべてのシグニチャ エンジンに共通するパラメータの詳細については、「Master エンジン」（P.B-4）を参照してください。