シグニチャ エンジン

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。この章は、次の項で構成されています。

シグニチャ エンジンについて

シグニチャ エンジンは、特定のカテゴリの多数のシグニチャをサポートするように設計された 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）　5.1 エンジンは、標準化された正規表現をサポートします。

IPS 5.1 には次のシグニチャ エンジンが搭載されています。

- AIC：Web トラフィックの徹底的な分析を行います。

  HTTP プロトコルの不正利用を防止するために、HTTP セッションを精密に制御します。これは、インスタント メッセージや、gotomypc など、特定のポート上でトンネリングを試行するアプリケーションに対する管理制御を可能にします。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 FTP と AIC HTTP の 2 つの AIC エンジンがあります。

  AIC エンジン シグニチャを設定する方法の詳細については、P.7-15 の「AIC シグニチャの設定」を参照してください。
- Atomic：Atomic エンジンは、現在、2 つのエンジンが組み合され、マルチレベルで選択できます。たとえば、IP + TCP のように、レイヤ 3 およびレイヤ 4 のアトリビュートを組み合せて 1 つのシグニチャを作成できます。Atomic エンジンは標準化された正規表現をサポートします。
  - Atomic IP：IP プロトコル パケットと関連付けられたレイヤ 4 転送プロトコルを検査します。

    このエンジンでは、IP ヘッダーとレイヤ 4 ヘッダー内のフィールドと照合する値を入力し、正規表現を使用してレイヤ 4 ペイロードを検査できます。

    （注）　すべての IP パケットが Atomic IP エンジンに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 エンジンに取って代わります。

  - Atomic ARP：レイヤ 2 ARP プロトコルを検査します。ほとんどのエンジンはレイヤ 3 IP に基づいているため、Atomic ARP エンジンはその他の大半のエンジンとは異なっています。
- Flood：ホストおよびネットワークを宛先とする ICMP および UDP フラッドを検出します。

  Flood HOST と Flood NET の 2 つの Flood エンジンがあります。
- Meta：スライドする時間間隔内で、関連する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：複数の文字列を 1 つのシグニチャに対して照合し、レイヤ 4 転送プロトコルと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。

  （注）　Multi String エンジンは、IPS 5.1 からの新機能です。

- Normalizer：IP および TCP の正規化エンジンがどのように機能するかを設定し、IP および TCP の正規化エンジンに関連するシグニチャ イベントの設定を行います。RFC に準拠させることができます。

- 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 タイプがあります。
  - 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。
  - GENERIC：カスタム サービスとペイロードをデコードします。
  - H225：VoIP トラフィックを検査します。

    ネットワーク管理者は、VoIP ネットワークに入力された SETUP メッセージが有効であり、ポリシーの規定範囲内であることを確認できます。また、url-ids、email-ids、および表示情報などのアドレス フィールドや Q.931 文字列フィールドが固有の長さであり、考えられる攻撃パターンは含まれていないことを確認できます。
  - HTTP：HTTP トラフィックを検査します。

    WEBPORTS 変数が HTTP トラフィックの検査ポートを定義します。
  - IDENT：IDENT（クライアントとサーバ）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。
  - RPC：RPC トラフィックを検査します。
  - SMB：SMB トラフィックを検査します。
  - 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。
- State：SMTP などのプロトコル内の文字列をステートフル検索を行います。

  State エンジンには現在、非表示の設定ファイルがあります。これは、シグニチャ アップデートで新しい状態定義を配信できるように状態遷移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て正規表現文字列を検索します。

  String ICMP、String TCP、および String UDP の 3 つの String エンジンがあります。
- Sweep：1 つのホスト（ICMP および TCP）、宛先ポート（TCP および 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。
- Traffic ICMP：TFN2K、LOKI、および DDOS など、非標準の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 などの非標準のプロトコルからのトラフィックを分析します。

  Bo2k、Tfn2k、および UDP の 3 つの Trojan エンジンがあります。これらのエンジンには、ユーザ設定可能なパラメータはありません。

Master エンジン

Master エンジンは、その他のエンジンに構造とメソッドを提供し、設定からの入力とアラート出力を処理します。この項では、Master エンジンについて説明します。取り上げる事項は次のとおりです。

汎用パラメータ

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。

表 B-1 に、Master エンジンの汎用パラメータを示します。

表 B-1　Master エンジンの汎用パラメータ

パラメータ	説明	値
alert-severity	アラートの重大度： • 危険なアラート • 中レベルのアラート • 低レベルのアラート • 情報アラート	high medium low informational
engine	シグニチャが所属するエンジンを指定します。	—
event-counter	イベント カウント設定をグループ化します。	—
event-count	アラートを生成するまでのイベントの発生回数。	1 ～ 65535
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： • 攻撃者のアドレス • 攻撃者アドレスと被害先アドレス • 攻撃者アドレスと被害先ポート • 被害先のアドレス • 攻撃者と被害先のアドレスおよびポート	Axxx AxBx Axxb xxBx AaBb
specify-alert-interval	アラートの間隔をイネーブルにします。	yes | no
alert-interval	イベント カウントをリセットするまでの秒数。	2 ～ 1000
promisc-delta	アラートの重大度を決定するために使用されるデルタ値。	0 ～ 30
sig-fidelity-rating	このシグニチャの忠実度の評価。	0 ～ 100
sig-description	シグニチャの説明をグループ化します。	—
sig-name	シグニチャの名前。	*sig-name*
sig-string-info	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補足情報。	*sig-string-info*
sig-comment	このシグニチャに関するコメント。	*sig-comment*
alert-traits	このシグニチャについて文書化する特性。	0 ～ 65335
release	シグニチャが最近更新されたリリース。	*release*
status	シグニチャが使用可能か使用不可か、アクティブかリタイア（これ以上使用しない）かどうかを示します。	enabled retired

注意　シグニチャの promisc-delta 設定を変更することはお勧めしません。

混合モードでは、混合デルタは特定のアラートの RR より小さくなります。センサーはターゲット システムのアトリビュートを認識しませんが、混合モードではパケットを拒否できないため、混合アラートの優先順位を（優先順位の低いリスク評価より）低く設定しておくと役立ちます。そうすることで、管理者は優先順位の高いリスク評価アラートの調査に集中できます。

インライン モードでは、センサーが違反パケットを拒否することができます。その場合、違反パケットがターゲット ホストに到達することはないので、ターゲットが脆弱であっても問題になりません。攻撃はネットワーク上で許されなかったため、リスク評価値は下げません。

サービス、OS、およびアプリケーションに固有のもの以外のシグニチャの混合デルタは 0 です。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ものである場合は、5、10、または 15 の混合デルタがカテゴリごとに 5 つのポイントから計算されます。

アラートの頻度

アラート頻度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 の 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要約が開始されます。

表 B-2 にアラート頻度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	要約に使用するモード。	—
fire-all	すべてのイベントについてアラートを生成します。	—
fire-once	1 回だけアラートを生成します。	—
global-summarize	攻撃者や被害先の数に関係なく 1 回だけアラートが生成されるようにアラートを要約します。	—
summarize	アラートを要約します。	—
specify-summary-threshold	（オプション）サマリーのしきい値をイネーブルにします。	yes | no
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
specify-global-summary-threshold	グローバル サマリーのしきい値をイネーブルにします。	yes | no
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 サマリーに要約されます。	1 ～ 65535
summary-interval	各サマリー アラートを生成する間隔（秒数）。	1 ～ 1000

表 B-2　Master エンジンのアラート頻度パラメータ（続き）

パラメータ	説明	値
summary-key	シグニチャを要約するストレージ タイプ： • 攻撃者のアドレス • 攻撃者アドレスと被害先アドレス • 攻撃者アドレスと被害先ポート • 被害先のアドレス • 攻撃者と被害先のアドレスおよびポート	Axxx AxBx Axxb xxBx AaBb

イベント アクション

次のほとんどのイベント アクションは、特定のエンジンに特化していない限り、各シグニチャに属しています。

表 B-3 は、イベント アクションについて説明しています。

表 B-3　イベント アクション

イベント アクション名	説明
Deny Attacker Inline	（インライン モードのみ）指定された期間、攻撃者アドレスから発信されたこのパケットおよび将来のパケットを送信しません。[1] （注）これは、拒否アクションの中で最も重大です。これは、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を拒否します。拒否された攻撃者のエントリをすべてクリアするには、**Monitoring** > **Denied Attackers** > **Clear List** をクリックします。これによって、これらのアドレスのネットワークへの再接続が許可されます。手順については、P.6-23 の「拒否された攻撃者リストのモニタリングとクリア」を参照してください。
Deny Attacker Service Pair Inline	（インライン モードのみ）指定された期間、攻撃者アドレスと被害先ポートのペアで、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、攻撃者と被害先のアドレスのペアで、このパケットおよび将来のパケットを送信しません。 （注）拒否アクションの場合、指定された期間と拒否された攻撃者の最大数を指定するには、**Configuration** > **Event Action Rules** > **General Settings** をクリックします。手順については、P.6-21 の「汎用設定」を参照してください。
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 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。

表 B-3　イベント アクション（続き）

イベント アクション名	説明
Log Attacker Packets	攻撃者アドレスを含む IP ロギング パケットを開始します。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Log Pair Packets	攻撃者と被害先のアドレスのペアを含む IP ロギング パケットを開始します。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Log Victim Packets	被害先アドレスを含む IP ロギング パケットを開始します。
Modify Packet Inline	パケット データを変更して、エンドポイントでパケットがどう処理されるかに関してあいまいな部分を除去します。 （注）　Modify Packet Inline は、Add Event Action Filter または Add Event Action Override のオプションではありません。
Produce Alert	イベントをアラートとしてイベント ストアに書き込みます。
Produce Verbose Alert	違反パケットの符号化ダンプをアラートに組み込みます。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。
Request Block Connection	この接続をブロックする要求を ARC に送信します。 （注）　ブロッキング　デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。
Request Block Host	この攻撃者ホストをブロックする要求を ARC に送信します。 （注）　ブロッキング　デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。 （注）　ブロック　アクションの場合、ブロックの期間を設定するには、**Configuration > Event Action Rules > General Settings** をクリックします。手順については、第 6 章「汎用設定」を参照してください。

表 B-3　イベント アクション（続き）

イベント アクション名	説明
Request Rate Limit	レート制限を実行するレート制限要求を ARC に送信します。レート制限デバイスは、このアクションを実装するよう設定されている必要があります。詳細については、第 10 章「ブロッキングとレート制限のための ARC の設定」を参照してください。
Request SNMP Trap	SNMP 通知を実行する要求を NotificationApp に送信します。 （注）　このアクションを実行すると、Produce Alert が選択されていない場合でも、イベント ストアにアラートが書き込まれます。SNMP は、このアクションを実装するようセンサーで設定されている必要があります。詳細については、第 11 章「SNMP の設定」を参照してください。
Reset TCP Connection	TCP リセットを送信し、TCP フローを乗っ取って終了します。 （注）　Reset TCP Connection は、単一の接続を分析する TCP シグニチャでのみ機能します。スイープやフラッドに対しては機能しません。

1. センサーは、システムによって拒否されている攻撃者のリストを保持します。拒否された攻撃者のリストからエントリを削除するには、攻撃者のリストを表示してリスト全体をクリアするか、タイマーで有効期限が切れるのを待ちます。タイマーは、エントリごとにスライドするタイマーです。したがって、攻撃者 A が拒否されているときに別の攻撃が発行されると、攻撃者 A のタイマーはリセットされ、そのタイマーの有効期限が切れるまで攻撃者 A は拒否された攻撃者リストに残ります。拒否された攻撃者のリストがいっぱいになり、新規エントリを追加することができない場合でも、パケットは拒否されます。

注意　シグニチャに対してアラートをイネーブルにした場合、Produce Alert アクションは自動にはなりません。イベント ストアでアラートを作成するには、Produce Alert を選択する必要があります。2 番目のアクションを追加する場合、イベント ストアにアラートを送信するには、Produce Alert を組み込む必要があります。また、イベント アクションを設定するたびに、新規リストが作成され、古いリストが置換されます。各シグニチャに必要なイベント アクションを必ずすべて組み込んでください。

AIC エンジン

AIC エンジンは、HTTP Web トラフィックを検査し、FTP コマンドを実行します。この項では、AIC エンジンとそのパラメータについて説明します。取り上げる事項は次のとおりです。

概要

AIC エンジンは、Web トラフィックを詳細に検査するためのシグニチャを定義します。また、FTP コマンドの実行権限を持ちそれを実行するシグニチャも定義します。

AIC HTTP と AIC FTP の 2 つの AIC エンジンがあります。

AIC エンジンには、次の機能があります。

- Web トラフィック：
  - RFC 準拠の実施
  - HTTP 要求メソッドの許可と実施
  - 応答メッセージの検証
  - MIME タイプの指定
  - 転送符号化タイプの検証
  - 転送されるメッセージのコンテンツとデータのタイプに基づくコンテンツ制御
  - URI 長の指定
  - 設定されたポリシーとヘッダーに応じたメッセージ サイズの指定
  - トンネリング、P2P およびインスタント メッセージの実施

    この実施は、正規表現を使用して実行されます。事前定義済むのシグニチャが存在しますが、リストは拡張できます。
- FTP トラフィック：
  - FTP コマンドの許可と実施

AIC エンジンのパラメータ

AIC は Web トラフィックの徹底的な分析を行います。HTTP プロトコルの不正利用を防止するために、HTTP セッションを精密に制御します。これは、インスタント メッセージや、gotomypc など、特定のポート上でトンネリングを試行するアプリケーションに対する管理制御を可能にします。これらのアプリケーションが HTTP を介して稼働している場合は、P2P およびインスタント メッセージの検査とポリシー チェックを実行できます。

AIC は、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方法を提供します。

事前定義されたシグニチャをイネーブルまたはディセーブルにすることもできますし、カスタム シグニチャでポリシーを作成することもできます。

（注）　AIC エンジンは、HTTP トラフィックが AIC Web ポートで受信されたときに実行されます。トラフィックが Web トラフィックであっても、AIC Web ポートで受信されない場合は、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ている任意のポートで実行できます。検査されるトラフィックは HTTP トラフィックです。

AIC エンジン シグニチャの設定手順については、P.7-15 の「AIC シグニチャの設定」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、P.7-43 の「AIC MIME タイプ シグニチャ」を参照してください。

表 B-4 に、AIC HTTP エンジンに固有のパラメータを示します。

表 B-4　AIC HTTP エンジンのパラメータ

パラメータ	説明
signature-type	AIC シグニチャのタイプを指定します。
content-types	MIME タイプを処理する AIC シグニチャ： • define-content-type は、特定の MIME タイプ（image/gif）の拒否、メッセージサイズ違反の定義、ヘッダーと本体に記述された MIME タイプの不一致の判別といったアクションを関連付けます。 • define-recognized-content-types は、センサーが認識したコンテンツ タイプのリストを表示します。
define-web-traffic-policy	非準拠の HTTP トラフィックを検出した場合に実行するアクションを指定します。**alarm-on-non-http-traffic [true | false]** コマンドはシグニチャをイネーブルにします。このシグニチャはデフォルトではディセーブルです。
max-outstanding-requests-overrun	接続あたりの HTTP 要求の許容最大数（1 ～ 16）。
msg-body-pattern	正規表現を使用して、メッセージ本体内の特定のパターンを検索するシグニチャを定義します。
request-methods	アクションを HTTP 要求メソッドに関連付けることのできる AIC シグニチャ： • 取得、挿入などの define-request-method • recognized-request-methods　はセンサーが認識したメソッドのリストを表示します。
transfer-encodings	転送符号化を処理する AIC シグニチャ： • define-transfer-encoding は、圧縮、チャンクなどのアクションを各メソッドに関連付けます。 • recognized-transfer-encodings は、センサーが認識したメソッドのリストを表示します。 • chunked-transfer-encoding-error は、チャンク符号化エラーを検出した場合に実行するアクションを指定します。

表 B-5 に、AIC FTP エンジンに固有のパラメータを示します。

表 B-5　AIC FTP エンジンのパラメータ

パラメータ	説明
signature-type	AIC シグニチャのタイプを指定します。
ftp-commands	アクションを FTP コマンドに関連付けます。 • ftp-command：検査する FTP コマンドを選択します。
unrecognized-ftp-command	認識されない FTP コマンドを検査します。

Atomic エンジン

Atomic エンジンには、アラートの生成を引き起こす単純な単一のパケットの条件に対応するシグニチャが含まれます。この項では、Atomic エンジンについて説明します。取り上げる事項は次のとおりです。

Atomic ARP エンジン

Atomic ARP エンジンは、基本レイヤ 2 ARP シグニチャを定義し、ARP スプーフィング ツールの dsniff および ettercap の高度な検出も実行します。

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

表 B-6　Atomic ARP エンジンのパラメータ

パラメータ	説明
specify-mac-flip	この IP アドレスに対して MAC アドレスが指定した回数以上変化した場合にアラームを生成します。
specify-type-of-arp-sig	生成する ARP シグニチャのタイプを指定します。 • Source Broadcast（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ムを生成します。 • Destination Broadcast：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ムを生成します。 • Same Source and Destination：同一の送信元および宛先 MAC アドレスを持つ 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ムを生成します。 • Source Multicast：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ムを生成します。
specify-request-inbalance	IP アドレスに対する応答と比べて、要求が指定した数より多い場合にアラームを生成します。
specify-arp-operation	このシグニチャの ARP 演算コード。

Atomic IP エンジン

Atomic IP エンジンは、IP プロトコル ヘッダー、および関連付けられたレイヤ 4 転送プロトコル（TCP、UDP、および ICMP）とペイロードを検査するシグニチャを定義します。

（注）　Atomic エンジンは複数のパケットにまたがって固定データを保存しません。その代わりに、1 つのパケットの解析からアラートを発生できます。

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
fragment-status	フラグメント化するかどうかを指定します。
specify-ip-payload-length	IP データグラムのペイロード長を指定します。
specify-ip-header-length	IP データグラムのヘッダー長を指定します。
specify-ip-addr-options	IP アドレスを指定します。
specify-ip-id	IP 識別子を指定します。
specify-ip-total-length	IP データグラムの合計の長さを指定します。
specify-ip-option-inspection	IP オプションの検査を指定します。
specify-l4-protocol	レイヤ 4 プロトコルを指定します。
specify-ip-tos	サーバのタイプを指定します。
specify-ip-ttl	存続可能時間を指定します。
specify-ip-version	IP プロトコルのバージョンを指定します。

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監視するシグニチャを定義します。たとえば、1 秒あたりに（特定タイプの）150 以上のパケットが被害先のホストに送信されていることを検出した場合に生成されるシグニチャを作成できます。

Flood Host と Flood Net の 2 つの Flood エンジンがあります。

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

表 B-8　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	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 プロトコルを選択した場合の送信元ポートを指定します。	0 ～ 65535[3] a-b[,c-d]

1. 秒あたりのパケット数よりもレートが大きくなると、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-9 に、Flood Net エンジンに固有のパラメータを示します。

表 B-9　Flood Net エンジンのパラメータ

パラメータ	説明	値
gap	フラッド シグニチャの許容ギャップ時間（秒単位）。	0 ～ 65535
peaks	フラッド トラフィックの許容ピーク回数。	0 ～ 65535
protocol	検査するトラフィックの種類。	ICMP TCP UDP
rate	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. 秒あたりのパケット数よりもレートが大きくなると、アラートが起動します。

Meta エンジン

Meta エンジンは、スライドする時間間隔内で、関連する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。シグニチャ イベントが生成されると、Meta エンジンがそれらを検査して、1 つまたはいくつかの Meta 定義と一致するかどうかを判別します。Meta エンジンは、このイベントに対するすべての要件が満たされた後に、シグニチャ イベントを生成します。

シグニチャ イベントはすべて SEAP によって Meta エンジンに渡されます。SEAP は、minimum hits オプションを処理した後で、イベントを渡します。要約とイベント アクションは、Meta エンジンがコンポーネント イベントを処理した後に処理されます。SEAP の詳細については、P.6-2 の「Signature Event Action Processor」を参照してください。

注意　多数の Meta シグニチャが、意図せずセンサー パフォーマンス全体に影響を及ぼす可能性があります。

表 B-10 に、Meta エンジンに固有のパラメータを示します。

表 B-10　Meta エンジンのパラメータ

パラメータ	説明	値
meta-reset-interval	META シグニチャをリセットする時間間隔（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：リストに新しいエントリを挿入します。 　– 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 リストの終わりに配置します。 　– inactive:エントリを非アクティブ リストに入れます。 　– before：エントリを指定したエントリの前に配置します。 　– after：エントリを指定したエントリの後ろに配置します。 • move：リスト内のエントリを移動します。	*name1*
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 • 攻撃者アドレスと被害先アドレス • 攻撃者と被害先のアドレスおよびポート • 被害先のアドレス	AaBb AxBx Axxx xxBx
unique-victim-ports	Meta シグニチャごとに一意の必須被害先ポートの番号。	1 ～ 256
component-list-in-order	コンポーネント リストを順序立てて生成するかどうかを指定します。	true | false

カスタム Meta エンジン シグニチャの例については、P.7-39 の「MEG シグニチャの例」を参照してください。

Multi String エンジン

Multi String エンジンでは、1 つのシグニチャに対して複数の文字列を照合することでレイヤ 4 転送プロトコル（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。シグニチャを生成するために一致している必要の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の方向で実行されます。

1 つ以上の正規表現パターンを指定する必要がある場合に Multi String エンジンを使用します。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、該当するプロトコル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-11 に、Multi String エンジンに固有のパラメータを示します。

表 B-11　Multi String エンジンのパラメータ

パラメータ	説明	値
inspect-length	生成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のあるストリームまたはパケットの長さ。	0 ～ 4294967295
protocol	レイヤ 4 プロトコル選択。	icmp tcp udp
regex-component	正規表現コンポーネントのリスト： • regex-string：検索文字列。 • spacing-type：前回の一致か所から、またはそれが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スペーシングのタイプ。	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最小バイト数。	0 ～ 4294967296
swap-attacker-victim	アドレス（およびポート）の送信元と宛先がアラートメッセージでスワップされる場合は、true。スワップされない場合は 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、source-ports 値が 80 の場合、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。

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。

Normalizer エンジン

Normalizer 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。この項では、Normalizer エンジンについて説明します。取り上げる事項は次のとおりです。

概要

Normalizer エンジンは、IP フラグメント再構成および TCP ストリーム再構成を処理します。Normalizer エンジンでは、たとえば、センサーが同時に追跡するフラグメントの最大数など、システム リソースの使用状況に関する制限を設定できます。

（注）　Normalizer エンジンに、カスタム シグニチャを追加することはできません。既存のシグニチャはチューニングできます。

- IP フラグメンテーションの正規化

  意図的または意図しない IP データグラムのフラグメンテーションは、不正利用を隠し、それらの検出を困難または不可能にしてしまうことがあります。フラグメント化は、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することもできます。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メソッドを使用して、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送出したりします。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方法をすべてチェックする必要がある場合は、センサーが DoS 攻撃（サービス拒絶攻撃）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 フラグメント化の正規化の装置は、この機能を実行します。

- TCP 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非表示になることもあります。ポリシーが false positive や false negative 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ないはずです。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 スタックの実装が破損しています。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。センサーが 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、正規化エンジンによって回避や攻撃に関係する異常なパケットが検索されます。

混合モードのセンサーは、違反を検出すると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、modify-packet-inline などの event-action パラメータに指定したアクションを実行します。

Normalizer エンジン シグニチャの設定手順については、P.7-23 の「IP フラグメント再構成」および P.7-26 の「TCP ストリーム再構成の設定」を参照してください。

Normalizer エンジンのパラメータ

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

表 B-12　Normalizer 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	（オプション）フラグメント再構成タイムアウトをイネーブルにします。
specify-hijack-max-old-ack	（オプション）hijack-max-old-ack をイネーブルにします。
specify-max-dgram-size	（オプション）最大データグラム サイズをイネーブルにします。
specify-max-fragments	（オプション）最大フラグメントをイネーブルにします。
specify-max-fragments-per-dgram	（オプション）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	（オプション）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	（オプション）最大部分データグラムをイネーブルにします。
specify-max-small-frags	（オプション）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	（オプション）最小フラグメント サイズをイネーブルにします。
specify-service-ports	（オプション）サービス ポートをイネーブルにします。
specify-syn-flood-max-embryonic	（オプション）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	（オプション）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	（オプション）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	（オプション）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	（オプション）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	（オプション）TCP 最大キューをイネーブルにします。
specify-tcp-min-mss	（オプション）TCP 最小 mss をイネーブルにします。
specify-tcp-option-number	（オプション）TCP オプション番号をイネーブルにします。

Service エンジン

Service エンジンは、2 つのホスト間で L5+ トラフィックを解析します。これらは、固定データを追跡する 1 対 1 シグニチャです。このエンジンは、ライブ サービスに似た方法で L5+ ペイロードを解析します。

Service エンジンには共通の特性があります。ただし、各エンジンには、検査対象のサービスに関する固有の情報があります。Service エンジンは、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、アルゴリズム向けの汎用文字列エンジンの機能を補足します。

この項では、次のトピックについて説明します。

Service DNS エンジン

Service DNS エンジンは高度な DNS デコード専用です。これには次の複数のジャンプのような反回避技術が含まれています。長さ、命令コード、文字列などの多数のパラメータがあります。Service DNS エンジンは、TCP ポート 53 と UDP ポート 53 の両方で稼働し、2 つのプロトコルに対応するインスペクタです。TCP ではストリーム、UDP ではクワッドが使用されます。

表 B-13 に、Service DNS エンジンに固有のパラメータを示します。

表 B-13　Service DNS エンジンのパラメータ

パラメータ	説明	値
protocol	インスペクタの対象プロトコル。	TCP UDP
specify-query-chaos-string	（オプション）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*
specify-query-class	（オプション）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	（オプション）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 クエリ長	true | false

表 B-13　Service DNS エンジンのパラメータ（続き）

パラメータ	説明	値
specify-query-jump-count-exceeded	（オプション）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	（オプション）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	（オプション）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	（オプション）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 データ長	0 ～ 65535
specify-query-src-port-53	（オプション）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 パケットの送信元ポート 53	true | false
specify-query-stream-len	（オプション）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	（オプション）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	（オプション）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

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** コマンドと 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、**port** コマンドのデコードで、さまざまなエラー トラップ条件に割り当てられるブール値です。Service FTP エンジンは TCP ポート 20 および 21 で稼働します。ポート 20 はデータ用で、Service FTP エンジンはこのポートを検査しません。ポート 21 の制御トランザクションを検査します。

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

表 B-14　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 ポートに向かうトラフィック。 • クライアント ポートからサービス ポートに向かうトラフィック。	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 pasv
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	アドレス（およびポート）の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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。

Service Generic エンジン

Service Generic エンジンでは、設定ファイルのみのシグニチャ アップデートで、プログラム シグニチャを発行できます。設定ファイルで定義された簡易マシンおよびアセンブリ言語を持っています。仮想マシンを通じて（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 コードで指定された比較および演算を実行します。

これは、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。

（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 Service Generic エンジンのチューニングは、熟練したユーザのみが行ってください。

表 B-15 に、Service Generic エンジンに固有のパラメータを示します。

表 B-15　Service Generic エンジンのパラメータ

パラメータ	説明	値
specify-dst-port	（オプション）宛先ポートをイネーブルにします。 • dst-port：シグニチャの対象宛先ポート。	0 ～ 65535
specify-ip-protocol	（オプション）IP プロトコルをイネーブルにします。 • ip-protocol：インスペクタが検査する IP プロトコル。	0 ～ 255
specify-payload-source	（オプション）ペイロード送信元の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-data
specify-src-port	（オプション）送信元ポートをイネーブルにします。 • src-port：シグニチャの対象送信元ポート。	0 ～ 65535

Service H225 エンジン

この項では、Service H225 エンジンについて説明します。取り上げる事項は次のとおりです。

- 概要（P.B-21）
- Service H255 エンジンのパラメータ（P.B-22）

概要

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から構成されており、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 シグナリングとステータス 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の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 スタックの実装を稼働しています。Service H225 エンジンは、H.255.0 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への攻撃を検出します。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パケットを詳細に検査します。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。このアドレスは、制御プロトコル（H.245）チャネルの確立に使用されます。

SETUP コール シグナリング メッセージは特に重要です。これが、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となるからです。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チェックし、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法です。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 長さシグニチャをチューニングし、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 検証と長さチェック
- Q.931 情報要素の検証
- Q.931 情報要素のテキスト 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 ASN.1 シグニチャの数は固定です。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のみ変更する必要があります。ASN.1 の場合、パラメータは一切変更しないでください。Q.931 シグニチャでは、テキスト フィールドの新しい正規表現シグニチャを追加できます。SETUP シグニチャの場合は、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チェックするためのシグニチャを追加できます。

Service H255 エンジンのパラメータ

表 B-16 に、Service H225 エンジンに固有のパラメータを示します。

表 B-16　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

表 B-16　Service H.225 エンジンのパラメータ（続き）

パラメータ	説明	値
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。 • フィールドの検証を検査する。 • 値を検査する。 TPKT シグニチャの場合は regex と presence は有効な値ではありません。	length presence regex validate value
specify-field-name	（オプション）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 タイプの場合のみ有効です。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	（オプション）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索引をイネーブルにします。 • invalid-packet-index：無効なパケット索引を検査します。	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。 • 単一の TCP パケット内で検索する正規表現。 • （オプション）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。TPKT シグニチャには設定しないでください。	regex-string specify-min-match-length
specify-value-range	length または value ポリシー タイプの場合に有効です（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

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。

Service HTTP エンジン

この項では、Service HTTP エンジンについて説明します。取り上げる事項は次のとおりです。

概要

Service HTTP エンジンは、サービス固有で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よく使用されているプロトコルです。また、これには最も長い事前処理時間が必要であり、検査を必要とする最大数のシグニチャを持つため、システムのパフォーマンス全体にとって重大となります。

Service HTTP エンジンは、複数のパターンを結合して単一のパターン マッチング テーブルにし、単一のデータ検索を可能にする、正規表現ライブラリを使用します。このエンジンは、Web サービスだけに送られるトラフィック、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。このエンジンの各シグニチャで、該当する個別の Web ポートを指定できます。

HTTP 解読は、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。これは、ASCII 正規化とも呼ばれます。

HTTP パケットを検査するには、まずデータを、ターゲット システムでデータの処理時に表示されるものと同じ表記に解読または正規化する必要があります。どのオペレーティング システムおよび Web サーバ バージョンがターゲットで動作しているかを認識している、カスタマイズされたデコード技術をホスト ターゲット タイプごとに用意することをお勧めします。Service HTTP エンジンには、Microsoft IIS Web サーバに対するデフォルトの解読動作があります。

Service HTTP エンジンのパラメータ

カスタム Service HTTP シグニチャの例は、P.7-36 の「Service HTTP シグニチャの例」を参照してください。

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

表 B-17　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。	true | false
max-field-sizes	最大フィールド サイズ グループ。	—
specify-max-arg-field-length	（オプション）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	（オプション）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	（オプション）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

表 B-17　Service HTTP エンジンのパラメータ（続き）

パラメータ	説明	値
specify-max-uri-field-length	（オプション）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	（オプション）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（Content-Length で定義されているとおり、? の後ろのエンティティ本体の中）で検索する正規表現	—
specify-header-regex	（オプション）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-regex : HTTP 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	（オプション）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-regex : HTTP URI および HTTP 引数の両方のフィールド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	（オプション）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
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	アドレス（およびポート）の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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。

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を提供します。

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

表 B-18　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-bad-port	不良ポートのペイロードを検査します。	true | false
has-newline	ペイロードの非終端改行文字を検査します。	true | false
size	指定値より長いペイロード長を検査します。	0 ～ 65535
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　ポートに向かうトラフィック。 • クライアント　ポートからサービス　ポートに向かうトラフィック。	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。

Service MSRPC エンジン

この項では、Service MSRPC エンジンについて説明します。取り上げる事項は次のとおりです。

- 概要（P.B-26）
- Service MSRPC エンジンのパラメータ（P.B-27）

概要

Service MSRPC エンジンは、MSRPC パケットを処理します。MSRPC によって、ネットワーク環境内の複数のコンポーネント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、チャネルを確立するために一連の通信が行われ、処理要求と応答が渡されます。

MSRPC は、ISO レイヤ 5 ～ 6 プロトコルで、UDP、TCP、SMB などのその他の転送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に対応したファシリティ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 のセキュリティ上の弱点の原因です。

Service MSRPC エンジンは、最も一般的なトランザクション タイプに対応し DCE および RPC プロトコルだけをデコードします。

Service MSRPC エンジンのパラメータ

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

表 B-19　Service MSRPC エンジンのパラメータ

パラメータ	説明	値
protocol	インスペクタの対象プロトコル。	tcp udp
specify-operation	（オプション）MSRPC 動作の使用をイネーブルにします。 • operation：要求された MSRPC 動作。SMB_COM_TRANSACTION コマンドに必要です。完全一致。	0 ～ 65535
specify-regex-string	（オプション）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のある実際のストリーム 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のあるバイトの最小数。	0 ～ 65535
specify-uuid	（オプション）UUID をイネーブルにします。 • uuid：MSRPC UUID フィールド	000001a000000000 c000000000000046

Service MSSQL エンジン

Service MSSQL エンジンは、Microsoft の SQL server（MSSQL）によって使用されるプロトコルを検査します。

1 つの MSSQL シグニチャがあります。デフォルトの sa アカウントを使用した MSSQL へのログイン試行を検出すると、アラートを生成します。

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

表 B-20　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	（オプション）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中のユーザのユーザ名（完全一致）	sa

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。1 つの NTP シグニチャ（NTPd readvar オーバーフロー シグニチャ）があります。これは、readvar コマンドに NTP サービスが取り込むには大き過ぎる NTP データが指定されていることを検出した場合にアラートを生成します。

モードや制御パケットのサイズなどの NTP プロトコルの値に基づいて、シグニチャのチューニングやカスタム シグニチャの作成を行えます。

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

表 B-21　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	Inspects NTP パケット： • control-opcode：RFC1305、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

Service RPC エンジン

Service RPC エンジンは、RPC プロトコル専用で、反回避方式としてフル デコードが含まれています。この方式によって、断片化メッセージ（複数のパケット内の 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のメッセージ）を処理できます。

RPC ポートマッパーは、ポート 111 で動作します。正規の RPC メッセージは番号が 551 以上の任意のポートでやり取りできます。RPC スイープは、有効な RPC メッセージが送信されたときに一意のポートだけをカウントする点を除けば、TCP ポート スイープと同様です。RPC は UDP 上でも稼働します。

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

表 B-22　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 ポートに向かうトラフィック。 • クライアント ポートからサービス ポートに向かうトラフィック。	from-service to-service
protocol	対象のプロトコル。	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]

表 B-22　Service RPC エンジンのパラメータ（続き）

パラメータ	説明	値
specify-is-spoof-src	（オプション）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生成します。	true | false
specify-port-map-program	（オプション）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：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	（オプション）RPC 最小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生成します。	0 ～ 65535
specify-rpc-procedure	（オプション）IP プロシージャをイネーブルにします。 • rpc-procedure：シグニチャの対象 RPC プロシージャ番号	0 ～ 1000000
specify-rpc-program	（オプション）IP プログラムをイネーブルにします。 • rpc-program：シグニチャの RPC プログラム番号	0 ～ 1000000

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。

Service SMB エンジン

Service SMB エンジンは、SMB パケットを検査します。SMB 制御トランザクション交換および SMB NT_Create_AndX 交換に基づいて、SMB シグニチャのチューニングやカスタム SMB シグニチャの作成を行えます。

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

表 B-23　Service SMB 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-allocation-hint	（オプション）MSRPC 割り当てのヒントをイネーブルにします。 • allocation-hint：MSRPC 割り当てヒント。SMB_COM_TRANSACTION コマンドの解析で使用されます。[2]	0 ～ 42949677295
specify-byte-count	（オプション）バイト カウントをイネーブルにします。 • byte-count：SMB_COM_TRANSACTION 構造からのバイト カウント[3]	0 ～ 65535
specify-command	（オプション）SMB コマンドをイネーブルにします。 • command：SMB コマンド値[4]	0 ～ 255

表 B-23　Service SMB エンジンのパラメータ（続き）

パラメータ	説明	値
specify-direction	（オプション）トラフィックの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します。 － サービス ポートからクライアント ポートに向かうトラフィック。 － クライアント ポートからサービス ポートに向かうトラフィック。	from-service to service
specify-file-id	（オプション）トランザクション ファイル ID の使用をイネーブルにします。 • file-id：トランザクション ファイル ID[5] （注）このパラメータはシグニチャを特定の不正利用インスタンスだけに制限することがあるため、使用する場合は慎重に検討する必要があります。	0 ～ 65535
specify-function	（オプション）名前付きパイプ機能をイネーブルにします。 • function：名前付きパイプ機能[6]	0 ～ 65535
specify-hit-count	（オプション）ヒット カウントをイネーブルにします。 • hit-count：スキャン間隔の間の発生数のしきい値。この値を超えるとアラートを生成します。[7]	0 ～ 65535
specify-operation	（オプション）MSRPC 動作をイネーブルにします。 • operation：要求された MSRPC 動作。SMB_COM_TRANSACTION コマンドに必要です。完全一致は必須です。	0 ～ 65535
specify-resource	（オプション）リソースをイネーブルにします。 • resource：アラートを制限するために使用するパイプまたは SMB ファイル名を指定します。ASCII 形式です。完全一致は必須です。	*resource*
specify-scan-interval	（オプション）トラフィックの方向をイネーブルにします。 • scan-interval：アラート率を計算するために使用する間隔（秒単位）[8]	0 ～ 131071
specify-set-count	（オプション）セットアップ ワードのカウントをイネーブルにします。 • set-count：セットアップ ワードの数[9]	0 ～ 255
specify-type	（オプション）MSRPC パケットのタイプ フィールドの検索をイネーブルにします。 • type：MSRPC パケットのタイプ フィールド。0 = Request、2 = Response、11 = Bind、12 = Bind Ack	0 ～ 255
specify-word-count	（オプション）コマンド パラメータのワード カウントをイネーブルにします。 • word-count：SMB_COM_TRANSACTION コマンド パラメータのワード カウント[10]	0 ～ 255

表 B-23　Service SMB エンジンのパラメータ（続き）

パラメータ	説明	値
swap-attacker-victim	アドレス（およびポート）の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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. 完全一致はオプションです。
3. 完全一致はオプションです。
4. 完全一致は必須です。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。
5. 完全一致はオプションです。
6. 完全一致は必須です。SMB COM TRANSACTION コマンドに必要です。
7. シグニチャ 3302 および 6255 に対してのみ有効です。
8. シグニチャ 3302 および 6255 に対してのみ有効です。
9. 完全一致は必須です。通常、SMB COM TRANSACTION コマンドの場合は 2 個必要です。
10. 完全一致は必須です。16 個のワードのトランザクションだけがデコードされます。

Service SNMP エンジン

Service SNMP エンジンは、ポート 161 宛てのすべての 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 シグニチャのチューニングやカスタム SNMP シグニチャの作成を行えます。

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

表 B-24　Service SNMP 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。	—
brute-force-inspection	総当たり攻撃を検査します。 • 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 宛ての非 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、つまり 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*

Service SSH エンジン

Service SSH エンジンはポート 22 の SSH トラフィック専用です。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、エンジンはセットアップのフィールドのみを監視します。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はチューニングできますが、カスタム シグニチャは作成できません。

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

表 B-25　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 鍵の長さ： – length：鍵がこれより大きくなると、RSAREF オーバーフローが発生します。 • user-length：ユーザ長 SSH 検査： – length：鍵がこれより大きくなると、RSAREF オーバーフローが発生します。	0 ～ 65535
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	（オプション）パケット数をイネーブルにします。 • packet-depth：セッション鍵が失われたと判断するまでに監視するパケット数	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。

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、出力またはアラーム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-26 に、State エンジンに固有のパラメータを示します。

表 B-26　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。	—
cisco-login	Cisco login のステート マシンを指定します。 • state-name：シグニチャがアラートを起動する前に必要な状態の名前。 — 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 Format String の弱点を検査するステート マシンを指定します。 • state-name：シグニチャがアラートを起動する前に必要な状態の名前。 — LPR Format String 検査を終了させる中断状態 — 形式文字の状態 — 開始状態	abort format-char start
smtp	SMTP プロトコルのステート マシンを指定します。 • state-name：シグニチャがアラートを起動する前に必要な状態の名前。 — LPR Format String 検査を終了させる中断状態 — 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　ポートに向かうトラフィック。 • クライアント　ポートからサービス　ポートに向かうトラフィック。	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 a-b[,c-d] [1]

表 B-26　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	（オプション）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のある実際のストリーム オフセット。	0 ～ 65535
specify-min-match-length	（オプション）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のあるバイトの最小数。	0 ～ 65535
swap-attacker-victim	アドレス(およびポート)の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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。

String エンジン

この項では、String エンジンについて説明します。取り上げる事項は次のとおりです。

概要

String エンジンは、ICMP、TCP、および UDP の各プロトコル用の汎用のパターン マッチング検査エンジンです。String エンジンは、複数のパターンを結合して単一のパターン マッチング テーブルにし、単一のデータ検索を可能にする、正規表現エンジンを使用します。

String ICMP、String TCP、および String UDP の 3 つの String エンジンがあります。

String ICMP エンジンのパラメータ

カスタム String エンジン シグニチャの例は、P.7-32 の「String TCP シグニチャの例」を参照してください。

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

表 B-27　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 ポートに向かうトラフィック。 • クライアント ポートからサービス ポートに向かうトラフィック。	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	（オプション）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のある実際のストリーム オフセット。	0 ～ 65535
specify-min-match-length	（オプション）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のあるバイトの最小数。	0 ～ 65535
swap-attacker-victim	アドレス（およびポート）の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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。

String TPC エンジンのパラメータ

カスタム String エンジン シグニチャの例は、P.7-32 の「String TCP シグニチャの例」を参照してください。

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

表 B-28　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 ポートに向かうトラフィック。 • クライアント ポートからサービス ポートに向かうトラフィック。	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	（オプション）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のある実際のストリーム オフセット。	0 ～ 65535
specify-min-match-length	（オプション）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のあるバイトの最小数。	0 ～ 65535
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	アドレス（およびポート）の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は 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。

String UDP エンジンのパラメータ

カスタム String エンジン シグニチャの例は、P.7-32 の「String TCP シグニチャの例」を参照してください。

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

表 B-29　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 ポートに向かうトラフィック。 • クライアント ポートからサービス ポートに向かうトラフィック。	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-29　String UDP エンジン（続き）

パラメータ	説明	値
specify-exact-match-offset	（オプション）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のある実際のストリーム オフセット。	0 ～ 65535
specify-min-match-length	（オプション）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のあるバイトの最小数。	0 ～ 65535
swap-attacker-victim	アドレス（およびポート）の送信元と宛先がアラートメッセージでスワップされる場合は、true。スワップされない場合は 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。

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解析します。既存のシグニチャをチューニングするか、またはカスタム シグニチャを作成できます。Sweep エンジンには、ICMP、UDP、および TCP 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、一定の間隔内にアドレス 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 トラッキング処理はカウンティングと呼ばれます。

一意のパラメータは、Sweep エンジンのすべてのシグニチャに対して指定する必要があります。スイープに適用される制限は 2 ～ 40（包含）です。2 はスイープの絶対最小数です。2 未満の場合は、スイープではありません（1 つのホストまたはポート）。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値範囲は、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 スイープでは、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。

表 B-30 に、Sweep エンジンに固有のパラメータを示します。

表 B-30　Sweep エンジンのパラメータ

パラメータ	説明	値
protocol	インスペクタの対象プロトコル。	icmp udp tcp
specify-icmp-type	（オプション）ICMP ヘッダー タイプをイネーブルにします。 • icmp-type：ICMP ヘッダーの TYPE 値。	0 ～ 255
specify-port-range	（オプション）検査対象としてのポート範囲の使用をイネーブルにします。 • port-range：検査に UDP ポート範囲を使用します。	0 ～ 65535 a-b[,c-d]
fragment-status	フラグメント化するかどうかを指定します。 • 任意のフラグメント状態。 • 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

表 B-30　Sweep エンジンのパラメータ（続き）

パラメータ	説明	値
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 • 攻撃者アドレスと被害先アドレス • 攻撃者アドレスと被害先ポート	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、アラートを生成しない。	true | false
swap-attacker-victim	アドレス(およびポート)の送信元と宛先がアラート メッセージでスワップされる場合は、true。スワップされない場合は false（デフォルト）。	true | false
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

Traffic ICMP エンジン

Traffic ICMP エンジンは、TFN2K、LOKI、および DDOS などの非標準プロトコルを解析します。ユーザ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）のみがあります。

Tribe Flood Net 2000（TFN2K）は比較的新しいバージョンの TFN です。これは感染したコンピュータ（ゾンビ）による調整された攻撃を制御して、無数の未確認の攻撃ホストから偽のトラフィックフラッドで 1 つのコンピュータ（またはドメイン）をターゲットとするために使用される Distributed DOS（DDoS）エージェントです。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キャラクタ 64「A」が検出されたかどうかを示しています。TFN2K は、任意のポートで実行可能で、ICMP、TCP、UDP、またはこれらのプロトコルの組み合せで通信できます。

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、悪意のあるコードが ICMP 応答で小さなペイロードを送信するために使用できる「ICMP トンネル」を作成します（ICMP をブロックするように設定していない場合、この応答はファイアウォールを通過できます）。LOKI シグニチャは、応答に対する ICMP エコー要求のアンバランス、簡易 ICMP コードおよびペイロード識別子を監視します。

（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（Tribe Flood Net）と Stacheldraht です。これらは TFN2K と同様に動作しますが、ICMP のみに依存し、固定コマンド（整数および文字列）があります。

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

表 B-31　TRAFFIC.ICMP エンジンのパラメータ

パラメータ	説明	値
parameter-tunable-sig	シグニチャに設定可能なパラメータがあるかどうかを示します。	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	要求に対する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、アラートを生成します。	0 ～ 65535
want-request	アラートの生成前に、ECHO REQUEST の検出が必要となります。	true | false

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解析します。Trojan BO2K、Trojan TFN2K、および Trojan UDP の 3 つの Trojan エンジンがあります。

BackOrifice（BO）は、UDP 上でのみ実行された最初の Windows のバック ドア型トロイの木馬でした。すぐに BackOrifice 2000（BO2K）がそれに取って代わりました。BO2K は、基本的な XOR 暗号化された UDP および TCP の両方に対応しています。それらには、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、ほとんど認知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。

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