DCE SMB インスペクタ

- DCE SMB インスペクタの概要 （1 ページ）
- DCE SMB インスペクタのパラメータ （4 ページ）
- DCE SMB インスペクタのルール （8 ページ）
- DCE インスペクタの侵入ルールのオプション （11 ページ）

DCE SMB インスペクタの概要

タイプ	インスペクタ（サービス）
使用方法	検査
インスタンス タイプ	マルチトン
その他のインスペクタが必要	なし
有効	`true`

DCE/RPC プロトコルにより、別々のネットワーク ホスト上のプロセスが、同一ホストに配置されている場合と同様に通信できます。通常、このようなプロセス間通信はホスト間で TCP および UDP 経由で転送されます。TCP 転送では、DCE/RPC が Windows Server Message Block（SMB）プロトコルまたは Samba でさらにカプセル化されることがあります。Samba は、Windows や UNIX または Linux のオペレーティングシステムから構成される混合環境でプロセス間通信に使用されるオープンソースの SMB 実装です。

ほとんどの DCE/RPC エクスプロイトは、DCE/RPC サーバ（ネットワーク上の Windows または Samba が稼働している任意のホスト）を対象とした DCE/RPC クライアント要求で発生します。またエクスプロイトはサーバ応答でも発生することがあります。

IP によりすべての DCE/RPC トランスポートがカプセル化されます。TCP は、SMB など、すべてのコネクション型 DCE/RPC を伝送します。

`dce_smb` インスペクタは、SMB プロトコルでコネクション型 DCE/RPC を検出し、ヘッダー長やデータフラグメントの順序などのプロトコル固有の特性を使用して、次のことを行います。

- SMB トランスポートでカプセル化されている DCE/RPC 要求と応答を検出します。
- DCE/RPC データストリームを分析し、DCE/RPC トラフィック内の異常な動作と回避技術を検出します。
- SMB データストリームを分析し、異常な SMB 動作と回避技術を検出します。
- SMB のセグメント化を解除し、DCE/RPC をデフラグします。
- ルールエンジンで処理できるように DCE/RPC トラフィックを正規化します。

次の図に、DCE SMB インスペクタが SMB トランスポートのためにトラフィックの処理を開始するポイントを示します。

dce_smb インスペクタは通常、NetBIOS セッションサービス用のウェルノウン TCP ポート 139 か、または同様に実装されているウェルノウン Windows ポート 445 で SMB トラフィックを受信します。SMB には DCE/RPC の伝送以外にも多数の機能があるため、インスペクタは SMB トラフィックが DCE/RPC トラフィックを伝送しているかどうかをまずテストします。伝送していない場合は処理を停止し、伝送している場合は処理を続行します。

dce_smb インスペクタのパラメータと機能の説明には、Microsoft Remote Procedure Call（MSRPC）と呼ばれる DCE/RPC の Microsoft の実装と、SMB および Samba の両方が含まれています。

ターゲットベースのポリシー

Windows および Samba の DCE/RPC の実装は大きく異なります。たとえば、Windows のすべてのバージョンは、DCE/RPC トラフィックの最適化時に最初のフラグメントの DCE/RPC コンテキスト ID を使用しますが、Samba のすべてのバージョンは、最後のフラグメントのコンテキスト ID を使用します。また、特定の関数呼び出しを識別するために、Windows Vista では最初のフラグメントの opnum（操作番号）ヘッダーフィールドを使用しますが、Samba とその他のすべてのバージョンの Windows では最後のフラグメントの opnum フィールドを使用します。

Windows と Samba の SMB の実装には大きな違いがあります。たとえば、Windows は名前付きパイプの操作時に SMB OPEN および READ コマンドを認識しますが、Samba はこれらのコマンドを認識しません。

そのため、dce_smb インスペクタはターゲットベースのアプローチを使用します。dce_smb インスペクタインスタンスを設定すると、policy パラメータは DCE/RPC SMB プロトコルの実装を指定します。これをホスト情報と組み合わせることで、デフォルトのターゲットベースのサーバーポリシーが確立されます。必要に応じて、他のホストおよび DCE/RPC SMB の実装を対象とする追加のインスペクタを設定できます。デフォルトのターゲットベースのサーバーポリシーで指定されている DCE/RPC SMB の実装は、別の dce_smb インスペクタインスタンスの対象になっていないすべてのホストに適用されます。

dce_smb インスペクタが policy パラメータを使用して対象にできる DCE/RPC SMB の実装は次のとおりです。

- WinXP（デフォルト）
- Win2000
- WinVista
- Win2003
- Win2008
- Win7
- Samba
- Samba-3.0.37
- Samba-3.0.22
- Samba-3.0.20

ファイル検査

dce_smb インスペクタは、SMB バージョン 1、2、および 3 のファイル検査をサポートします。

dce_smb インスペクタは、通常の SMB ファイル転送を確認します。これには、ファイル処理によるファイルタイプと署名の確認と、file_data ルールオプションのポインタの設定が含まれます。 dce_smb インスペクタは、file_id インスペクタ（Snort 3 オープンソースドキュメントで説明されており、https://www.snort.org/snort3 で入手可）と連携して使用する場合、SMB バージョン 1、2、および 3 の通常の SMB ファイル転送の検査をサポートします。ファイル検査を有効にするには、必要に応じて file_id インスペクタを構成し、dce_smb smb_file_inspection および smb_file_depth パラメータを設定します。smb_file_depth パラメータは file_data IPS ルールオプションが示すポインタから開始して、file_id が確認するファイルデータのバイト数を示します。詳細については、https://www.snort.org/snort3 で入手可能な Snort 3 のオープンソースのドキュメントを参照してください。

最適化

dce_smb インスペクタは、フラグメント化されたデータパケットのリアセンブルをサポートしています。この機能は、インラインモードで、完全な最適化が実行される前の検査プロセスの早い段階でエクスプロイトをキャッチしたり、フラグメント化を利用してそれ自体を隠すエクスプロイトをキャッチしたりするのに役立ちます。最適化を無効にすると、多数の誤検知が発生する可能性があることに注意してください。

DCE SMB インスペクタのパラメータ

DCE SMB ポートの設定

binder インスペクタは、DCE SMB ポートの設定を定義します。詳細については、『バインダインスペクタの概要』を参照してください。

例：

```
[
   {
       "when": {
           "role":"any",
           "service":"netbios-ssn",
           "ports": ""
       },
        "use": {
            "type":"dce_smb"
       }
   }
]
```

max_frag_len

最適化のために許可される最大フラグメント長をバイト単位で指定します。より大きなフラグメントを処理する場合、インスペクタは、最適化する前にパケットコンテンツをこのサイズに考慮します。

（注） このパラメータで指定する値は、確実に検出するためにルールでデータを調べる必要がある深さ以上にする必要があります。すべてのデータの検出が確実に行われるようにするには、デフォルト値を使用します。

型：整数

有効な範囲：1514 ～ 65535

デフォルト値：65535

smb_max_compound

1 つの SMB 要求で処理するコマンドの最大数を指定します。

型：整数

有効な範囲：0 ～ 255

デフォルト値：3

smb_max_chain

許可されている連結 SMB AndX コマンドの最大数を指定します。通常、多数の連結 AndX コマンドは異常な動作を表し、場合によっては回避試行を示している可能性があります。連結コマンドを許可しない場合は `1` を指定し、連結コマンドの数の検出を無効にするには `0` を指定します。

`dce_smb` インスペクタは最初に連結コマンドの数をカウントし、関連する SMB インスペクタのルールが有効であり、連結コマンドの数が設定されている値と等しいかそれ以上の場合にはイベントが生成されます。その後、処理が続行されます。

イベントを生成し、インライン展開でこのパラメータの問題のあるパケットをドロップするには、ルール 133:20 を有効にします。

型：整数

有効な範囲：`0` 〜 `255`

デフォルト値：`3`

disable_defrag

フラグメント化された DCE/RPC トラフィックを最適化するかどうかを指定します。有効にすると、`dce_smb` インスペクタは異常を検出して DCE/RPC データをルールエンジンに送信しますが、フラグメント化された DCE/RPC データでのエクスプロイトを見落とすリスクがあります。

`disable_defrag` は、トラフィックを最適化しない柔軟性を提供して処理を高速化しますが、ほとんどの DCE/RPC エクスプロイトは、エクスプロイトを隠蔽するフラグメント化を試行します。このパラメータを有効にすると、既知のエクスプロイトのほとんどがバイパスされ、誤検知が大量に発生します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

limit_alerts

DCE アラートをフローごとの署名ごとに最大 1 つに制限するかどうかを指定します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`true`

reassemble_threshold

リアセンブルされたパケットをルールエンジンに送信する前にキューに入れるには、DCE/RPC の最適化と最適化バッファの最小バイト数を指定します。このパラメータは、完全な最適化が実行される前の早い段階でエクスプロイトを検出する可能性があるため、インラインモードで役立ちます。

低い値を指定すると、早期検出の可能性が高くなりますが、パフォーマンスに悪影響を及ぼす可能性があることに注意してください。このパラメータを有効にした場合は、パフォーマンスへの影響をテストする必要があります。

値 0 は、リアセンブルを無効にします。

型：整数

有効な範囲：0 ～ 65535

デフォルト値：0

policy

モニタ対象のネットワークセグメント上のターゲットホスト（複数可）が使用する Windows または Samba の DCE/RPC の実装を指定します。

型：列挙体

有効な値：Win2000、WinXP、WinVista、Win2003、Win2008、Win7、Samba、Samba-3.0.37、Samba-3.0.22、Samba-3.0.20 から選択した文字列

デフォルト値：WinXP

smb_max_credit

未処理の要求の最大数を指定します。

型：整数

有効な範囲：1 ～ 65536

デフォルト値：8192

smb_file_depth

ファイルが SMB トラフィックで検出されたときに検査されるバイト数を指定します。これは、file_data IPS ルールオプションで指定した場所から始まります（https://www.snort.org/snort3 で入手可能な Snort 3 オープンソースドキュメントで説明されています）。

ファイルの検査を無効にするには、-1 を指定します。

ファイルの検査を無制限に行うことを示すには、0 を指定します。

型：整数

有効な範囲：-1 ～ 32767

デフォルト値：16384

SMB トラフィックでファイルが検出された場合、smb_file_depth パラメータは、file_data IPS ルールオプションで設定されたポインタから開始してインスペクタが確認するファイルデータのバイト数を示します。ファイルタイプと署名を検査するために、dce_smb は、file_id インスペクタで設定された enable_type、type_depth、enable_signature、および signature_depth のパラメータを使用します。file_id インスペクタの詳細については、https://www.snort.org/snort3 で入手可能な Snort オープンソースドキュメントを参照してください。

memcap

インスペクタに割り当てられる最大メモリ制限をバイト単位で指定します。最大メモリキャップに達するか、または超過すると、dce_smb インスペクタは最近の使用頻度の最も低いデータを削除して、より多くの領域を生み出します。

型：整数

有効な範囲：512 ～ 9,007,199,254,740,992 (maxSZ)

デフォルト値：8,388,608

smb_fingerprint_policy

インスペクタが SMB Session Setup AndX の要求と応答で識別されている Windows バージョンまたは Samba のバージョンを検出するようにします。検出されたバージョンが policy インスペクタのパラメータに設定されている Windows バージョンまたは Samba バージョンと異なる場合、検出されたバージョンでそのセッションに設定されているバージョンのみがオーバーライドされます。

たとえば、policy を Windows XP に設定し、インスペクタが Windows Vista を検出した場合は、インスペクタはそのセッションに Windows Vista ポリシーを使用します。その他の設定は引き続き有効です。

型：列挙体

有効な値：none、client、server、または both

- サーバー/クライアント トラフィックでポリシータイプを検査するには、client を使用します。
- クライアント/サーバー トラフィックでポリシータイプを検査するには、server を使用します。
- サーバー/クライアントトラフィックとクライアント/サーバー トラフィックでポリシーを検索するには、both を使用します。
- Windows または Samba のバージョン検査を無効にするには、none を使用します。

デフォルト値：none

smb_legacy_mode

smb_legacy_mode が true の場合、システムは SMB バージョン 1 トラフィックにのみ SMB 侵入ルールを適用し、トランスポートとして SMB バージョン 1 を使用して DCE/RPC 侵入ルールを DCE/RPC トラフィックに適用します。

smb_legacy_mode が false の場合、システムは SMB バージョン 1 と 2 を使用して SMB 侵入ルールをトラフィックに適用し、次を実行します。

- バージョン 7.0 と 7.0.x の場合、システムは SMB バージョン 1 のみのトランスポートとして SMB を使用して、DCE/RPC 侵入ルールを DCE/RPC トラフィックに適用します。

- バージョン 7.1 以降の場合、システムは SMB バージョン 1 と 2 のトランスポートとして SMB を使用して、DCE/RPC 侵入ルールを DCE/RPC トラフィックに適用します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

valid_smb_versions

検査する SMB バージョンを指定します。複数の SMB バージョンは余白文字で区切ります。

型：文字列

有効な値：`v1`、`v2`、`v3`、`all`

デフォルト値：`all`

DCE SMB インスペクタのルール

`dce_smb` インスペクタのルールを有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

表 1 : DCE SMB インスペクタのルール

GID:SID	ルール メッセージ
133:2	SMB：NetBIOS セッションのサービスセッションタイプが不適切（SMB - bad NetBIOS session service session type）
133:3	SMB：SMB メッセージタイプが不適切（SMB - bad SMB message type）
133:4	SMB：SMB ID が不適切（SMB1 に対して \xffSMB ではなく、SMB2 に対して \xfeSMB でない）（SMB - bad SMB Id (not \xffSMB for SMB1 or not \xfeSMB for SMB2)）
133:5	SMB：単語数または構造サイズが不適切（SMB - bad word count or structure size）
133:6	SMB：バイト数が不適切（SMB - bad byte count）
133:7	SMB：フォーマットタイプが不適切（SMB - bad format type）
133:8	SMB：オフセットが不適切（SMB - bad offset）
133:9	SMB：合計データ数がゼロになっている（SMB - zero total data count）
133:10	SMB：NetBIOS データ長が SMB ヘッダー長未満（SMB - NetBIOS data length less than SMB header length）

GID:SID	ルール メッセージ
133:11	SMB：残りの NetBIOS データ長がコマンド長未満（SMB - remaining NetBIOS data length less than command length）
133:12	SMB：残りの NetBIOS データ長がコマンドバイト数未満（SMB - remaining NetBIOS data length less than command byte count）
133:13	SMB：残りの NetBIOS データ長がコマンドデータサイズ未満（SMB - remaining NetBIOS data length less than command data size）
133:14	SMB：残りの合計データ数がこのコマンド データ サイズ未満（SMB - remaining total data count less than this command data size）
133:15	SMB： 送信された合計データ（STDu64）が予想されるコマンドの合計データよりも多い（SMB - total data sent (STDu64) greater than command total data expected）
133:16	SMB：バイト数がコマンドデータサイズ未満（STDu64）（SMB - byte count less than command data size (STDu64)）
133:17	SMB：バイト数のコマンドデータサイズが無効（SMB - invalid command data size for byte count）
133:18	SMB：保留中のツリー接続応答があるツリー接続要求が多すぎる（SMB - excessive tree connect requests with pending tree connect responses）
133:19	SMB：保留中の読み取り応答がある読み取り要求が多すぎる（SMB - excessive read requests with pending read responses）
133:20	SMB：コマンドチェーンが多すぎる（SMB - excessive command chaining）
133:21	SMB：チェーンログイン要求が複数ある（SMB - multiple chained login requests）
133:22	SMB：チェーンツリー接続要求が複数ある（SMB - multiple chained tree connect requests）
133:23	SMB：チェーン/複合ログインの後にログオフする（SMB - chained/compounded login followed by logoff）
133:24	SMB：チェーン/複合ツリー接続の後にツリーが切断される（SMB - chained/compounded tree connect followed by tree disconnect）
133:25	SMB：チェーン/複合オープンパイプ後のクローズパイプ（SMB - chained/compounded open pipe followed by close pipe）
133:26	SMB：無効な共有アクセス（SMB - invalid share access）

GID:SID	ルール メッセージ
133:44	SMB：無効な SMB バージョン 1 が表示された（SMB - invalid SMB version 1 seen）
133:45	SMB：無効な SMB バージョン 2 が表示された（SMB - invalid SMB version 2 seen）
133:46	SMB：ユーザー、ツリー接続、ファイルバインディングが無効（SMB - invalid user, tree connect, file binding）
133:47	SMB：コマンドの複合化が過度（SMB - excessive command compounding）
133:48	SMB：データ数がゼロ（SMB - zero data count）
133:50	SMB：未処理の要求の最大数を超えた（SMB - maximum number of outstanding requests exceeded）
133:51	SMB：未処理要求の MID が同じ（SMB - outstanding requests with same MID）
133:52	SMB：非推奨の言語がネゴシエートされた（SMB - deprecated dialect negotiated）
133:53	SMB：非推奨のコマンドが使用された（SMB - deprecated command used）
133:54	SMB：異常なコマンドが使用された（SMB - unusual command used）
133:55	SMB：コマンドの無効なセットアップ数（SMB - invalid setup count for command）
133:56	SMB：クライアントがセッションで複数言語のネゴシエーションを試みた（SMB - client attempted multiple dialect negotiations on session）
133:57	SMB：クライアントがファイルの属性を読み取り専用/非表示/システムに作成または設定しようとした（SMB - client attempted to create or set a file's attributes to readonly/hidden/system）
133:58	SMB：提供されたファイルオフセットが指定したファイルサイズを超えている（SMB - file offset provided is greater than file size specified）
133:59	SMB：SMB2 ヘッダーに指定した次のコマンドがペイロード境界を超えている（SMB - next command specified in SMB2 header is beyond payload boundary）

DCE インスペクタの侵入ルールのオプション

dce_iface

次のコンマ区切りの要素を指定します。

- サービスインターフェイスの UUID。
- インターフェイスのバージョン（オプション）。デフォルト設定は、どのバージョンにも一致します。
- ルールが要求内のいずれかのフラグメントに一致する必要があるかどうかのインジケータ（オプション）。デフォルト設定は、最初のフラグメントのみに一致します。

DCE/RPC プロトコルでは、クライアントはサービスを呼び出す前にサービスにバインドする必要があります。クライアントは、バインド要求をサーバーに送信するときに、バインド先の 1 つ以上のサービスインターフェイスを指定できます。各インターフェイスはUUIDで表され、各インターフェイスの UUID は一意のインデックス（またはコンテキスト ID）とペアになっており、このインデックスを使用して、クライアントが呼び出しているサービスを今後の要求で参照できます。サーバーは、有効なものとして受け入れるインターフェイス UUID で応答し、クライアントがそれらのサービスに要求を行うことを許可します。クライアントが要求を行うと、コンテキストIDが指定されるため、サーバーはクライアントが要求を行っているサービスを認識します。

`dce_iface` ルールオプションを使用すると、ルールは、クライアントが特定のインターフェイスUUIDにバインドされているかどうかと、このクライアント要求がそのインターフェイスへの要求を行っているかどうかをインスペクタに問い合わせることができます。これにより、インスペクタがバインド UUID を要求で使用されるコンテキスト ID に関連付けることができるため、複数のサービスが正常にバインドされている場合の誤検知を排除できます。

`dce_iface` オプションは、インスペクタでのコネクション型の DCE/RPC に対するサーバーの Bind Ack 応答と Alter Context 応答だけでなく、クライアントの Bind 要求と Alter Context 要求の追跡が必要です。Bind および Alter Context 要求ごとに、クライアントは、インターフェイスを参照するために DCE/RPC セッション中に使用される各インターフェイス UUID のハンドル（またはコンテキスト ID）とともに、インターフェイス UUID のリストを指定します。サーバー応答は、クライアントが要求を行うことを許可するインターフェイスを示します。これは、特定のインターフェイスにバインドするクライアントの要求を受け入れるか拒否します。この追跡は、要求が処理されるときに、要求で使用されるコンテキスト ID を、それがハンドルであるインターフェイス UUID と関連付けることができるようにするために必要です。

`dce_iface` ルールオプションは、次の場合に一致します。

- 指定したインターフェイス UUID が DCE/RPC 要求のインターフェイス UUID（コンテキスト ID によって参照される）と一致する

および

- version 引数が指定されていないか、または version 引数が指定されていて、DCE/RPC 要求のインターフェイス UUID と一致する

および

- any_frag 引数が指定されているか、any_frag 引数がなく、dce_iface オプションが最初の要求フラグメントの UUID とバージョン基準に一致する

例：

```
dce_iface:4b324fc8-1670-01d3-1278-5a47bf6ee188;
dce_iface:4b324fc8-1670-01d3-1278-5a47bf6ee188, <2;
dce_iface:4b324fc8-1670-01d3-1278-5a47bf6ee188,any_frag;
dce_iface:4b324fc8-1670-01d3-1278-5a47bf6ee188, =1, any_frag;
```

dce_iface.uuid

DCE/RPC 要求では、UUID 番号がビッグエンディアンまたはリトルエンディアンのどちらで表されるかを指定できます。要求でのインターフェイス UUID の表現は、要求で指定したエンディアンに応じて異なります。dce_rpc インスペクタは UUID を正規化します。つまり、dce_iface ルールオプションの UUID の指定は、要求のエンディアンに関係なく、同じように記述する必要があります。

たとえば、リトルエンディアンのバインド要求は、次のように UUID を表します。

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

ビッグエンディアンのバインド要求は、次のように同じ UUID を表します。

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

dce_iface オプションを使用する Snort 3 ルールでは、要求のエンディアンに関係なく、ビッグエンディアン順を使用して UUID を文字列で表す必要があります。

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

型：文字列

シンタックス：`dce_iface: <UUID>;`

有効な値： *UUID* は 32 の 16 進数で、ハイフンで区切られた 5 つの グループ（xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx の形式）で表示されます。

例： `dce_iface: 5a7b91f8-ff00-11d0-a9b2-00c04fb6e6fc;`

dce_iface.version

サービスインターフェイスには、それに関連付けられたバージョンがあります。インターフェイスの一部のバージョンは、特定のエクスプロイトに対して脆弱ではない場合があります。そのため、dce_iface オプションで 1 つ以上のバージョン番号を指定して、特定のエクスプロイトを確認する必要があるかどうかを特定できます。

型：間隔

シンタックス：`dce_iface: <range_operator><positive integer>;` または `dce_iface: <positive integer><range_operator><positive integer>;`

有効な値：1 つ以上の一連の正のバージョン番号と、範囲形式の表で指定されている `range_operator` の1 つ。

例：`dce_iface: =6;`

dce_iface.any_frag

DCE/RPC 要求は、1 つ以上のフラグメントに分割できます。DCE/RPC ヘッダーにフラグが設定されて現在のフラグメントが要求の最初のフラグメントか、中間のフラグメントか、または最後のフラグメントであるかを示します。DCE/RPC 要求内のデータの確認の多くは、DCE/RPC 要求が最初のフラグメント（または完全な要求）である場合にのみ関連します。したがって、最初のフラグメントに続くフラグメントには、DCE/RPC 要求のより深いデータが含まれます。たとえば、要求の最初の 5 バイト（長さフィールドなど）内のデータを検索するルールでは、最初のフラグメント以外のフラグメントで誤ったデータを検出します。後続のフラグメントの開始は、要求の開始からある程度の長さでオフセットされます。これは、フラグメント化された DCE/RPC トラフィックの誤検知の原因となる可能性があります。

そのため、デフォルトでは、DCE_RPC インスペクタは要求の最初のフラグメントのみを照合します。インスペクタが一致の要求内のすべてのフラグメントを調べるようにするには、`dce_iface` ルールオプションに `any_frag` を追加します。最適化された DCE/RPC 要求は完全な要求と見なされることに注意してください。

シンタックス：`dec_iface: any_frag;`

例：`dce_iface: any_frag;`

dce_opnum

DCE RPC 操作番号、操作番号の範囲、または操作番号のリストと照合します。このオプションは、インターフェイスに対して実行できる 1 つ以上の特定の関数呼び出しを表します。クライアントが特定のサービスインターフェイスにバインドして要求を行った後、ルールは、クライアントがサービスに対して行っている関数呼び出しを特定して、関数呼び出し内に存在する可能性のあるエクスプロイトを確認する必要があります。関数呼び出しは、操作番号（opnums）の二重引用符で囲まれたリストとして指定されます。

型：文字列

シンタックス：`dce_opnum: <opnum_list>;`

`opnum_list` は次のいずれかです。

- 単一の整数。
- コンマ区切りの整数のリスト。
- 範囲内の最小数と最大数をハイフンで区切って指定した整数の範囲。
- 上記の組み合わせ。

有効な値：DCE/RPC 要求の opnum のリスト。

例：

```
dce_opnum: "15";
dce_opnum: "15-18";
dce_opnum: "15, 18-20";
dce_opnum: "15, 17, 20-22";
```

dce_stub_data

このオプションは、先行するルールオプションに関係なく、DCE/RPC スタブデータの先頭に検出カーソル（ルール処理でパケットペイロードを通過させるために使用）を配置します。このオプションは、DCE/RPC スタブデータが存在する場合に一致します。

シンタックス：`dce_stub_data;`

例：`dce_stub_data;`

表 *2:* 範囲の形式

範囲の形式	演算子	説明
operator i		
	<	より少ない
	>	右辺と比較して大きい
	=	等しい
	≠	等しくない
	≤	以下
	≥	以上
j operator k		
	<>	j よりも大きく、k よりも小さい
	<=>	j 以上で k 以下

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。