CHAPTER 49

データベースとディレクトリのプロトコル インスペクションの設定

この章では、アプリケーション レイヤ プロトコル インスペクションを設定する方法について説明します。インスペクション エンジンは、ユーザのデータ パケット内に IP アドレッシング情報を埋め込むサービスや、ダイナミックに割り当てられるポート上でセカンダリ チャネルを開くサービスに必要です。これらのプロトコルでは、高速パスでパケットを渡すのではなく、ASA で詳細なパケット インスペクションを行う必要があります。そのため、インスペクション エンジンがスループット全体に影響を与えることがあります。

ASA では、デフォルトでいくつかの一般的なインスペクション エンジンがイネーブルになっていますが、ネットワークによっては他のインスペクション エンジンをイネーブルにしなければならない場合があります。

この章は、次の項で構成されています。

ILS インスペクション

ILS インスペクション エンジンは、LDAP を使用してディレクトリ情報を ILS サーバと交換する Microsoft NetMeeting、SiteServer、および Active Directory の各製品に対して Network Address Translation（NAT; ネットワーク アドレス変換）をサポートします。

ASA は ILS に対して NAT をサポートします。NAT は、ILS または SiteServer Directory のエンドポイントの登録および検索で使用されます。LDAP データベースには IP アドレスだけが保存されるため、Port Address Translation（PAT; ポート アドレス変換）はサポートされません。

LDAP サーバが外部にある場合、内部ピアが外部 LDAP サーバに登録された状態でローカルに通信できるように、検索応答に対して NAT を行うことを検討してください。このような検索応答では、最初に xlate が検索され、次に DNAT エントリが検索されて正しいアドレスが取得されます。これらの検索が両方とも失敗した場合、アドレスは変更されません。NAT 0（NAT なし）を使用していて、DNAT の相互作用を想定していないサイトの場合は、パフォーマンスを向上させるためにインスペクション エンジンをオフにすることをお勧めします。

ILS サーバが ASA 境界の内部にある場合は、さらに設定が必要なことがあります。この場合、外部クライアントが指定されたポート（通常は TCP 389）の LDAP サーバにアクセスするためのホールが必要となります。

ILS トラフィックはセカンダリ UDP チャネルだけで発生するため、TCP 接続は一定の間隔 TCP アクティビティがなければ切断されます。デフォルトでは、この間隔は 60 分です。この値は、**timeout** コマンドを使用して調整できます。

ILS/LDAP はクライアント/サーバ モデルに従っており、セッションは 1 つの TCP 接続で処理されます。クライアントのアクションに応じて、このようなセッションがいくつか作成されることがあります。

接続ネゴシエーション時間中、クライアントからサーバに BIND PDU が送信されます。サーバから成功を示す BIND RESPONSE を受信すると、ILS Directory に対する操作を実行するためのその他の操作メッセージ（ADD、DEL、SEARCH、MODIFY など）が交換される場合があります。ADD REQUEST PDU および SEARCH RESPONSE PDU には、NetMeeting セッションを確立するために H.323（SETUP および CONNECT メッセージ）によって使用される、NetMeeting ピアの IP アドレスが含まれている場合があります。Microsoft NetMeeting v2.X および v3.X は、ILS をサポートしています。

ILS インスペクションでは、次の操作が実行されます。

- BER 復号化機能を使用して LDAP REQUEST PDU/RESPONSE PDU を復号化する。
- LDAP パケットを解析する。
- IP アドレスを抽出する。
- 必要に応じて IP アドレスを変換する。
- BER 符号化機能を使用して、変換後のアドレスが含まれる PDU を符号化する。
- 新しく符号化された PDU を元の TCP パケットにコピーする。
- TCP チェックサムとシーケンス番号の増分を調整する。

ILS インスペクションには、次の制限事項があります。

- 参照要求および応答はサポートされない。
- 複数のディレクトリ内のユーザは統合されない。
- 1 人のユーザが複数のディレクトリで複数の ID を持つ場合、NAT はそのユーザを認識できない。

（注） H225 コール シグナリング トラフィックはセカンダリ UDP チャネルだけで発生するため、TCP **timeout** コマンドで指定された間隔が経過すると、TCP 接続は切断されます。デフォルトで、この間隔は 60 分に設定されています。

SQL*Net インスペクション

SQL*Net インスペクションはデフォルトでイネーブルになっています。

SQL*Net プロトコルは、さまざまなパケット タイプで構成されています。ASA はこれらのパケットを処理して、ASA のどちらの側の Oracle アプリケーションにも一貫性のあるデータ ストリームが表示されるようにします。

SQL*Net のデフォルトのポート割り当ては 1521 です。これは、Oracle が SQL*Net 用に使用している値ですが、Structured Query Language（SQL; 構造化照会言語）の IANA ポート割り当てとは一致しません。SQL*Net インスペクションを一連のポート番号に適用するには、**class-map** コマンドを使用します。

（注）　SQL 制御 TCP ポート 1521 と同じポートで SQL データ転送が行われる場合は、SQL*Net のインスペクションをディセーブルにします。SQL*Net インスペクションがイネーブルになっていると、セキュリティ アプライアンスはプロキシとして機能し、クライアントのウィンドウ サイズを 65000 から約 16000 に減らすため、データ転送の問題が発生します。

ASA は、すべてのアドレスを変換し、パケットを調べて、SQL*Net バージョン 1 用に開くすべての埋め込みポートを見つけます。

SQL*Net バージョン 2 の場合、データ長ゼロの REDIRECT パケットの直後に続くすべての DATA パケットまたは REDIRECT パケットはフィックスアップされます。

フィックスアップが必要なパケットには、埋め込みホスト アドレスおよびポート アドレスが次の形式で含まれています。

```
(ADDRESS=(PROTOCOL=tcp)(DEV=6)(HOST=a.b.c.d)(PORT=a))
```

SQL*Net バージョン 2 の各 TNSFrame タイプ（Connect、Accept、Refuse、Resend、Marker）は、NAT 対象のアドレスがあるかどうかがスキャンされません。また、インスペクションがパケット内に埋め込まれたポートにダイナミック接続を開くこともありません。

SQL*Net バージョン 2 の TNSFrame、Redirect パケット、および Data パケットは、ペイロードのデータ長がゼロの REDIRECT TNSFrame タイプの後に続く場合、開くポートおよび NAT 対象のアドレスがあるかどうかスキャンされます。データ長ゼロの Redirect メッセージが ASA を通過すると、後に続く Data メッセージまたは Redirect メッセージは変換対象であり、ポートはダイナミックに開かれると想定するフラグが、接続データ構造に設定されます。先行するパラグラフの TNS フレームのいずれかが Redirect メッセージの後に到着した場合、フラグはリセットされます。

SQL*Net インスペクション エンジンは、チェックサムを再計算し、IP および TCP の長さを変更し、新旧のメッセージの長さの差を使用してシーケンス番号と確認応答番号を再調整します。

SQL*Net バージョン 1 では、その他のすべての場合を想定しています。TNSFrame タイプ（Connect、Accept、Refuse、Resend、Marker、Redirect、Data）とすべてのパケットは、ポートおよびアドレスがあるかどうかスキャンされます。アドレスが変換され、ポート接続が開かれます。

Sun RPC インスペクション

この項では、Sun RPC アプリケーション インスペクションについて説明します。この項は、次の内容で構成されています。

Sun RPC インスペクションの概要

Sun RPC インスペクション エンジンは、Sun RPC プロトコルのアプリケーション インスペクションをイネーブルまたはディセーブルにします。Sun RPC は、NFS および NIS で使用されます。Sun RPC サービスはどのポート上でも実行できます。サーバ上の Sun RPC サービスにアクセスしようとするクライアントは、そのサービスが実行されているポートを知る必要があります。そのためには、予約済みポート 111 でポート マッパー プロセス（通常は rpcbind）に照会します。

クライアントがサービスの Sun RPC プログラム番号を送信すると、ポート マッパー プロセスはサービスのポート番号を応答します。クライアントは、ポート マッパー プロセスによって特定されたポートを指定して、Sun RPC クエリーをサーバに送信します。サーバが応答すると、ASA はこのパケットを代行受信し、そのポートで TCP と UDP の両方の初期接続を開きます。

Sun RPC インスペクションには、次の制限事項が適用されます。

- Sun RPC ペイロード情報の NAT または PAT はサポートされていません。
- Sun RPC インスペクションは、着信アクセス リストだけをサポートします。Sun RPC インスペクションは発信アクセス リストはサポートしません。これは、インスペクション エンジンでセカンダリ接続でなくダイナミック アクセス リストが使用されるためです。ダイナミック アクセス リストは常に入力方向に追加され、出力方向には追加されません。したがって、このインスペクション エンジンは発信アクセス リストをサポートしません。ASA に設定されているダイナミック アクセス リストを表示するには、**show asp table classify domain permit** コマンドを使用します。**show asp table classify domain permit** コマンドの詳細については、CLI コンフィギュレーション ガイド を参照してください。

Sun RPC サービスの管理

Sun RPC サービス テーブルを使用して、確立された Sun RPC セッションに基づいて ASA を経由する Sun RPC トラフィックを制御します。Sun RPC サービス テーブルのエントリを作成するには、グローバル コンフィギュレーション モードで **sunrpc-server** コマンドを使用します。

```
hostname(config)# sunrpc-server interface_name ip_address mask service service_type
protocol {tcp | udp} port[-port] timeout hh:mm:ss
```

このコマンドを使用して、Sun RPC アプリケーション インスペクションで開いたピンホールを閉じるまでのタイムアウトを指定できます。たとえば、IP アドレスが 192.168.100.2 の Sun RPC サーバに対して 30 分のタイムアウトを作成するには、次のコマンドを入力します。

```
hostname(config)# sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003
protocol tcp 111 timeout 00:30:00
```

このコマンドは、Sun RPC アプリケーション インスペクションで開いたピンホールが 30 分後に閉じるように指定します。この例では、Sun RPC サーバは TCP ポート 111 を使用する内部インターフェイスにあります。UDP、別のポート番号、ポート範囲を指定することもできます。ポート範囲を指定するには、範囲の開始ポート番号と終了ポート番号をハイフンで区切ります（111-113 など）。

サービス タイプは、特定のサービス タイプとそのサービスに使用するポート番号の間のマッピングを特定します。サービス タイプ（この例では 100003）を判定するには、Sun RPC サーバ マシンの UNIX または Linux コマンドラインで、**sunrpcinfo** コマンドを使用します。

Sun RPC コンフィギュレーションを消去するには、次のコマンドを入力します。

```
hostname(config)# clear configure sunrpc-server
```

これによって、**sunrpc-server** コマンドを使用して実行されるコンフィギュレーションが削除されます。**sunrpc-server** コマンドを使用して、指定したタイムアウト値を持つピンホールを作成できます。

アクティブな Sun RPC サービスを消去するには、次のコマンドを入力します。

```
hostname(config)# clear sunrpc-server active
```

これによって、そのサービス（NFS、NIS など）で Sun RPC アプリケーション インスペクションが開いたピンホールが消去されます。

Sun RPC インスペクションの確認とモニタリング

この項の出力例では、Sun RPC サーバの IP アドレスは 192.168.100.2 で内部インターフェイスにあり、Sun RPC クライアントの IP アドレスは 209.168.200.5 で外部インターフェイスにあるものとします。

現在の Sun RPC 接続に関する情報を表示するには、**show conn** コマンドを入力します。次に、**show conn** コマンドの出力例を示します。

```
hostname# show conn
15 in use, 21 most used
UDP out 209.165.200.5:800 in 192.168.100.2:2049 idle 0:00:04 flags -
UDP out 209.165.200.5:714 in 192.168.100.2:111 idle 0:00:04 flags -
UDP out 209.165.200.5:712 in 192.168.100.2:647 idle 0:00:05 flags -
UDP out 192.168.100.2:0 in 209.165.200.5:714 idle 0:00:05 flags i
hostname(config)#
```

Sun RPC サービス テーブル コンフィギュレーションに関する情報を表示するには、**show running-config sunrpc-server** コマンドを入力します。次に、**show running-config sunrpc-server** コマンドの出力例を示します。

```
hostname(config)# show running-config sunrpc-server
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003 protocol UDP port 111
timeout 0:30:00
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100005 protocol UDP port 111
timeout 0:30:00
```

この出力では、IP アドレスが 192.168.100.2 で内部インターフェイスにある Sun RPC サーバの UDP ポート 111 で、タイムアウト間隔が 30 分に設定されていることが示されています。

Sun RPC サービスで開かれているピンホールを表示するには、**show sunrpc-server active** コマンドを入力します。次に、**show sunrpc-server active** コマンドの出力例を示します。

```
hostname# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-----------------------------------------------
1 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.200.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.200.5/0 192.168.100.2/650 100005 0:30:00
```

LOCAL カラムのエントリは、内部インターフェイスのクライアントまたはサーバの IP アドレスを示します。FOREIGN カラムの値は、外部インターフェイスのクライアントまたはサーバの IP アドレスを示します。

Sun RPC サーバで実行されている Sun RPC サービスに関する情報を表示するには、Linux または UNIX サーバのコマンドラインから **rpcinfo -p** コマンドを入力します。次に、**rpcinfo -p** コマンドの出力例を示します。

```
sunrpcserver:~ # rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 632 status
100024 1 tcp 635 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 32771 nlockmgr
100021 3 udp 32771 nlockmgr
100021 4 udp 32771 nlockmgr
100021 1 tcp 32852 nlockmgr
100021 3 tcp 32852 nlockmgr
```

```
100021 4 tcp 32852 nlockmgr
100005 1 udp 647 mountd
100005 1 tcp 650 mountd
100005 2 udp 647 mountd
100005 2 tcp 650 mountd
100005 3 udp 647 mountd
100005 3 tcp 650 mountd
```

この出力では、ポート 647 が UDP 上で実行されている mountd デーモンに対応しています。mountd プロセスは、通常、ポート 32780 を使用します。この例では、TCP 上で実行されている mountd プロセスがポート 650 を使用しています。