Solaris のシステム管理 (システム管理エージェント)

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819–0387–11
2006 年 11 月

原典: Solaris System Management Agent Administration Guide

Part No: 817-3000-11

Revision A

061107@15490

目次

表目次

図目次

はじめに

『Solaris のシステム管理 (システム管理エージェント)』 では、システム管理エージェント (SMA) のインストール、構成、および操作方法について説明します。

注 – このリリースでは、 SPARC® および x86 系列のプロセッサアーキテクチャー (UltraSPARC®、SPARC64、AMD64、Pentium、Xeon EM64T) を使用するシステムをサポートします。サポートされるシステムについては、Solaris 10 Hardware Compatibility List (http://www.sun.com/bigadmin/hcl) を参照してください。本書では、プラットフォームにより実装が異なる場合は、それを特記します。

本書では、「x86」という用語は AMD64 あるいは Intel Xeon/Pentium 製品系列と互換性のあるプロセッサを使用して製造された 32 ビットおよび 64 ビットシステムを意味します。サポートされるシステムについては、Solaris 10 Hardware Compatibility List を参照してください。

対象読者

このマニュアルは、システム管理エージェントを使って Solaris オペレーティングシステム上のオブジェクトやデバイスを管理するシステム管理者を対象としています。また、別のエージェントからシステム管理エージェントへ管理エージェントタスクを移行するシステム管理者も対象としています。

このマニュアルをお読みになる前に

このマニュアルの読者は、Solaris システムの一般的な管理方法を把握している必要があります。SNMP および SNMP MIB に関する一般的な知識も役立ちます。このほか、次の分野の知識が必要です。

- SNMPv1、SNMPv2c、および SNMPv3 プロトコル
- SMI (Structure of Management Information: 管理情報構造) v1 および v2
- MIB (Management Information Base: 管理情報ベース) の構造
- ASN.1 (Abstract Syntax Notation)

内容の紹介

このマニュアルは次の章で構成されています。

第 1 章では、SNMP の概要とシステム管理エージェントの基本情報について説明します。

第 2 章では、システム管理エージェントの構成に使用するファイルについて説明します。

第 3 章では、システム管理エージェントの停止、再起動などの基本操作について説明します。

第 4 章では、セキュリティー管理とユーザー管理について説明します。

第 5 章では、別のエージェントからシステム管理エージェントへデバイスの管理を移行する方法について説明します。

付録 A では、システム管理エージェント付属のマニュアルページ、ツール、およびユーティリティーに関するリファレンス情報を提供します。

用語集では、このマニュアルで使用される用語およびその定義について説明します。

関連文書

SNMP と SNMP MIB の知識があれば理想的です。

- 「Internet Engineering Task Force RFC 2741 on AgentX 」
- 「Internet Engineering Task Force RFC 3411 on An Architecture for Describing Simple Network Management Protocol Management Frameworks」
- 『Understanding SNMP MIBs』、Perkins、McGinnis 共著 (Prentice Hall)
- 『Solaris System Management Agent Developer's Guide 』
- 『Solaris のシステム管理 (上級編)』
- 『Solaris のシステム管理 (セキュリティサービス)』

マニュアル、サポート、およびトレーニング

Sun の Web サイトでは、次のサービスに関する情報も提供しています。

- マニュアル (http://jp.sun.com/documentation/)
- サポート (http://jp.sun.com/support/)
- トレーニング (http://jp.sun.com/training/)

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用します。

表 P–1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 system%
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	system% **su** password:
AaBbCc123	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm *filename* と入力します。
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第 5 章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% **grep '^#define ** **XV_VERSION_STRING'**

コード例は次のように表示されます。

- C シェル

  machine_name% **command y|n** [*filename*]

- C シェルのスーパーユーザー

  machine_name# **command y|n** [*filename*]

- Bourne シェルおよび Korn シェル

  $ **command y|n** [*filename*]

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[ ] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

◆ ◆ ◆ 第 1 章

1 システム管理エージェントの紹介

システム管理エージェントは、オープンソースの Net-SNMP エージェントの、Sun Microsystems による実装です。この章では、SNMP の主要原理について説明します。この章ではまた、システム管理エージェントの概要を紹介します。

この章で扱うトピックは次のとおりです。

SNMP とネットワーク管理の概要

SNMP (Simple Network Management Protocol) は、インターネット標準のプロトコルです。SNMP は、IP ネットワークに接続されたデバイスを照会、監視、および管理するための共通の方式を提供します。このプロトコルは、RFC 2571 に定義されています。詳細については、http://www.ietf.org/rfc/rfc2571.txt を参照してください。SNMP の詳細情報は、その他の RFC にも定義されています。

SNMP は、システム、ネットワークデバイス、およびネットワークを効果的に管理するため、企業ネットワーク内で広く使用されています。SNMP の利点の 1 つとして、増加し続けるネットワークコンポーネントやアプリケーションに対応するためのソリューションを、迅速に提供できることが挙げられます。SNMP ネットワーク内では、システム、コンポーネント、およびアプリケーションを、「エンティティー」と呼びます。管理を必要とするエンティティーの数は急速に増加しています。

SNMP は、「マネージャー」と「エージェント」からなるアーキテクチャーを使用します。SNMP マネージャーはネットワーク上のホストで実行されるプログラムの一種で、「ネットワーク管理ステーション (NMS)」とも呼ばれます。この SNMP マネージャーには、ネットワークに接続されたデバイス上で実行されている 1 つ以上の SNMP エージェントに、要求を送信する働きがあります。SNMP マネージャーからの SNMP 要求を待機するプログラムを「エージェント (デーモン)」と呼びます。

エージェント階層は、単一のマスターエージェントと複数のサブエージェントで構成されます。マスターエージェントは、SNMP マネージャーから SNMP に基づく管理要求を受信します。マスターエージェントは、これらの管理要求に対する応答を送信します。応答は、それぞれのサブエージェントから適切な値を取得したあとで送信されます。

サブエージェントは、それぞれ異なったコンポーネントの管理を担当します。この管理は、コンポーネントまたはアプリケーション専用に設計された管理情報ベース (MIB) に基づいて行われます。MIB は、管理情報の定義を含む仕様です。MIB を利用して、ネットワークやネットワークシステムをリモート操作で監視、構成、および制御できます。

エージェントは、要求を受け取ると、MIB 内の情報を検索して、マネージャーに情報を返します。MIB 内の各オブジェクトは、管理対象デバイスに関するデータを表します。オブジェクトごとに、MIB 内で一意の識別子が割り当てられています。マネージャーとエージェントが管理対象デバイスに関する情報をやりとりするためには、それぞれが同一の MIB にアクセスする必要があります。マネージャーは、MIB を使って、エージェントが使用する情報の識別子を指定します。エージェントは、同じ MIB を使って、マネージャーの SNMP 要求で渡された識別子を検索します。そしてエージェントは、要求されたデータの値を取得または設定します。システム管理エージェントがサポートする MIB の一覧については、22 ページの「サポートされる MIB」を参照してください。

SNMP のバージョン

システム管理エージェントは、3 つの SNMP プロトコルをサポートします。これらのプロトコルおよび関連する RFCは、次のとおりです。

SNMPv1	SNMP v1 は、RFC 1155 および 1157 (http://www.ietf.org/rfc/rfc1155.txt および http://www.ietf.org/rfc/rfc1157.txt) で定義されています
SNMPv2c	SNMPv2c は、RFC 1901 (http://www.ietf.org/rfc/rfc1901.txt) で定義されています
SNMPv3	SNMPv3 は、RFC 2570 (http://www.ietf.org/rfc/rfc2570.txt) で定義されています

システム管理エージェントでサポートされる SNMP のこれらのバージョンは、共存が可能です。共存のガイドラインについては、RFC 3584 (http://www.ietf.org/rfc/rfc3584.txt) を参照してください。

このマニュアルで紹介するセキュリティーモデルやその他のインスタンスの中には、SNMP の一部のバージョンをサポートしていないものがあります。使用可能な SNMP のバージョンの制約については、このマニュアルおよび関連するマニュアルページに記載されています。この制約は、SNMPv3 の拡張パケット構造に部分的に起因しています。SNMPv3 パケット構造については、 図 1–1 を参照してください。

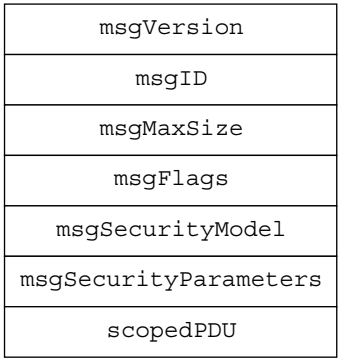

図 1–1 SNMPv3 パケット構造

図 1–1 に示された各パケットの概要は、次のとおりです。

`msgVersion` パケットの SNMP バージョン。使用可能な値は 1、2、3 です。3 の場合、SNMPv3 を表します。

`msgID` マネージャーとエージェント間でやりとりされる要求メッセージと応答メッセージの調整に使用されます。応答の `msgID` は要求の `msgID` と一致している必要があります。

`msgMaxSize` 送信側が別の SNMP エンジンから受信できるメッセージの最大サイズ。

`msgFlags` メッセージの処理方法を指定する 8 ビット。詳細については、47 ページの「VACM セキュリティー情報の格納場所」を参照してください。

`msgSecurityModel` メッセージの生成に使用されるセキュリティーモデルを指定します。詳細については、47 ページの「VACM セキュリティー情報の格納場所」を参照してください。

`msgSecurityParameters` セキュリティーモデルに関するデータを含む 8 ビットの文字列。詳細については、47 ページの「VACM セキュリティー情報の格納場所」を参照してください。

`scopedPDU` 標準のプロトコルデータユニット (PDU) と、この PDU の処理に使用される管理上一意のコンテキストの識別情報を含みます。詳細については、47 ページの「VACM セキュリティー情報の格納場所」を参照してください。

管理情報構造

MIB の書記法は、「管理情報構造 (SMI: Structure of Management Information)」と呼ばれる一連の規則に従っています。この文書セットには、次の情報を指定するものとして業界で広く受け入れられている方法および規則が含まれています。

- 管理情報のモデル
- 管理情報の型
- イベントの型

システム管理エージェントはSMIv2 を使用します。SMIv2 は、論理アクセスが可能になるようにオブジェクト名編成を規定しています。SMIv2 によれば、管理対象の各オブジェクトには次の属性が必要です。

名前	この名前が、オブジェクトを一意に識別するオブジェクト識別子 (OID) です。オブジェクトに OID 値を割り当てると、そのオブジェクトが登録されます。詳細については、22 ページの「ISO 名前空間ツリー」を参照してください。
構文	構文は、整数型、8 ビット文字列型などのデータ型を定義します。
エンコーディング	エンコーディングは、管理対象オブジェクトに関連付けられた情報をマシン間で伝送するために直列化する方法を示します。

コミュニティー文字列

SNMP では、マネージャー (複数可) と 1 つのエージェントで、1 つの「コミュニティー」を構成します。SNMPv1 および v2c のメッセージには、コミュニティーの名前が含まれています。これは、「コミュニティー文字列」と呼ばれます。SNMPv3 パケットが USM 設定で指定されたユーザーに関連付けられるのに対して、SNMPv2 および v1 のパケットは、コミュニティー文字列に関連付けられています。コミュニティー文字列は 8 ビットの文字変数であり、これを使って次のチェックを行うことができます。

- 要求側エンティティーの識別。
- 要求側エンティティーの場所の識別。
- MIB 表示情報の判別。

SMA でサポートされる VACM には、コミュニティー文字列モデルと動的アクセス制御モデルの詳細が記載されています。SNMPv3 の動的アクセス制御モデルについては、46 ページの「VACM によるアクセス制御」を参照してください。

`com2sec` トークンは、コミュニティーで VACM ビューを使用できるようにするため、コミュニティーと SNMPv3 セキュリティー名のマッピングを行います。詳細については、第 4 章を参照してください。

システム管理エージェントの概要

システム管理エージェントは RFC 3411 (http://www.ietf.org/rfc/rfc3411.txt) を実装しています。SMA は、SNMP プロトコルを使ってシステム管理を行う軽量エージェントです。SMA は、Solaris ソフトウェアに標準 SNMP エージェント基盤を提供します。SMA は、アプリケーションプログラミングインタフェースに書き込まれたモジュールと AgentX サブエージェントを使って拡張できます。システム管理エージェントでモジュールを拡張する方法については、『Solaris System Management Agent Developer's Guide 』を参照してください。AgentX の詳細については、http://www.ietf.org/rfc/rfc2741.txt を参照してください。

システム管理エージェントは、スタンドアロンエージェントとして設計されています。SNMP プロトコルを使って SMA と通信する管理アプリケーションは、複数で同時に SMA にアクセスできます。SMA は既存の SNMP エージェントと共存できます。SMA は、従来の SNMP エージェントの一部の代わりになります。

SMA は、Net-SNMP オープンソース実装バージョン 5.0.9 に基づく、Sun の新しい SNMP エージェントです。このオープンソース実装については、http://www.net-snmp.org/ を参照してください。このオープンソース実装は、以前は「UCD-SNMP」と呼ばれていました。システム管理エージェントは、最新の SNMP 標準をサポートするように設計されています。

今回の Solaris リリースでは、システム管理エージェントと Solstice Enterprise Agents™ ソフトウェアを共存させることができます。Solstice Enterprise Agents ソフトウェアの詳細については、『Solstice Enterprise Agents 1.0 ユーザーズガイド』を参照してください。システム管理エージェントは、SNMP マネージャーとしては、Solstice Enterprise Agents ソフトウェアと同様に機能します。Solstice Enterprise Agents ソフトウェアとは異なり、システム管理エージェントは SNMPv3 をサポートしています。システム管理エージェントは、Solstice Enterprise Agents ソフトウェアよりも多くのデフォルト MIB をサポートしています。

Solstice Enterprise Agents から システム管理エージェントへの移行方法については、67 ページの「Solstice Enterprise Agents ソフトウェアからの移行」を参照してください。Solstice Enterprise Agents から システム管理エージェントへのアプリケーションの移行方法については、『Solaris System Management Agent Developer's Guide 』を参照してください。

システム管理エージェントのコンポーネント

システム管理エージェントは、SNMP 管理フレームワークに関連する標準のエージェントコンポーネントを実装しています。このフレームワークは、複数の標準で構成されています。たとえば、次のような標準があります。

- AgentX プロトコル: RFC 2741 に定義されている業界標準のメカニズム (http://www.ietf.org/rfc/rfc2741.txt)。
- USM: RFC 3414 に定義されている、認証およびプライバシ保護のためのユーザーに基づくセキュリティーモデル (http://www.ietf.org/rfc/rfc3414.txt)。42 ページの「USM による認証とメッセージプライバシ」も参照してください。
- VACM: RFC 3415 に定義されている、承認用のビューに基づくアクセス制御モデル (http://www.ietf.org/rfc/rfc3415.txt)。46 ページの「VACM によるアクセス制御」も参照してください。

関連する他の RFC の詳細については、22 ページの「サポートされる MIB」を参照してください。システム管理エージェントの構成は変更可能です。構成の変更やその他の単純な SNMP 操作を実行するためのコマンド行ツールが用意されています。システム管理エージェントは、動的モジュールと Agent-X サブエージェントを使って拡張できます。詳細については、『Solaris System Management Agent Developer's Guide 』を参照してください。

システム管理エージェントに付属している各種パッケージの概要については、25 ページの「プラットフォームとパッケージ」を参照してください。

システム管理エージェントの一部のコンポーネントの関係については、図 1–2 を参照してください。

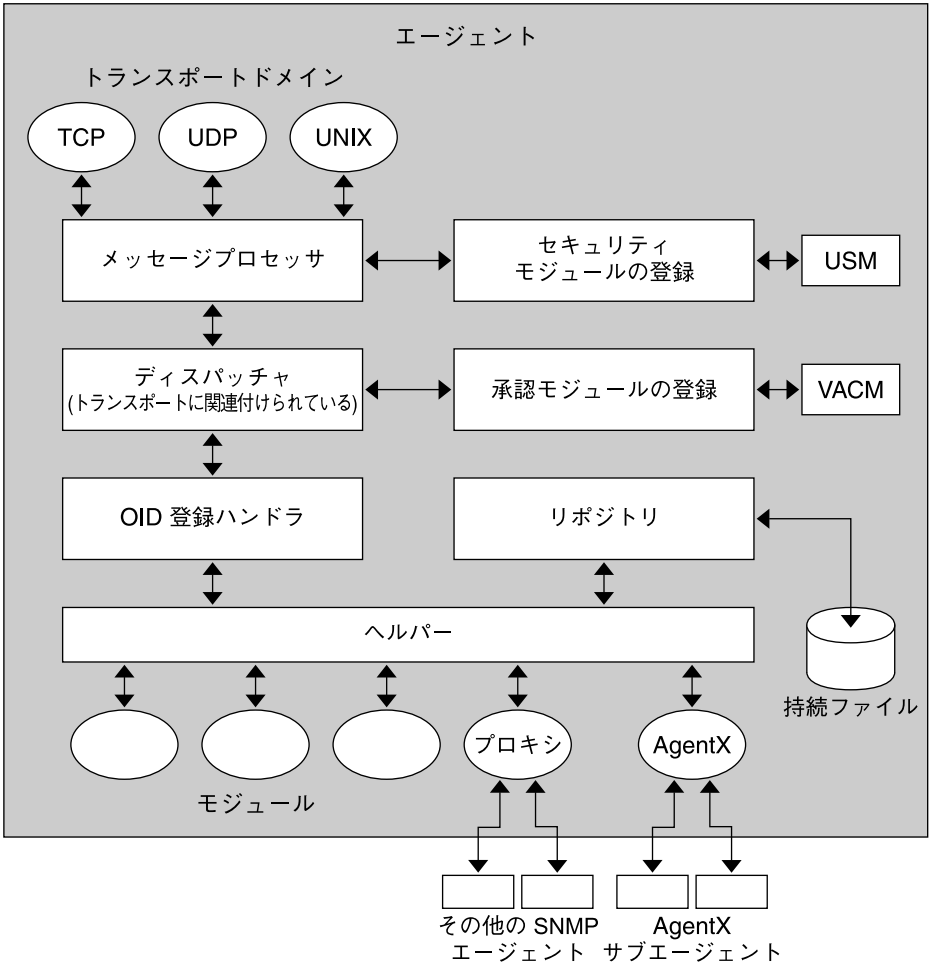

図 1–2 システム管理エージェントのコンポーネント

この図は、セキュリティーおよび承認を使用して、メッセージプロセッサ、ディスパッチャー、および OID の登録処理を行うプログラム群の相互関係を示しています。この図では、その他の SNMP エージェントが、プロキシを介してシステム管理エージェントと通信しています。この図ではまた、AgentX サブエージェントが、AgentX プロトコルを使ってシステム管理エージェントと通信しています。AgentX の詳細については、30 ページの「AgentX プロトコルの使用」を参照してください。図 1–2 に記載されたコンポーネントの通信の詳細については、『Solaris System Management Agent Developer's Guide』の「Overview of the System Management Agent」を参照してください。

ISO 名前空間ツリー

すべての管理対象オブジェクト (デバイスまたはデバイスの特性) は、名前、構文、およびエンコーディングを持っています。この名前が、オブジェクトを一意に識別するオブジェクト識別子 (OID) です。OID は、整数をピリオドで区切った形式で表されます。たとえば、1.3.6.1.2.1.1.1.0 は、管理サブツリーのシステムグループ内のシステムの説明を指定します。OID スキーマの一部は、ISO 組織によって作成されました。このため、OID 値を表すのに使用されるツリー構造の図を「ISO 図」と呼びます。システム管理エージェント全体の ISO 図を、図 1–3 に示します。

SMA の OID は `1.3.6.1.4.1.42.2.2.4` です。

この OID は次のデータに相当します。

```
iso.org.dod.internet.private.enterprises.sun.products.management.sma
```

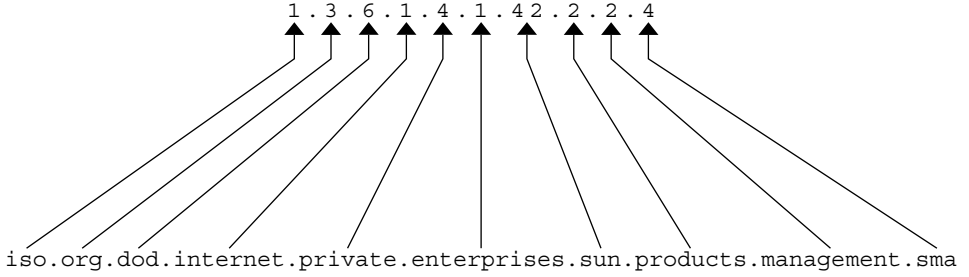

図 1–3 ISO 名前空間ツリーの図

サポートされる MIB

システム管理エージェントは次の MIB をサポートしています。

SNMP-COMMUNITY MIB
　RFC 2576 で定義されています。 http://www.ietf.org/rfc/rfc2576.txt を参照してください

SNMPv2-TM (トランスポートのマッピング)
　RFC 3417 で定義されています。http://www.ietf.org/rfc/rfc3417.txt を参照してください

SNMP-MPD-MIB (メッセージ処理と振り分け)
　RFC 3412 で定義されています。http://www.ietf.org/rfc/rfc3412.txt を参照してください

SNMP-TARGET-MIB (トラップのターゲットの仕様)
　RFC 3413 で定義されています。http://www.ietf.org/rfc/rfc3413.txt を参照してください

SNMP-NOTIFICATION-MIB (トラップフィルタリング)
　RFC 3413 で定義されています。http://www.ietf.org/rfc/rfc3413.txt を参照してください

SNMP-PROXY-MIB (トラップ転送)
RFC 3413 で定義されています。http://www.ietf.org/rfc/rfc3413.txt を参照してください

SNMP-USER-BASED-SM-MIB (SNMPv3 用のユーザーに基づくセキュリティーモデル)
RFC 3414 で定義されています。http://www.ietf.org/rfc/rfc3414.txt を参照してください

SNMP-VIEW-BASED-ACM-MIB (SNMP 用のビューに基づくアクセス制御モデル)
RFC 3415 で定義されています。http://www.ietf.org/rfc/rfc3415.txt を参照してください

SNMPv2-MIB
RFC 3418 で定義されています。http://www.ietf.org/rfc/rfc3418.txt を参照してください

MIB II
RFC 1213 で定義されています。http://www.ietf.org/rfc/rfc1213.txt を参照してください

ホストリソース MIB
RFC 2790 で定義されています。http://www.ietf.org/rfc/rfc2790.txt を参照してください

Sun MIB
Solstice Enterprise Agents ソフトウェアからの移行に関連しています。詳細については、67 ページの「Solstice Enterprise Agents ソフトウェアからの移行」を参照してください。Solstice Enterprise Agents ソフトウェアからのアプリケーションの移行については、『Solaris System Management Agent Developer's Guide』の第 10 章「Migration of Solstice Enterprise Agents to the System Management Agent」を参照してください。

システム管理エージェントの起動後に初期化されるMIB の一覧については、35 ページの「初期化された MIB を確認するには」に記述されている手順に従うと参照できます。

MIB 定義のテキストファイルは、`/etc/sma/snmp/mibs/` にあります。

◆ ◆ ◆ 第 2 章

システム管理エージェントの構成

この章では、ネットワーク内で使用できるようにシステム管理エージェントを構成する方法について説明します。この章では、システム管理エージェントの構成ファイルやセキュリティー機能について取り上げます。この章で扱うトピックは次のとおりです。

プラットフォームとパッケージ

システム管理エージェントは、今回の Solaris リリースにバンドルされています。システム管理エージェントを Solaris ソフトウェア上にインストールするには、バンドル製品の標準インストール手順に従ってください。これらの手順は、『Solaris 10 インストールガイド (基本編)』および『Solaris 10 インストールガイド (カスタム JumpStart/ 上級編)』に記載されています。以前にシステム管理エージェントのパッケージをインストールし、削除したことがある場合は、`pkgadd` コマンドだけで再インストールできます。

SMA パッケージは、2 つの部分に分割されます。これは、Solaris 10 オペレーティングシステムが SPARC プラットフォームと x86 プラットフォームでサポートされるためです。

SMA の実行時用製品には、次のような独自のパッケージが付属しています。

`SUNWsmaS` `SUNWsmaS` パッケージには、システム管理エージェントの再構築に必要なソースファイルが格納されています。Net-SNMP バージョン 5.0.9 のソースコードは、これらのソースファイルで構成されます。`SUNWsmaS` 内のファイルは、軽量デーモンを構成する上で役立ちます。`SUNWsmaS` はソースコードパッケージなので、デフォルトでは、Solaris ソフトウェアと同時にインストールされることはありません。このパッケージをインストールするには、CD をマウントし、次の `pkgadd` コマンドを使用します。

```
# pkgadd -d /sol_10_sparc_4/Solaris_10/Product SUNWsmaS
```

`SUNWsmagt` `SUNWsmagt` パッケージには、32 ビットと 64 ビットのライブラリが格納されています。このパッケージには、 `snmpd` エージェントと `snmptrapd` トラップデーモンも格納されています。このパッケージには、さらに、SMA の構築に必要なヘッダーファイルも格納されています。

`SUNWsmcmd` `SUNWsmcmd` パッケージには、SMA SNMP アプリケーションおよびユーティリティーが格納されています。これらのアプリケーションおよびユーティリティーには、`snmpget` などの開発ツールと、`mib2c` などの Perl スクリプトが含まれます。`SUNWsmcmd` パッケージには、SDK デモモジュールも格納されています。SDK デモモジュールは、一部のデータモデリングの実装方法を示します。デモモジュールの詳細については、『Solaris System Management Agent Developer's Guide 』を参照してください。

`SUNWsmdoc` `SUNWsmdoc` パッケージには、SMA の HTML 文書ファイルが格納されています。これらのファイルは Net-SNMP ソースから生成されています。これらの生成された HTML ファイルを、SMA の製品マニュアルと混同しないでください。SMA の製品マニュアルには、本書に加え、『Solaris System Management Agent Developer's Guide 』とマニュアルページが含まれます。この製品マニュアルは Sun によって提供されています。

`SUNWsmmgr` `SUNWsmmgr` パッケージには、`/etc/sma` にインストールされる全ファイルが格納されています。次のものが含まれます。

- すべての MIB。詳細については、22 ページの「サポートされる MIB」を参照してください。
- デフォルトの `snmpd.conf` ファイル。詳細については、28 ページの「構成ファイルと構成スクリプト」を参照してください。
- `mib2c` 関連のヘルパースクリプト。`mib2c` の詳細については、『Solaris System Management Agent Developer's Guide 』を参照してください。

パッケージの削除

この章で紹介したパッケージを削除すると、システム管理エージェント関連のファイルがすべて削除されます。

注 – アンインストールを実行する前に、システム管理エージェントを停止してください。エージェントを停止せずにパッケージを削除すると、パッケージを削除したあとも、あちこちにエージェントファイルがインストールされたままになる可能性があります。エージェントの初回起動時に作成されたファイルを削除するには、エージェントを停止してからパッケージを削除してください。エージェントの停止の詳細については、33 ページの「システム管理エージェントの起動と停止」を参照してください。

どんなパッケージでも、アンインストールするときはスーパーユーザーでログインする必要があります。その後、次の手順に従ってパッケージをアンインストールします。

システム管理エージェントを停止してからこれらの SUNW パッケージを削除すれば、次のファイルとその持続ストア (存在する場合) も削除されます。

- /etc/sma/snmp/snmptrapd.conf
- /etc/sma/snmp/snmp.conf
- /var/sma_snmp/snmp.conf
- /var/sma_snmp/snmptrapd.conf

▼ システム管理エージェントのパッケージをアンインストールするには

1. スーパーユーザーで、**SMA** サービスを停止します。

   ```
   # svcadm disable svc:/application/management/sma:default
   ```

2. SUNWsmaS パッケージを削除します。

   ```
   # pkgrm SUNWsmaS
   ```

3. SUNWsmdoc パッケージを削除します。

   ```
   # pkgrm SUNWsmdoc
   ```

4. SUNWsmcmd パッケージを削除します。

   ```
   # pkgrm SUNWsmcmd
   ```

5. SUNWsmmgr パッケージを削除します。

   ```
   # pkgrm SUNWsmmgr
   ```

6. SUNWsmagt パッケージを削除します。

   ```
   # pkgrm SUNWsmagt
   ```

ソフトウェアのデフォルトの場所

システム管理エージェントのデーモンは、snmpd と呼ばれます。このデーモンは、/usr/sfw/sbin/ ディレクトリにあります。

システム管理エージェントのトラップデーモンは、snmptrapd と呼ばれます。このトラップデーモンは、/usr/sfw/sbin/ ディレクトリにあります。

トラップの使用後に、snmptrapd.conf ファイルが作成されます。

システム管理エージェントの主要構成ファイルは、 snmpd.conf という名前です。この構成ファイルは、デフォルトで /etc/sma/snmp/ ディレクトリにインストールされます。snmpd.conf ファイルの詳細については、28 ページの「構成ファイルと構成スクリプト」を参照してください。

snmpd デーモンの起動時に、次のファイルが作成されます。

- /etc/sma/snmp/mibs/.index
- /var/log/snmpd.log
- /var/sma_snmp/snmpd.conf (持続ファイル)

32 ビット x86 プラットフォームのライブラリファイルは、/usr/sfw/lib ディレクトリに格納されています。

64 ビット SPARC プラットフォームのライブラリファイルは、/usr/sfw/lib/sparcv9 ディレクトリに格納されています。

構成スクリプトとその他のコマンド群は、/usr/sfw/bin に格納されています。

構成ファイルと構成スクリプト

システム管理エージェントは、標準 Net-SNMP を実装しているため、主にユーザー構成ファイル snmpd.conf を使って構成できます。このファイルの使用方法については、30 ページの「主要構成ファイルによる構成の管理」を参照してください。システム管理エージェントで使用するデフォルト設定を構成できるように、snmp.conf という名前の構成ファイルが個別に提供されています。このファイルについては、付録 A を参照してください。

一部の構成ファイル、スクリプト、およびマニュアルページの名前は非常に似通っています。次の一覧で、これらの違いをわかりやすくまとめます。

snmpconf　SMA 構成ファイルの作成と変更に役立つスクリプト。SMA snmpconf スクリプトは /usr/sfw/bin/ ディレクトリに格納されています。関連するマニュアルページは snmpconf(1M) です。

snmp.conf　システム管理エージェントの構成ファイル。アプリケーションの動作を定義します。このファイルを使ってデフォルト設定を構成すると、SNMPv3 のデフォルトユーザーの定義時など、SNMP コマンドを使用する際に必要な引数の数が少なくて済みます。関連マニュアルページは snmp.conf(4) です。

`snmpd.conf` 今回の Solaris リリースでは、`snmpd.conf` という名前のファイルが複数存在します。次のファイルがあります。

- もっとも重要な `snmpd.conf` ファイルは、システム管理エージェントの動作を制御する構成ファイルです。このファイルは `/etc/sma/snmp` ディレクトリに格納されています。関連マニュアルページは `snmpd.conf(4)` です。
- Solstice Enterprise Agents 構成ファイルにも `snmpd.conf` という名前が付けられています。このファイルは `/etc/snmp/conf` ディレクトリに格納されています。
- SMA の持続的記憶領域ファイルにも `snmpd.conf` という名前が付けられています。この持続的記憶領域ファイルは `/var/sma_snmp/` ディレクトリに格納されています。このファイルについては、29 ページの「持続的記憶領域ファイル」を参照してください。
- Sun Fire™ サーバーの移行スクリプトで使用されるテンプレートファイルにも、`snmpd.conf` という名前が付けられています。このテンプレートファイルは `/usr/sfw/lib/sma_snmp/` ディレクトリに格納されています。Sun Fire サーバーは、移行スクリプトを使ってシステム管理エージェントの主要構成ファイル `snmpd.conf` に変更を加えます。Sun Fire サーバーの管理エージェントから SMA への移行の詳細については、72 ページの「Sun Fire Management Agent からの移行」を参照してください。

`snmpd` この Solaris リリースには、次の2 つの `snmpd` デーモンが提供されています。

- システム管理エージェントの `snmpd` デーモン。SMA ソフトウェアの要求を実行する SNMP エージェントです。SMA `snmpd` デーモンは `/usr/sfw/sbin/` ディレクトリに格納されています。関連するマニュアルページは `snmpd(1M)` です。
- Sun SNMP Management Agent for Sun Fire and Netra Systems が使用するエージェントにも、`snmpd` という名前が付けられています。

`sma_snmp` システム管理エージェントの全般を紹介するマニュアルページ。`sma_snmp(5)` マニュアルページのエイリアス、つまり別名は、`netsnmp(5)` です。

`snmp_config` `snmp_config(4)` のマニュアルページは、システム管理エージェントの構成ファイル `snmpd.conf` の概要を示します。

詳細については、77 ページの「マニュアルページ」を参照にしてください。

持続的記憶領域ファイル

持続的記憶領域ファイル `/var/sma_snmp/snmpd.conf` には、USM セキュリティー情報と、持続的記憶領域用に設定された MIB コンポーネントが含まれています。このファイルに

は、engineID と engineID ブートも含まれています。この持続的記憶領域ファイルは、システム管理エージェントの起動時に自動的に更新されます。システム管理エージェントが停止すると、snmpusm および snmpvacm ユーティリティーにより、この記憶領域ファイルにユーザーセキュリティー情報が書き込まれます。セキュリティーの詳細については、第 4 章を参照してください。

持続的記憶領域ファイルは、/etc/sma/snmp/snmpd.conf にある主要ユーザー構成ファイル内のトークンに基づいて生成されます。詳細については、snmpd.conf(4) のマニュアルページを参照してください。

主要構成ファイルによる構成の管理

システム管理エージェントの主要構成ファイルは snmpd.conf です。このファイルは、/etc/sma/snmp ディレクトリに格納されています。作業をスムーズに開始できるように、標準テンプレートとして、最小限の情報を含むファイルが用意されています。

標準 Net-SNMP と同様に、構成管理用の各種トークンを使用できます。snmpd.conf ファイルを使って、これらのトークンを管理します。各トークンには、システム管理エージェントの起動時に実行される init モジュールがあります。

システム管理エージェントには、標準 Net-SNMP 実装のほかに、いくつかの追加モジュールが付属しています。追加モジュールには、seaProxy モジュールおよび seaExtensions モジュールがあります。これらのモジュールについては、67 ページの「Solstice Enterprise Agents ソフトウェアからの移行」を参照してください。

snmpd.conf ファイルの詳細については、snmpd.conf(4) のマニュアルページを参照してください。

システム管理エージェントは SNMP エージェントであるため、ポート 161 で実行する必要があります。ポート 161 でほかのプロセスを実行中の場合、システム管理エージェントは起動しません。システム管理エージェントを起動できない理由が、ポート 161 で別のエージェントが実行されているためかどうかを確認するには、/var/log/snmpd.log ログファイルをチェックします。起動時にその他のエラーが発生した場合も、このログファイルに詳細が記載されます。

AgentX プロトコルの使用

システム管理エージェントは、AgentX プロトコルをサポートします。システム管理エージェントには、デフォルトで、セキュリティー保護されたプロファイル (読み取り専用アクセス) が付属しています。AgentX は、UNIX ドメインソケット上で AgentX をサポートする、サードパーティーのサブエージェントとの通信を可能にします。セキュリティー上の理由から、TCP/UDP 上では、AgentX はサポートされません。AgentX プロトコルの詳細については、http://www.ietf.org/rfc/rfc2741.txt を参照してください。

システム管理エージェントで AgentX プロトコルを使用するには、主要構成ファイル /etc/sma/snmp/snmpd.conf を編集します。デフォルトでは、AgentX プロトコルは無効になっています。AgentX プロトコルを有効にするには、次の手順を実行します。

▼ AgentX プロトコルを有効にするには

1. スーパーユーザーで、主要構成ファイル /etc/sma/snmp/snmpd.conf を編集します。
   次の行を追加します。

   ```
   master agentx
   ```

2. システム管理エージェントを再起動します。

   ```
   # svcadm restart svc:/application/management/sma:default
   ```

参照　AgentX プロトコルには、さまざまなオプションを設定できます。たとえば、AgentX 要求のタイムアウト時間を設定できます。これらのオプションについては、snmpd.conf(4) のマニュアルページを参照してください。

◆ ◆ ◆ 第 3 章

3 システム管理エージェントの操作

この章では、システム管理エージェントの一般的な操作方法について説明します。この章で扱うトピックは次のとおりです。

システム管理エージェントの起動と停止

SMA を起動または停止するには、`snmpd` デーモンを起動または停止します。このデーモンを起動または停止する方法は多数ありますが、一部の方法は、`snmpd.conf` ファイルおよび `snmp.conf` ファイルを無効にします。システム管理エージェントを起動または停止するときは、この節で説明するように、`svcadm` コマンドを利用することをお勧めします。`snmpd` デーモンの詳細については、`snmpd(1M)` のマニュアルページを参照してください。

注 – システム管理エージェントは標準 SNMP エージェントであるため、ポート 161 で実行する必要があります。ポート 161 でほかのプロセスを実行中の場合、システム管理エージェントは起動しません。

▼ システム管理エージェントを起動するには

システムでいったん `svcadm` コマンドを使用してシステム管理エージェントを起動したあとは、Solaris システムがブートされるたびに `snmpd` デーモンが起動するようになります。その他のエージェントを使用している場合、ブート時に `snmpd` デーモンが起動しないように設定して、システム管理エージェントが初期化されるのを防ぐこともできます。ブート時に `snmpd` デーモンが起動するのを防ぐ方法については、68 ページの「ブート時にシステム管理エージェントが初期化されるのを防ぐには」を参照してください。

1. スーパーユーザーで、**SMA** サービスを起動します。

   ```
   # svcadm enable svc:/application/management/sma:default
   ```

2. /var/log/snmpd.log ファイルの内容から、システム管理エージェントの起動時にエラーが発生していないかどうかをチェックします。

   このログファイルに、ポート 161 が使用中であることが報告されている場合は、34 ページの「SMA ポート上でその他のプロセスが実行中でないかどうかをチェックするには」に記載されている手順を実行します。

▼ システム管理エージェントを再起動するには

SMA の主要構成ファイル /etc/sma/snmp/snmpd.conf への変更を有効にするには、SMA デーモン snmpd へシグナルを送信する必要があります。このシグナルにより、snmpd.conf への変更が読み取られ、システム管理エージェントが再起動します。

- スーパーユーザーで、**SMA** を再起動します。

  ```
  # svcadm restart svc:/application/management/sma:default
  ```

  システム管理エージェントの再起動は、この方法で行うことをお勧めします。

▼ システム管理エージェントを停止するには

- スーパーユーザーで、**SMA** サービスを停止します。

  ```
  # svcadm disable svc:/application/management/sma:default
  ```

システム管理エージェントでの一般的な操作

▼ SMA ポート上でその他のプロセスが実行中でないかどうかをチェックするには

ポート 161 はシステム管理エージェント用に予約されています。詳細については、30 ページの「主要構成ファイルによる構成の管理」を参照してください。

- netstat コマンドを使用します。

  ```
  # netstat -anv|grep 161
  ```

  値 161 が返された場合、ポート 161 上で実行中のプロセスが存在します。

▼ エージェントの状態を表示するには

◗ スーパーユーザーで、サービスの状態を取得します。

```
# svcs svc:/application/management/sma:default
```

このコマンドに対する一般的な応答は、次のようになります。

```
STATE          STIME    FMRI
online         Aug_24   svc:/application/management/sma:default
```

▼ 初期化された MIB を確認するには

◗ SMA の起動時に初期化された MIB は、次のいずれかの方法で一覧できます。

- 次のコマンドを実行して、生成されたデバッグトレースを調べます。

  ```
  # /usr/sfw/sbin/snmpd  -Dregister_mib -Dmib_init -L
  ```

- 別の方法として、net-snmp-config コマンドで、コンパイルされたモジュールの一覧を表示することもできます。

  ```
  # /usr/sfw/bin/net-snmp-config --snmpd-module-list
  ```

▼ ローカルマシンまたはリモートマシン上のディスク容量をチェックするには

まずディスクの総ディスク容量を確認し、次にこの容量のうちどの程度が使用されているかを確認します。この 2 つの容量の差が使用可能なディスク容量になります。

1 指定のホスト上で使用可能なディスクの数を確認します。

```
# snmpwalk -v1 -c public hostname HOST-RESOURCES-MIB::hrStorageIndex
```

このコマンドは、ホスト *hostname* 上のディスクの一覧を返します。

```
HOST-RESOURCES-MIB::hrStorageIndex.1 = INTEGER: 1
HOST-RESOURCES-MIB::hrStorageIndex.2 = INTEGER: 2
HOST-RESOURCES-MIB::hrStorageIndex.3 = INTEGER: 3
HOST-RESOURCES-MIB::hrStorageIndex.4 = INTEGER: 4
HOST-RESOURCES-MIB::hrStorageIndex.5 = INTEGER: 5
HOST-RESOURCES-MIB::hrStorageIndex.6 = INTEGER: 6
HOST-RESOURCES-MIB::hrStorageIndex.7 = INTEGER: 7
HOST-RESOURCES-MIB::hrStorageIndex.8 = INTEGER: 8
HOST-RESOURCES-MIB::hrStorageIndex.9 = INTEGER: 9
HOST-RESOURCES-MIB::hrStorageIndex.10 = INTEGER: 10
```

```
HOST-RESOURCES-MIB::hrStorageIndex.101 = INTEGER: 101
HOST-RESOURCES-MIB::hrStorageIndex.102 = INTEGER: 102
```

ディスクには、次のようにインデックス番号が付けられています。

```
HOST-RESOURCES-MIB::hrStorageIndex.1 = INTEGER: 1
```

この出力はディスク 1 (`/dev/dsk/c0t0d0s0`) を表しています。

2 `snmpget` コマンドを使って、このディスクの総記憶容量を確認します。
次のコマンドは、ディスク 1 の総記憶容量を表示します。

```
# snmpget -v1 -c public hostname HOST-RESOURCES-MIB::hrStorageSize.1
```

このコマンドは、行末に総ディスク容量を返します。

```
HOST-RESOURCES-MIB::hrStorageSize.1 = INTEGER: 2561695
```

3 各ディスクが使用するディスク容量の一覧を確認します。

```
# snmpwalk -v1 -c public hostname HOST-RESOURCES-MIB::hrStorageUsed
```

```
HOST-RESOURCES-MIB::hrStorageUsed.1 = INTEGER: 2121747
HOST-RESOURCES-MIB::hrStorageUsed.2 = INTEGER: 0
HOST-RESOURCES-MIB::hrStorageUsed.3 = INTEGER: 0
HOST-RESOURCES-MIB::hrStorageUsed.4 = INTEGER: 0
HOST-RESOURCES-MIB::hrStorageUsed.5 = INTEGER: 11
HOST-RESOURCES-MIB::hrStorageUsed.6 = INTEGER: 48
HOST-RESOURCES-MIB::hrStorageUsed.7 = INTEGER: 1892576
HOST-RESOURCES-MIB::hrStorageUsed.8 = INTEGER: 0
HOST-RESOURCES-MIB::hrStorageUsed.9 = INTEGER: 130565552
HOST-RESOURCES-MIB::hrStorageUsed.10 = INTEGER: 26036932
HOST-RESOURCES-MIB::hrStorageUsed.101 = INTEGER: 55995
HOST-RESOURCES-MIB::hrStorageUsed.102 = INTEGER: 17171
```

4 `snmpget` コマンドを使って、当該ディスクにより使用されている容量を確認します。

```
# snmpget -v1 -c public hostname HOST-RESOURCES-MIB::hrStorageUsed.1
```

このコマンドは、ディスク 1 上で使用されているディスク容量を返します。

```
HOST-RESOURCES-MIB::hrStorageUsed.1 = INTEGER: 2121747
```

5 この数値と総ディスク容量の差から、使用可能なディスク容量がわかります。
2561695 – 2121747 = 439948

snmpnetstat コマンド

netstat コマンドと同様の方法で snmpnetstat コマンドを実行して、システム管理エージェントを使用しているネットワークの状態を確認できます。

すべてのソケットの状態を表示するには、snmpnetstat コマンドで –a オプションを指定します。このオプションを指定した場合、サーバープロセスによって使用されているソケットを除くすべてのアクティブなソケットが表示されます (デフォルトの表示)。

```
# snmpnetstat -v 2c -c public -a testhost
```

通常、ローカルアドレス、リモートアドレス、およびプロトコルを含む次のような情報が表示されます。

```
Active Internet (tcp) Connections (including servers)
     Proto Local Address Foreign Address (state)
     tcp   *.echo        *.*             LISTEN
     tcp   *.discard     *.*             LISTEN
     tcp   *.daytime     *.*             LISTEN
     tcp   *.chargen     *.*             LISTEN
     tcp   *.ftp         *.*             LISTEN
     tcp   *.telnet      *.*             LISTEN
     tcp   *.smtp        *.*             LISTEN

     Active Internet (udp) Connections
     Proto Local Address
     udp   *.echo
     udp   *.discard
     udp   *.daytime
     udp   *.chargen
     udp   *.time
```

ネットワークインタフェースの状態を表示するには、snmpnetstat コマンドで –i オプションを使用します。このオプションを指定した場合、転送パケット、エラー、衝突、インタフェースのネットワークアドレス、および最大転送単位 (MTU) を示す統計情報テーブルが表示されます。

```
# snmpnetstat -v 2c -c public -i testhost
```

通常、ローカルアドレス、リモートアドレス、およびプロトコルを含む次のような表が表示されます。

```
Name Mtu  Network   Address  Ipkts     Ierrs  Opkts  Oerrs Queue
     eri0 1500 10.6.9/24 testhost 170548881 245601 687976 0     0
     lo0  8232 127      localhost 7530982   0      7530982 0    0
```

注 – snmpnetstat コマンドが返す Ipkts (着信パケット数) の値は、netstat コマンドが返す値と同一ではありません。snmpnetstat コマンドは、ユニキャストパケット、マルチキャストパケット、ブロードキャストパケットの合計数を表示します。netstat コマンドは、ブロードキャストパケット数を除いた、ユニキャストパケットとマルチキャストパケットの合計数を表示します。

リソースの使用

snmpd デーモンの常駐サイズは、 SMA の使用方法によって異なります。

snmpd デーモンは、特定の MIB テーブルデータに動的にメモリーを割り当てます。たとえば、プリンタを定義したあと、ホスト資源 MIB を使用する場合などです。snmpd デーモンの常駐サイズは、定義済みのプリンタ数によって、最大 100K バイトの範囲で増加します。

JDMK の相互運用性

Java™ Development Management Kit (JDMK) は、JMX 仕様を実装し、JDMK エージェントインフラストラクチャー内で SNMP ベースの計測を可能にします。システム管理エージェントと同様に、JDMK も次の標準をサポートします。

- SNMPv3
- SNMPv2c
- SNMPv1
- USM
- プロキシ

JDMK は AgentX をサポートしません。

JDMK も SMA も SNMP 計測に対応していますが、JDMK は Java ベースの環境に非常に適しています。SMA は、ネイティブの C 言語による実装に、より適しています。

JDMK による構成とプロキシ

JDMK と SMA が共存している Sun システムでは、SMA はデフォルトでポート 161 に常駐します。JDMK エージェントは、SMA からプロキシを介して、SNMP MIB を公開できます。プロキシは SMA 内のプロキシ転送メカニズムを使って設定できます。例 3–1 を参照してください。

セキュリティーの処理は、マスターエージェント (SMA) とプロキシされた JDMK エージェントによって行う必要があります。プロキシ定義に含まれるセキュリティーパラメータは、プロキシされた JDMK エージェントに転送されます。要求が SMA の認証およ

び承認を経てプロキシハンドラに転送されると、ディスパッチされた要求が JDMK エージェントにプロキシされます。JDMK エージェントの独自のローカルデータストアは、このメッセージを承認または拒否します。

複数の JDMK エージェントが同一の MIB を持っている場合、同じ MIB の異なるインスタンスを区別するためには、SNMP コンテキストをプロキシとともに使用する必要があります。コンテキスト名はプロセス ID に基づいて決定される場合があります。また、コンテキスト名は JDMK エージェントの実行ポートに基づいて決定される場合もあります。

例 3–1 JDMK プロキシ文の追加

JDMK エージェントへの着信要求は、システム管理エージェントで受信され、JDMK エージェントにプロキシされます。主要構成ファイル `snmpd.conf` 内のプロキシトークンを使って、JDMK プロキシエントリを設定します。次のようなプロキシ文を追加します。

```
# proxy --Cn jdmkMib -v3 -a MD5 -u
 SecureUser -l authNopriv -A 12345678 localhost:10161  1.3.6.1.4.1.42.5000.2
```

この例では、MIB はポート 10161 上で実行されており、コンテキスト `jdmkMib` で MIB リージョン 1.3.6.1.4.1.42.5000 を登録します。ユーザー `SecureUser` も、JDMK USM 内に存在しなければなりません。ユーザー `SecureUser` は、認証アルゴリズム HMACMD5 を使って、セキュリティーレベル `auth` を許可する必要があります。認証アルゴリズムの詳細については、43 ページの「認証プロトコルアルゴリズム」を参照してください。

詳細については、`snmpd.conf(4)` のマニュアルページを参照してください。プロキシの設定方法については、69 ページの「Solstice Enterprise Agents 要求のプロキシ処理」を参照してください。

◆ ◆ ◆ 第 4 章

セキュリティーの管理

この章では、システム管理エージェントのセキュリティーに関する基礎的な情報と、各種手順および例を示します。システム管理エージェントでは、SNMPv3 の機能により、ユーザーおよびネットワークデバイスの管理のセキュリティーレベルが拡張され、構成可能です。

この章で扱うトピックは次のとおりです。

セキュリティーの概要

システム管理エージェントは、SNMPv1、SNMPv2c、および SNMPv3 をサポートします。SNMPv1 と SNMPv2c の認証サービスは、管理ステーション上で定義されたコミュニティー文字列に基づいています。SNMPv3 認証サービスは、ユーザーに基づいています。各要求には、コミュニティー名またはユーザー名のどちらか (使用するプロトコルによる) が含まれていなければなりません。

SNMPv3 認証処理は、ユーザーに基づくセキュリティーモデル (USM) を実装しており、ユーザー名からセキュリティー名とセキュリティーレベルを取得します。同様に、SNMPv1 と SNMPv2c は、コミュニティー文字列からセキュリティーレベルを決定します。セキュリティー名とセキュリティーレベルは、コンテキスト文字列、グループ名、およびビュー名とともに、アクセス制御の実行に使用されます。アクセス制御は、ビューに基づくアクセス制御モデル (VACM) を使って行われます。このアクセス制御モデルは、認証処理の完了後に使用されます。つまり、USM が認証用であるのに対し、VACM は承認用であると言えます。

システム管理エージェントでサポートされる SNMP のバージョンについては、16 ページの「SNMP のバージョン」を参照してください。

USM による認証とメッセージプライバシ

システム管理エージェントでは、SNMPv3 パケットの認証、暗号化、および復号化に、ユーザーに基づくセキュリティーモデル (USM) を使用します。USM の使用目的は次のとおりです。

- SNMP ユーザーの認証
- 通信のプライバシ
- メッセージの整合性
- 再生保護

`snmpusm` ユーティリティーは、SNMP エージェントの USM テーブルの基本管理用 SNMP アプリケーションです。`usmUserTable` MIB テーブルへの書き込みアクセスが必要です。詳細については、`snmpusm(1M)` のマニュアルページを参照してください。

注 – `snmpusm` サブコマンドは、SNMPv1 および v2c ではサポートされません。プロキシなしでこれらのコマンドを実行できるのは、SNMPv3 ユーザーだけです。

エージェントの働きにより、ユーザーは、主要構成ファイル `snmpd.conf` と `snmpusm` コマンドを利用して、USM MIB を通してユーザーエントリを管理できます。システム管理エージェントは、USM MIB を利用して、ユーザー情報 (ユーザーが存在するかどうかの情報も含む) を検索します。ユーザーからの要求はすべて、USM MIB と照合されます。ユーザーが存在する場合、USM MIB は次のアクセス権をチェックします。

- ユーザーが認証済み要求を許可されているかどうか
- 許可されている認証符号化の種類

USM MIB は、ローカルストアキーを使って、MIB 内の特定のユーザーによって指定された認証プロトコルに基づく新しいダイジェストを計算します。計算されたダイジェストは、着信パケットから保存されたダイジェストと比較されます。2 つのダイジェストが一致すれば、そのユーザーは認証されます。メッセージダイジェストの詳細については、43 ページの「認証プロトコルアルゴリズム」を参照してください。

以下では、USM の設定について説明します。

ユーザー	SNMP エンジンとの通信を許可されたユーザーを表す
認証の型	使用する認証プロトコルアルゴリズム を指定する(MD5 または SHA)。詳細については、43 ページの「認証プロトコルアルゴリズム」を参照
認証パスワード	ユーザーの認証パスワードを指定する。パスワードは 8 文字以上で構成する必要がある。
プライバシの型	使用するプライバシプロトコルを指定する。システム管理エージェントの場合、DES (データ暗号化規格) が使用される。詳細については、44 ページの「メッセージプライバシ」を参照

プライバシパスワード　ユーザーのプライバシパスワードを指定する。パスワードは 8 文字以上で構成する必要がある。

セキュリティーレベル　認証とプライバシに関するユーザーのセキュリティーレベルを指定する。

- `noAuthNoPriv`　ユーザー名の一致だけを使って認証を行う
- `authNoPriv`　MD5 または SHA1 アルゴリズムに基づいた認証を行う
- `authPriv`　DES プロトコルに基づいたプライバシ (暗号化) を提供する

認証では、秘密鍵を使って MAC (メッセージ認証コード) を生成し、`usmSecurityParameters` を構成する `msgAuthenticationParameters` に格納します。送信側と受信側のエンティティーは、同じ秘密鍵を使って MAC を生成します。このため、送信側の MAC と受信側の MAC が一致すれば、メッセージは認証されます。

認証プロトコルアルゴリズム

システム管理エージェントに実装されている USM では、2 つの認証プロトコルがサポートされます。以下のリストに認証プロトコルを示します。

HMAC-MD5–96　システム管理エージェントでは、メッセージダイジェスト実装は HMAC-MD5–96。これは、MD5 に基づく単方向の暗号化であり、96 ビットのハッシュと16 オクテットのキー長を使用する。計算上、2 つのメッセージが同じメッセージダイジェストを持つことはできない。事前に指定されたターゲットメッセージダイジェストからメッセージを生成することもできない。MD5 アルゴリズムは電子署名アプリケーションを対象にしている。これらのアプリケーションでは、サイズの大きいファイルを安全に圧縮する必要がある。圧縮後、公開鍵暗号システムにより、秘密鍵を使って暗号化する。HMAC-MD5–96 アルゴリズムは 32 ビットマシンで使用可能。大規模な置換テーブルは不要。このアルゴリズムは非常にコンパクトにコード化することができる。MD5 の詳細については、RFC 1321 (http://www.ietf.org/rfc/rfc1321.txt) を参照

HMAC-SHA–96　システム管理エージェントでは、セキュアハッシュアルゴリズム (SHA) 実装は HMAC-SHA–96。この単方向暗号化は、96 ビットのハッシュと 20 オクテットのキー長を使用する。このアルゴリズムは、2^{64} ビット未満の長さのメッセージを入力として受け付ける。入力メッセージは 512 ビットブロックで処理される。このアルゴリズムは 160 ビットのメッセージダイジェスト出力を生成する。このメッセージダイジェストは、たとえば、メッセージの署名を生成または検証する署名アルゴリズムへの入力として使用される。メッセージダイジェストには、メッセージそのものではなく署名が

付いている。このため、元のメッセージよりサイズが小さくなるため、効率がよくなる。電子署名の作成者が SHA を使用している場合は、検証側でも SHA を使用する必要がある。通常、転送中にメッセージに変更が加えられた場合は、メッセージダイジェストにも変更が加えられる。その結果、電子署名の検証は失敗する。SHA の安全性が高いのは、計算上、2 つのメッセージが同じメッセージダイジェストを持つことができないためである。また、事前に指定されたターゲットメッセージダイジェストからメッセージを生成することもできない。SHA の設計は MD5 ファミリのハッシュ関数と類似している。SHA の詳細については、RFC 3174 (http://www.ietf.org/rfc/rfc3174.txt) を参照

システム管理エージェントの場合、デフォルトの認証プロトコルは HMAC-MD55–96 です。設定は `auth proto = MD5` です。

メッセージプライバシ

USM はメッセージのプライバシをサポートします。USM は、SNMPv3 パケットの暗号化と復号化に、CBC-DES 対称暗号化プロトコルを使用します。認証の場合と同様に、送信側でのメッセージの暗号化と受信側でのメッセージの復号化には、同じ秘密鍵を使用します。データ部分だけが暗号化されます。暗号化を使用するには、`auth` フラグを有効にし、セキュリティーレベルでプライバシを有効にする必要があります。`scopedPDU` だけが暗号化されます。詳細については、45 ページの「USM セキュリティー情報の格納場所」を参照してください。

現在、Solaris OS では、DES 暗号化がサポートされています。DES 暗号化では 56 ビットの鍵暗号化を使用します。これが、今回の Solaris ソフトウェアリリースの DES でサポートされている、現段階で最高レベルの暗号化です。

公開鍵

システム管理エージェントは、公開鍵暗号化標準 (PKCS) #11 をサポートします。このトークンインタフェース標準は、SSL モジュールと PKCS #11 モジュールの間の通信に使用するインタフェースを定義します。システム管理エージェントでコンパイルされた PKCS ライブラリは、PKCS#11 v2.11 標準に基づいています。

43 ページの「認証プロトコルアルゴリズム」に記述されているように、MD5 に加えて SHA1 アルゴリズムがサポートされます。システム上に PKCS ライブラリがない場合、SMA は DES のサポートなしで標準 Net-SNMP 内部 MD5 の使用を試みます。

USM セキュリティー情報の格納場所

USM 情報は、SNMPv3 パケット文字列内の次のフラグに含まれています。

`msgFlags` メッセージの処理方法を指定する 8 ビット。たとえば、`msgFlags` の 8 ビット中 2 ビットは、パケットが暗号化されているかどうかと、パケットが認証されているかどうかを指定する。このフラグを使って、メッセージのセキュリティーレベルが決定される。セキュリティーレベルは、主要ファイル `snmpd.conf` で次のように指定する

- `noAuthNoPriv` 次の整数で表される。1。 最小限のアクセス
- `authNoPriv` 次の整数で表される。2。 `noAuthNoPriv` 以上、 `authPriv` 以下のアクセス権
- `authPriv` 次の整数で表される。3。 最大限のアクセス。もっとも安全

`msgSecurityModel` メッセージの生成に使用するセキュリティーモデルを指定して、受信側エンティティーが適切なセキュリティー処理モデルを利用できるようにする。システム管理エージェントでサポートされているセキュリティーモデルは USM のみ

`msgSecurityParameters` セキュリティーモデルに関するデータを含む 8 ビット文字列。このデータは使用するセキュリティーモデル (複数可) によって定義される。このデータは使用するセキュリティーモデル専用として使用される。セキュリティーモデルは `msgSecurityModel` に指定される。USM は、このフィールドを使って、SNMPv3 メッセージの認証、暗号化、および復号化を行う

`scopedPDU` 標準のプロトコルデータユニット (`PDU`) と、この PDU の処理に使用される管理上一意のコンテキストの識別情報を含む。SNMPv2 メッセージと SNMPv3 メッセージは、どちらも同じ PDU フォーマットを使用する。トランザクションのプライバシが有効になっている場合、この `scopedPDU` 形式は暗号化される

USM の MIB 定義は、`/etc/sma/snmp/mibs/SNMP-USER-BASED-SM-MIB.txt` にあります。

USM の詳細については、RFC 3414 (http://www.ietf.org/rfc/rfc3414.txt) を参照してください。

VACM によるアクセス制御

ビューに基づくアクセス制御モデル (VACM) を使って、特定の管理対象オブジェクトへのアクセスが承認されているかどうかを調べることができます。アクセス制御は、次のタイミングで行われます。

- マネージャーからの取得要求メッセージを処理するとき
- マネージャーからの変更要求メッセージを処理するとき
- マネージャーに通知メッセージを送信する必要があるとき

VACM は、18 ページの「コミュニティー文字列」で説明されているコミュニティー文字列の概念に基づいています。VACM では、アクセス制御をシステム管理エージェントで簡単に管理できます。

アクセス制御は、主要構成ファイル `snmpd.conf` 内のトークンによって定義されます。SMA デーモン `snmpd` が認識する VACM アクセスセキュリティー用のトークンを、以下に示します。これらのトークンは主要構成ファイル `snmpd.conf` 内で使用できます。

- `group`
- `access`
- `view`
- `com2sec`

最初の 3 つのトークンについては、48 ページの「VACM テーブルについて」を参照してください。`com2sec` トークンには、NAME SOURCE オプションと COMMUNITY オプションを指定できます。このトークンを使って、SNMPv1 や SNMPv2 のユーザーおよびコミュニティーに、SNMPv3 のセキュリティー特権を付与することができます。`com2sec` トークンは、ソースとコミュニティーのペアからセキュリティー名へのマッピングを示します。

`snmpd.conf` ファイルは、より高速で有用なラッパーを提供しています。これらのラッパーは、SMA `snmpd` エージェントで認識可能です。これらのラッパーは、ユーザーやコミュニティー向けの読み取り書き込み (rw) および読み取り専用 (ro) 構文を使用して、次のように定義されています。

- `rwuser`
- `rouser`
- `rwcommunity`
- `rocommunity`

`rwuser` トークンエントリは、ユーザーが指定しなければならない最小限のアクセス権を指定します。

`rwuser1 priv`	ユーザーはプライバシパスワードを指定しなければならない
`rwuser2 auth`	プライバシパスワードを使って作成されたユーザーはプライバシパスワードを指定できる。それ以外の場合、ユーザーは認証パスワードを指定しなければならない

`rwuser3 none` ユーザーは、パスワードを指定しないか、または認証パスワードを指定できる。また、プライバシパスワードを使って作成されたユーザーはプライバシパスワードを指定できる

VACM セキュリティー情報の格納場所

VACM 情報は、SNMPv3 パケット文字列内の複数のパラメータに含まれています。これらのパラメータは `isAccessAllowed` 機構に渡されます。`isAccessAllowed` 機構は、アクセスを付与すべきかどうかをチェックする VACM 内の唯一のエントリポイントです。

VACM パラメータを以下に示します。

`msgFlags` メッセージの処理方法を指定する 8 ビット。詳細については、45 ページの「USM セキュリティー情報の格納場所」を参照

`msgSecurityModel` メッセージの生成時に使用されたセキュリティーモデルを指定して、受信側エンティティーが適切なセキュリティー処理モデルを利用できるようにする。SNMPv3 では、単一のセキュリティーモデルの使用か複数のセキュリティーモデルの使用かを選択できる

`msgSecurityParameters` セキュリティーモデルに関するデータを含む 8 ビット文字列。セキュリティーモデル (複数可) は `msgSecurityModel` に指定される

`scopedPDU` PDU を含む。PDU 処理に使用する管理情報の管理上一意のセレクタを示す。つまり、`scopedPDU` にはコンテキストと管理対象オブジェクトの OID が含まれる。`scopedPDU` には次のフィールドがある

- `contextEngineID` コンテキスト内の管理対象オブジェクトのインスタンスにアクセスできる SNMP エンティティーを一意に識別する
- `contextName` PDU データの属するコンテキストの名前。`contextName` は一意
- `PDU` SNMPv3 のプロトコルデータユニット (`PDU`) には、`contextName` 内のデータの操作が含まれる。`contextEngineID` と `contextName` の組み合わせにより識別される

SNMPv3 パケット文字列のその他のフィールドについては、16 ページの「SNMP のバージョン」を参照してください。

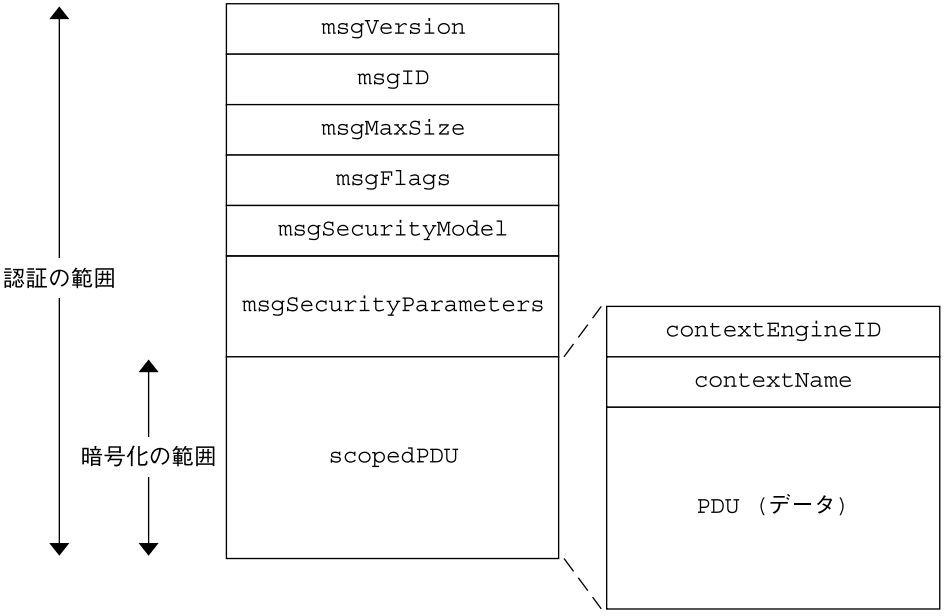

図 4–1 SNMPv3 パケットフォーマットと、認証と暗号化の範囲

VACM テーブルについて

メッセージにアクセスを付与するかどうかを決定する際、VACM は次の 4 つのテーブルを使用します。

- 49 ページの「コンテキストテーブル」
- 50 ページの「グループセキュリティーテーブル」
- 55 ページの「アクセステーブル」
- 51 ページの「ビューツリーファミリテーブル」

これらの VACM テーブルは、それぞれアクセス機構の特定の部分を処理します。各テーブルは、VACM MIB を使ってリモートで構成できます。VACM MIB は RFC 3415 (http://www.ietf.org/rfc/rfc3415.txt) で定義されています。

VACM は、SMA の USM によって認証された要求が、この要求に含まれる MIB オブジェクトへのアクセスを承認されるかどうかを判断します。`snmpvacm` ユーティリティーは、SNMP エージェントの VACM テーブルの基本管理用 SNMP アプリケーションです。`snmpvacm` ユーティリティーを使用するには、`snmpvacm` MIB テーブルの書き込みアクセス権が必要です。詳細については、`snmpvacm(1M)` のマニュアルページを参照してください。

この節では、各 VACM テーブルについて説明します。たとえば、テーブルの索引の付け方や、各行の内容などについて説明します。

コンテキストテーブル

vacmContextTable テーブルには、ローカルで使用可能なコンテキストが格納されています。コンテキストは、管理情報のセレクタです。単一の管理対象オブジェクトを複数のコンテキストで使用できます。たとえば、プリンタの状態を監視する単一のモジュールの場合を考えてみましょう。複数のプリンタが設置されているネットワークでは、このモジュールの複数のインスタンスにそれぞれ独自のプリンタ名を格納して実装できます。この場合、プリンタ名がコンテキストになります。

単一の SNMP エンティティーから複数のコンテキストにアクセスできます。

vacmContextTable テーブルは、 contextName で索引付けられています。各行には、読み取り可能な一意の文字列 *vacmcontextName* の形式でコンテキスト名が付けられます。

システム管理エージェントは、vacmContextTable 内で、scopedPDU 内の contextName を検索します。scopedPDUについては、16 ページの「SNMP のバージョン」を参照してください。システム管理エージェントが特定のメッセージの contextName を vacmContextTable 内で検出できない場合、アクセスは拒否されます。この場合、戻り値は noSuchContext になります。

contextName が見つかった場合、 図 4–2 のように、アクセスチェックが続行されます。例 4–1 に、vacmContextTable 内の一般的なエントリの例を示します。

例 **4–1** 一般的なコンテキストテーブルエントリの作成

モジュールにより作成される一般的な vacmContextTable エントリの例を、以下に示します。

```
SNMP-VIEW-BASED-ACM-MIB::vacmContextName."fileX" = STRING: fileX
SNMP-VIEW-BASED-ACM-MIB::vacmContextName."fileY" = STRING: fileY
```

この例の contextNames は fileX と fileY です。

コンテキストの詳細については、『Solaris System Management Agent Developer’s Guide 』を参照してください。システム管理エージェントには、コンテキストの概念を理解するのに役立つデモモジュールが付属しています。このデモモジュールでは、単一のモジュールの複数のインスタンスを実装する場合の、コンテキストの重要性を示しています。詳細については、『Solaris System Management Agent Developer’s Guide』の「Implementing Multiple Instances of a Module」を参照してください。

グループセキュリティーテーブル

vacmSecurityToGroupTable テーブルには、グループ情報が格納されています。ユーザーグループにはグループ名が付けられます。このグループ名は、アクセス権の管理に使用されます。グループには、SecurityModel と SecurityName の値のペアが含まれます。その結果、ペアは最大 1 つのグループにしかマッピングできません。vacmSecurityToGroupTable テーブルは、以下の項目で索引付けられています。

- securityModel
- securityName

vacmSecurityToGroupTable テーブルの各行には、次の情報が含まれます。

vacmSecurityModel SNMPv3 セキュリティーモデル。この例では USM。USM の詳細については、42 ページの「USM による認証とメッセージプライバシ」を参照。com2sec トークンを利用すると、SNMPv1 および v2c のセキュリティーモデルを使用できるようになる。com2sec トークンの詳細については、snmpd.conf(4) のマニュアルページを参照

vacmSecurityName USM では、*vacmSecurityName* と *userName* は一致する。セキュリティーモデルとは無関係のフォーマットでユーザーを表現する。com2sec トークンを利用すると、SNMPv1 および v2c のセキュリティー名を使用できるようになる。com2sec トークンの詳細については、snmpd.conf(4) のマニュアルページを参照

vacmGroupName 読み取り可能な文字列。このエントリに関連付けられているグループを示す

メッセージの認証と復号化に成功すると、SecurityName が msgSecurityModel 指示子によって取得されます。システム管理エージェントは、vacmSecurityToGroupTable テーブル内で、この msgSecurityModel 指示子および関連する SecurityName を検索します。vacmSecurityToGroupTable 内で msgSecurityModel 指示子および関連する SecurityName が見つからない場合、アクセスは拒否されます。この場合、戻り値は noSuchGroupName になります。

エントリが見つかった場合、対応する groupName が返されます。図 4–2 のように、アクセスチェックが続行されます。

例 4–2 に、vacmsecurityToGroupTable 内の一般的なエントリを示します。

例 4–2 一般的なグループセキュリティーテーブルエントリの作成

以前に作成したユーザー user2 および user5 のグループを作成します。この例では、ユーザーは、grpnam1 という新しく作成されたグループに配置されます。次のいずれかの方法を選択します。

- 主要構成ファイル /etc/sma/snmp/snmpd.conf に次の行を追加します。

```
group grpnam1 usm user2
group grpnam1 usm user5
```

主要構成ファイル /etc/sma/snmp/snmpd.conf に追加してグループを作成した場合、vacmSecurityToGroupTable テーブル内に次のエントリが作成されます。

```
SNMP-VIEW-BASED-ACM-MIB::vacmGroupName.3."user2" = STRING: grpnam1
SNMP-VIEW-BASED-ACM-MIB::vacmGroupName.3."user5" = STRING: grpnam1
SNMP-VIEW-BASED-ACM-MIB::vacmSecurityToGroupStorageType.3."user2" = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmSecurityToGroupStorageType.3."user5" =  INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmSecurityToGroupStatus.3."user2" = INTEGER: active(1)
SNMP-VIEW-BASED-ACM-MIB::vacmSecurityToGroupStatus.3."user5" = INTEGER: active(1)
```

リブートしてもエントリは削除されません。この VACM テーブルのエントリを削除するには、snmpvacm deleteGroup コマンドを使用します。この方法は、ストレージの型が nonVolatile の場合に使用できます。これ以外のストレージの型を持つ VACM テーブルエントリの場合は、主要構成ファイル /etc/sma/snmp/snmpd.conf からテーブルエントリを手動で削除する必要があります。主要構成ファイル /etc/sma/snmp/snmpd.conf を編集することで作成されたグループの場合、vacmSecurityToGroupTable テーブルのエントリを削除するには、主要構成ファイル /etc/sma/snmp/snmpd.conf を編集する必要があります。

- snmpvacm コマンドを使用します。user2 の場合、次のように snmpvacm コマンドを使ってグループを作成できます。

```
# snmpvacm -v3 -u myuser -a MD5 -A my_password -l authNoPriv localhost createSec2Group 3 user2 grpnam1
```

user5 の場合、次のように snmpvacm コマンドを使ってグループを作成できます。

```
# snmpvacm -v3 -u myuser -a MD5 -A my_password -l authNoPriv localhost createSec2Group 3 user5 grpnam1
```

ユーザー myuser のアクセスレベルは *rwuser* です。したがって、この例では、コンテキストに適していれば、グループエントリは myuser として作成されます。ユーザー user2 と user5 は、VACM テーブルを更新する権限を持ちません。

ビューツリーファミリテーブル

vacmViewTreeFamilyTable テーブルには、ビューサブツリーファミリの全コレクションが格納されます。これらのコレクションを「MIB ビュー」と呼びます。MIB ビューは、OID サブツリー値です。ファミリ名と、ファミリマスクであるビット文字列値から成り

ます。ファミリマスクは、MIB ビュー内に存在するファミリ名のサブ識別子を特定します。マスクは、次のいずれかで区切られた 16 オクテットのリストです。「.」または「:」。デフォルトは「ff」です。

MIB ビュー内の各ビューサブツリーファミリには、型があります。この型によって、そのビューサブツリーファミリが MIB ビューに含まれるかどうかが決まります。管理対象オブジェクトインスタンスは、次の条件が両方とも真である場合に限り、MIB ビューに含まれます。

- 管理対象オブジェクトの OID に、OID サブツリーと同数以上のサブ識別子が含まれている
- 対応するマスクのビットが 0 以外の場合、管理対象オブジェクトの OID サブ識別子が OID サブツリー内の対応するサブ識別子と一致する

マスクの構成値が短すぎてこれらの条件をチェックできない場合、暗黙のうちにこの値に 1 の連続が付加されます。したがって、マスクが 0 ビットのビューファミリサブツリーは、すべての値が 1 のマスクと同等であり、すなわち 1 つの MIB サブツリーと同等です。

`vacmViewTreeFamilyTable` テーブルは、以下の項目で索引付けられています。

`viewName`	`vacmAccessTable` テーブルで選択されたアクセス権によって指定される。アクセスチェックに使用される
MIB サブツリーの OID	PDU の OID が MIB ビューと比較される

`vacmViewTreeFamilyTable` テーブルの各行の値は次のとおりです。

vacmViewTreeFamilyViewName	MIB ビューの名前
vacmViewTreeFamilySubtree	OID サブツリー。OID サブツリーはマスクとともに MIB ビューサブツリーを構成する
vacmViewTreeFamilyMask	ビット文字列マスク。ビット文字列マスクは OID サブツリーとともに MIB ビューサブツリーを構成する
vacmViewTreeFamilyType	この型によって、そのビューサブツリーファミリが MIB ビューに含まれるかどうかが決まる

MIB ビューに検索対象の OID が含まれていない場合、アクセスは拒否されます。この場合、戻り値は `notInView` になります。MIB ビューに正しい OID が含まれていれば、アクセスは許可されます。この場合、戻り値は `accessAllowed` になります。

この VACM アルゴリズム全体のフローチャートは、図 4–2 のようになります。以下では、この図に含まれる RFC 推奨の用語について解説します。

`securityName` と `securityModel`
　アクセスの要求元を示す

`contextName`
　アクセスが許可される場所を特定する

`securityLevel` と `securityModel`
　アクセスを許可する方法を特定する

`viewType`
　読み取り、書き込み、通知のいずれか。グループまたはユーザーが特定のアクセスレベルを要求する理由を特定する

管理対象オブジェクトのオブジェクト型または OID
　チェック対象の管理データの型を示す

管理対象オブジェクトのインスタンス
　オブジェクト型と連携して、MIB ビュー内でチェックするインスタンスを特定する。選択肢は yes か no

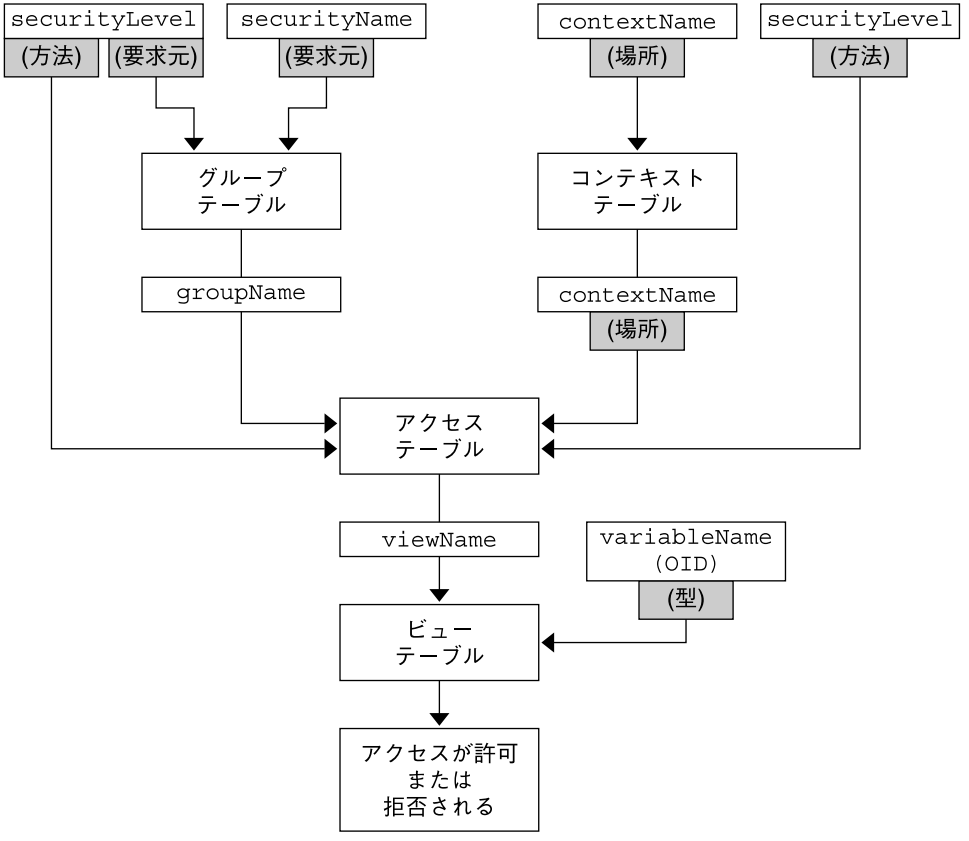

図 4–2 VACM 全体のフローチャート

例 4–3 に、`vacmViewTreeFamilyTable` の一般的なエントリを示します。

例 4–3 一般的なビューツリーファミリテーブルエントリの作成

ビューは次の 2 通りの方法で作成できます。

- 主要構成ファイル /etc/sma/snmp/snmpd.conf にビューを追加することで、ビューを作成できます。

```
view all included .1 FF
view none excluded .1 FF
view vwnam1 included .1.3.6.1 FF
```

この例では、マスクは「FF」になります。

主要構成ファイル /etc/sma/snmp/snmpd.conf にグループを追加することでビューを作成した場合、 vacmViewTreeFamilyTable テーブル内に次のエントリが作成されます。

```
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyMask."all".1.1 = STRING: "ÿ"
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyMask."none".1.1 = STRING: "ÿ"
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyMask."vwnam1".4.1.3.6.1
= STRING: "ÿ"
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyType."all".1.1
= INTEGER: included(1)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyType."none".1.1
 = INTEGER: excluded(2)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyType."vwnam1".4.1.3.6.1
 = INTEGER: included(1)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStorageType."all".1.1
 = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStorageType."none".1.1
 = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStorageType."vwnam1".4.1.3.6.1
 = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStatus."all".1.1
= INTEGER: active(1)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStatus."none".1.1
 = INTEGER: active(1)
SNMP-VIEW-BASED-ACM-MIB::vacmViewTreeFamilyStatus."vwnam1".4.1.3.6.1
 = INTEGER: active(1)
```

- snmpvacm コマンドを使ってビューを作成できます。

```
# snmpvacm -v3 -u myuser -a MD5
-A my_password -l authNoPriv -Ce localhost
createView all .1 FF

# snmpvacm -v3 -u myuser -a MD5
-A my_password -l authNoPriv localhost
createView none .1 FF
```

例 4–3 一般的なビューツリーファミリテーブルエントリの作成　(続き)

```
# snmpvacm -v3 -u myuser -a MD5
-A my_password -l authNoPriv localhost
createView vwnam1 .1.3.6.1 FF
```

ユーザー myuser のアクセスレベルは *rwuser* です。したがって、この例では、コンテキストに適していれば、myuser 用にビューエントリが作成されます。

snmpvacm コマンドを使って作成されたビューの場合、ストレージの型は nonVolatile になります。

アクセステーブル

vacmAccessTable テーブルには、各グループのアクセス権が格納されます。各グループに複数のアクセス権を付与できます。最も安全なアクセス権が選択されます。vacmAccessTable テーブルは、以下の項目で索引付けられています。

groupName	vacmSecurityToGroupTable テーブル内の検索から返される
contextPrefix	vacmContextTable テーブル内で一致する有効な contextName
securityModel	メッセージの msgSecurityModel パラメータ内に指定する
securityLevel	メッセージの msgFlags パラメータ内に指定する

vacmAccessTable テーブル内の各行には、次の値が含まれます。

vacmGroupName	グループ名。このグループには 1 つ以上のアクセス権がある
vacmAccessContextPrefix	contextName は *vacmAccessContextPrefix* の値と一致する必要がある。*vacmAccessContextMatch* を参照
vacmAccessSecurityModel	アクセス権の取得で使用する必要のあるセキュリティーモデルを示す
vacmAccessContextMatch	*vacmAccessContextMatch* が *exact* に設定されている場合、contextName は *vacmAccessContextPrefix* オブジェクトの値と正確に一致している必要がある *vacmAccessContextMatch* が *prefix* に設定されている場合、contextName は *vacmAccessContextPrefix* オブジェクトの値の最初の数文字と一致している可能性がある。この contextName は、vacmContextTable テーブル内ですでに一致している名前である
vacmAccessSecurityLevel	このアクセス権の使用に必要な最低限のセキュリティーレベルを示す。セキュリティーレベルについては、47 ページの「VACM セキュリティー情報の格納場所」を参照

vacmAccessReadViewName	読み取りアクセスが承認された MIB viewName。*vacmAccessReadViewName* が空の場合、読み取りアクセス用のアクティブなビューは存在しない
vacmAccessWriteViewName	書き込みアクセスが承認された MIB viewName。*vacmAccessWriteViewName* が空の場合、書き込みアクセス用のアクティブなビューは存在しない
vacmAccessNotifyViewName	通知アクセスが承認された MIB viewName。*vacmAccessWriteViewName* が空の場合、通知アクセス用のアクティブなビューは存在しない

アクセス権が見つからない場合、アクセスは拒否されます。この場合、戻り値は noAccessEntry になります。

アクセス権が選択されると、そのアクセス権により指定された viewName が選択されます。この viewName は PDU によって決定されます。PDU 内の SNMP 操作が GETNEXT または GET である場合、文字列 *vacmAccessReadViewName* が使用されます。PDU 内の SNMP 操作が TRAP である場合、文字列 *vacmAccessNotifyViewName* が使用されます。viewName が構成されていない場合、アクセスは拒否されます。この場合、戻り値は noSuchView になります。

正しく構成された viewName でアクセス権が選択されると、図 4–2 のように、引き続きアクセスチェックが続行されます。例 4–4 に、一般的なアクセステーブルエントリを示します。

例 4–5 では、この例と前の例のユーザー設定が存在し、VACM テーブル内に登録されているかどうかをチェックする方法を示します。

例 4–4 一般的なアクセステーブルエントリの作成

アクセステーブルエントリは次の 2 通りの方法で作成できます。

- 主要構成ファイル /etc/sma/snmp/snmpd.conf にエントリを追加して、アクセステーブルエントリを作成できます。

```
access grpnam1 fileX usm priv exact all none none
access grpnam1 "" usm auth exact all vwnam1 none
```

主要構成ファイル /etc/sma/snmp/snmpd.conf に追加することで作成されたグループの場合、 vacmAccessTable テーブル内に次のエントリが作成されます。

```
SNMP-VIEW-BASED-ACM-MIB::vacmAccessContextMatch.
"grpnam1"."".3.authNoPriv = INTEGER: exact(1)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessContextMatch.
"grpnam1"."fileX".3.authPriv = INTEGER: exact(1)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessReadViewName.
"grpnam1"."".3.authNoPriv = STRING: all
SNMP-VIEW-BASED-ACM-MIB::vacmAccessReadViewName.
"grpnam1"."fileX".3.authPriv = STRING: all
SNMP-VIEW-BASED-ACM-MIB::vacmAccessWriteViewName.
"grpnam1"."".3.authNoPriv = STRING: vwnam1
SNMP-VIEW-BASED-ACM-MIB::vacmAccessWriteViewName.
"grpnam1"."fileX".3.authPriv = STRING: none
SNMP-VIEW-BASED-ACM-MIB::vacmAccessNotifyViewName.
"grpnam1"."".3.authNoPriv = STRING: none
SNMP-VIEW-BASED-ACM-MIB::vacmAccessNotifyViewName.
"grpnam1"."fileX".3.authPriv = STRING: none
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStorageType.
"grpnam1"."".3.authNoPriv = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStorageType.
"grpnam1"."fileX".3.authPriv = INTEGER: permanent(4)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStatus.
"grpnam1"."".3.authNoPriv = INTEGER: active(1)
SNMP-VIEW-BASED-ACM-MIB::vacmAccessStatus.
"grpnam1"."fileX".3.authPriv = INTEGER: active(1)
```

主要構成ファイル /etc/sma/snmp/snmpd.conf を直接編集することで作成されたグループの場合、ストレージの型は permanent になります。

- snmpvacm コマンドを使ってアクセステーブルエントリを作成する方法もあります。

```
# snmpvacm -v3 -u myuser -a MD5
-A my_password -l authNoPriv localhost
createAccess grpnam1 "fileX" 3 3 1 all none none
```

ユーザー myuser のアクセスレベルは *rwuser* です。したがって、この例では、コンテキストに適していれば、アクセスエントリは myuser として作成されます。

例 4–4 一般的なアクセステーブルエントリの作成　(続き)

```
# snmpvacm -v3 -u myuser -a MD5
-A my_password -l authNoPriv localhost
createAccess grpnam1 "" 3 2 1 all vwnam1 none
```

snmpvacm コマンドで作成されたグループの場合、ストレージの型は nonvolatile になります。snmpvacm コマンドまたは snmpusm コマンドで作成されたオブジェクトの場合、ストレージの型は nonvolatile になります。

例 4–5 VACM テーブル内にユーザーが存在するかどうかのチェック

例 4–3 と 例 4–4 の情報を使って、例 4–2 で作成した SNMPv3 ユーザー user2 が存在することを確認します。user2 の値をチェックおよび設定して、このユーザー用に作成されたアクセスエントリを検証します。暗号化を使用して 1 回、暗号化を使用しないで 1 回、snmpget コマンドと snmpset コマンドを実行します。この方法により、user2 のアクセスエントリが最小限必要なセキュリティーレベル auth= 2 であることがわかります。この方法により、より安全なセキュリティーレベル priv の使用が可能であることもわかります。

snmpget コマンドを使って、暗号用の DES オプションが設定された新しいユーザーが存在することをチェックします。コンテキスト -n fileX が指定されています。

```
# snmpget -v3 -u user2 -a MD5
-A my_password -l authPriv -x DES -X my_password
-n fileX localhost 1.3.6.1.4.1.42.2.2.4.4.6.1.1.0
```

このコマンドは、user2 に設定したアクセスエントリのうち 1 つを検証します。snmpget コマンドの使用に関連付けられているオプションについては、snmpcmd(1M) のマニュアルページを参照してください。

snmpget コマンドで、次の情報が取得できます。

```
SNMPv2-SMI::enterprises.42.2.2.4.4.6.1.1.0 = INTEGER: 111
```

この出力の 111 は、指定の OID に格納された整数です。

同様に、snmpget コマンドを使って、DES オプションを設定せずに、新しいユーザーの存在をチェックできます。DES オプションを設定しない場合、暗号化は要求されません。この例では、ユーザー user2 はコンテキスト内にない操作を実行できます。

```
# snmpget -v3 -u user2 -a MD5
-A my_password -l authNoPriv localhost 1.3.6.1.2.1.1.3.0
```

snmpgetコマンドは、システム稼働時間に関する次の情報を取得します。

```
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (5375) 0:00:53.75
```

例 4–5 VACM テーブル内にユーザーが存在するかどうかのチェック　(続き)

sysLocation に新しい値を設定してみてください。

```
# snmpset -v3 -u user2 -a MD5
-A my_password -l authPriv -x DES -X my_password
localhost 1.3.6.1.2.1.1.6.0 s "new val"
```

このコマンドの s は文字列を表します。OID は sysLocation です。sysLocation に追加される値は new val です。

user2 は、DES のコンテキストに対する完全なアクセス権 (authPriv) を持っています。パスワードは *my_password* です。次の情報が返されます。

```
SNMPv2-MIB::sysLocation.0 = STRING: new val
```

snmpget コマンドを使って、これらの設定を確認します。

```
# snmpget -v3 -u user2 -a MD5
-A my_password -l authPriv -x DES -X my_password localhost 1.3.6.1.2.1.1.6.0

SNMPv2-MIB::sysLocation.0 = STRING: new val
```

同じコマンドを、DES 暗号化を設定しないで実行してみてください。

```
# snmpset -v3 -u user2 -a MD5
-A my_password -l authNoPriv localhost 1.3.6.1.2.1.1.6.0 s "new val2"
```

同じ結果が返されます。

```
SNMPv2-MIB::sysLocation.0 = STRING: new val2

# snmpget -v3 -u user2 -a MD5
-A my_password -l authNoPriv localhost 1.3.6.1.2.1.1.6.0

SNMPv2-MIB::sysLocation.0 = STRING: new val2
```

この出力によると、ユーザーは MIB-II への書き込みアクセスが許可されています。

user2 が snmptrapd.conf ファイルに定義されている場合は、snmptrapd コマンドを使って SNMP トラップデーモンを起動します。

```
# /usr/sfw/sbin/snmptrapd
```

また、snmpinform コマンドを使って INFORM-PDU トラップを送信します。snmpinform コマンドは、ユーザー user2 または user2 が通知を生成できることを確認します。コールド

例 4–5 VACM テーブル内にユーザーが存在するかどうかのチェック　(続き)

スタートを実行すると、通知が生成されます。コールドスタートは、通知 (トラップ) を生成します。このトラップは、/var/log/snmpd.log ファイルで確認できます。

```
# /usr/sfw/sbin/snmpinform  -v3 -u user2 -a MD5
 -A my_password -l authNoPriv localhost 42 coldStart.0
```

詳細については、snmptrapd.conf(4) および snmpinform(1M) のマニュアルページを参照してください。

VACM テーブルに関する問題の障害追跡

VACM テーブルエントリの作成時には、ユーザーとユーザーグループに正しいアクセス権を設定する必要があります。アクセス権の設定が正しくないと、主要ユーザーのアクセスが拒否される可能性があります。

ユーザーグループを大量に作成することは避けてください。グループの数が多いと、管理が難しくなります。ユーザーグループの数が多すぎると、問題の障害追跡が困難になります。

VACM テーブルの使用時、戻り値に次のメッセージが含まれることがあります。

noSuchContext	システム管理エージェントが、vacmContextTable 内で特定のメッセージの contextName を検出できない場合に返される値。アクセスは拒否される。コンテキストテーブルのエントリをチェックする必要がある。これらのエントリが正しく構成されているか確認する必要がある。各ユーザーにコンテキストを持っているかを確認する必要がある。詳細については、49 ページの「コンテキストテーブル」を参照
noSuchGroupName	msgSecurityModel 指示子および関連する SecurityName が vacmSecurityToGroupTable 内にない場合に返される値。アクセスは拒否される。グループセキュリティーテーブルのエントリをチェックする必要がある。これらのエントリが正しく構成されているか確認する必要がある。各ユーザーがグループ名を持っているかを確認する必要がある。ユーザーがテーブルに正しく入力されているかを確認する必要がある。詳細については、50 ページの「グループセキュリティーテーブル」を参照
notInView	この値は、MIB ビューに検索対象の OID が含まれていない場合に返される。アクセスは拒否される。詳細については、51 ページの「ビューツリーファミリテーブル」を参照
noAccessEntry	この値は、アクセス権が見つからない場合に返される。アクセスは拒否される。マスクが正しく設定されているか確認する必要がある。各グループは複数のアクセス権を持つことができるが、そのうちもっとも安全なアクセス権だけが選択される

vacmAccessContextMatch パラメータが *exact* に設定されているか確認する必要がある。*vacmAccessContextMatch* パラメータが exact に設定されている場合、`contextName` は完全に一致している必要がある。適切な場合は、*vacmAccessContextMatch* の値を *prefix* に設定してみるとよい。詳細については、55 ページの「アクセステーブル」を参照

注 – VACM テーブルが正しく構成されていないと、ネットワークが承認されていない不正なアクセスを受ける可能性があります。VACM 構成をテスト環境でテストしてから、ネットワークデバイスに実装してください。

VACM の詳細については、RFC 3415 (http://www.ietf.org/rfc/rfc3415.txt) を参照してください。

VACM の MIB 定義は、`/etc/sma/snmp/mibs/SNMP-VIEW-BASED-ACM-MIB.txt` にあります。

ユーザーの作成と管理

この節では、ユーザーを安全に作成する手順について説明します。システム管理エージェントでは、複数の方法でユーザーを作成できます。システム管理エージェントのインストール後、デフォルトの構成では、新しいユーザーは SNMPv1 および v2c ユーザーになります。

注 – デフォルトでは、エージェントは SNMPv3 ユーザーを作成するように設定されていません。システム管理エージェントで SNMPv3 ユーザーを作成するには、まず主要 ファイル `/etc/sma/snmp/snmpd.conf` を編集する必要があります。詳細については、`snmpd.conf(4)` のマニュアルページを参照してください。

この節の最初の手順、61 ページの「新しい SNMPv3 ユーザーを作成するには」では、最初の初期ユーザーを新規作成する方法を説明します。その他のユーザーは、この初期ユーザーを複製して作成します。ユーザーの認証やセキュリティーの型も、初期ユーザーから継承できます。これらの型はあとで変更可能です。複製を作成するときは、ユーザーの秘密鍵のデータを設定します。初期ユーザーと、あとで設定するユーザーのパスワードが必要になります。設定済みの初期ユーザーから、同時に複数の複製を作成することはできません。

▼ 新しい SNMPv3 ユーザーを作成するには

この手順で使用される `net-snmp-config` コマンドは、`/etc/sma/snmp/snmpd.conf` ファイルに 1 行を追加することにより、初期ユーザーに対し、エージェントへの読み取りおよび書き込みアクセスを許可します。

1 システム管理エージェントを停止します。

```
# svcadm disable -t svc:/application/management/sma:default
```

2 新規ユーザーを作成するには、net-snmp-config コマンドを実行します。

```
# /usr/sfw/bin/net-snmp-config --create-snmpv3-user -a "my_password" newuser
```

このコマンドは、newuser という新しいユーザーを作成します。パスワードは my_password と同じになります。新しいユーザーの作成には、MD5 とDES の両方を使用します。これらについては、43 ページの「認証プロトコルアルゴリズム」を参照してください。

デフォルトでは、これらの設定は、net-snmp-config コマンドを使ってユーザーを作成するとき、特に指定しなくても作成されます。

```
auth protocol = MD5 security level = rwuser auth
```

3 システム管理エージェントを起動します。

```
# svcadm enable svc:/application/management/sma:default
```

4 新しいユーザーが存在するかどうかを確認します。

```
# snmpget -v 3 -u newuser -l authNoPriv -a MD5 -A my_password localhost sysUpTime.0
```

注 – パスワードは 8 文字以上にする必要があります。

新しいユーザーに読み取りおよび書き込みアクセスを許可することが適切でない場合もあります。新しいユーザーに許可するアクセス権を減らすか変更する場合は、/etc/sma/snmp/snmpd.conf ファイルを編集します。詳細については、snmpd.conf(4) のマニュアルページを参照してください。

▼ システムプロンプトを使って新しいユーザーを作成するには

1 システム管理エージェントを停止します。

```
# svcadm disable -t svc:/application/management/sma:default
```

2 newuser という名前の新しいユーザーを作成し、my_password というパスワードを設定するには、net-snmp-config コマンドを対話的に使用します。

```
# /usr/sfw/bin/net-snmp-config --create-snmpv3-user

Enter a SNMPv3 user name to create:
```

3 適切なユーザー名を指定します。この例の場合、次のように指定します。

```
newuser

Enter authentication pass-phrase:
```

4 適切なパスワードを入力します。この例の場合、次のように入力します。

```
my_password

Enter encryption pass-phrase:
```

5 認証パスワードを再利用する場合は、**Return** キーを押します。

```
adding the following line to /var/sma_snmp/snmpd.conf:
createUser newuser MD5 "newuser_pass" DES
adding the following line to /etc/sma/snmp/snmpd.conf:
rwuser newuser
```

デフォルトでは、これらの設定は、`net-snmp-config` コマンドを使ってユーザーを作成するとき、特に指定しなくても作成されます。

```
auth protocol = MD5

security level = rwuser auth
```

6 システム管理エージェントを起動します。

```
# svcadm enable svc:/application/management/sma:default
```

7 新しいユーザーが存在するかどうかを確認します。

```
# snmpget -v 3 -u newuser -l authNoPriv -a MD5 -A my_password localhost sysUpTime.0
```

注 – パスワードは 8 文字以上にする必要があります。

新しいユーザーに読み取りおよび書き込みアクセスを許可することが適切でない場合もあります。新しいユーザーに許可するアクセス権を減らすか変更する場合は、`/etc/sma/snmp/snmpd.conf` ファイルを編集します。詳細については、`snmpd.conf(4)` のマニュアルページを参照してください。

▼ 追加の SNMPv3 ユーザーを安全に作成するには

安全な SNMP で新しいユーザーを作成する場合は、最初に設定した初期ユーザーを複製する方法をお勧めします。この方法では、61 ページの「新しい SNMPv3 ユーザーを作成するには」で設定したユーザーをコピーします。この方法では、42 ページの「USM による認証とメッセージプライバシ」で説明した `snmpusm` コマンドを使用します。詳細については、`snmpusm(1M)` のマニュアルページを参照してください。

1. システム管理エージェントが実行中かどうかをチェックします。

   ```
   # svcs svc:/application/management/sma:default
   ```

   エージェントがまだ起動していない場合は、起動します。

   ```
   # svcadm enable svc:/application/management/sma:default
   ```

2. snmpusm コマンドを使って新しいユーザーを作成します。

```
# snmpusm -v 3 -u newuser -a MD5 -A my_password -l authNoPriv localhost create lee newuser
```

   このコマンドでは、lee というユーザーを作成します。この新しいユーザーには、61 ページの「新しい SNMPv3 ユーザーを作成するには」で作成したソースユーザー newuser と同じパスワード my_password が与えられます。

3. 新しいユーザーのパスワードを変更します。

```
# snmpusm -v 3 -u lee -a MD5 -A my_password -l authNoPriv localhost passwd my_password lee_password
```

   このコマンドでは、ユーザー lee に新しいパスワード lee_password が与えられます。デフォルトの auth type は MD5 です。

4. /etc/sma/snmp/snmpd.conf ファイルを直接編集するか、snmpvacm コマンドを使って、関連する **VACM** エントリを作成します。

   snmpd.conf ファイルを直接編集する場合は、まずエージェントを一時的に停止する必要があります。

   ```
   # svcadm disable -t svc:/application/management/sma:default
   ```

5. lee にアクセスを割り当てます。

   - lee に読み取りおよび書き込みアクセスを許可するには、snmpd.conf ファイルに新しい rwuser 行を追加します。

     ```
     rwuser lee
     ```

   - lee に読み取り専用アクセスを許可するには、snmpd.conf ファイルに新しい rouser 行を追加します。

     ```
     rouser lee
     ```

   セキュリティーレベルを指定しなかった場合、デフォルトの authNoPriv が選択されます。詳細については、snmpd.conf(4) または snmpvacm(1M) のマニュアルページを参照してください。

6. システム管理エージェントを起動します。

   ```
   # svcadm enable svc:/application/management/sma:default
   ```

7 この手順が成功したかどうかを確認します。
新しいユーザーが存在するかどうかを確認します。

```
# snmget -v 3 -u lee -a MD5 -A lee_password -l authNoPriv localhost sysUpTime.0
```

SNMPv3 セキュリティーを使った SNMPv1 および SNMPv2c ユーザーの管理

SNMPv1 および v2c ユーザーのセキュリティー保護には、コミュニティー文字列が使用されます。SMA には、標準 Net-SNMP トークン `com2sec` が付属しています。`com2sec` トークンにより、SNMPv1 および v2c のホスト名とコミュニティー文字列のペアをセキュリティー名にマッピングできます。この場合、セキュリティーレベルは `noAuthNoPriv` になります。 `noAuthNoPriv` セキュリティーレベルとその他のセ キュリティレベルの詳細については、45 ページの「USM セキュリティー情報の格納場所」を参照してください。

プロキシ文とセキュリティー

システム管理エージェントでは、プロキシは SNMPv1 および v2c ユーザーに対してのみサポートされます。詳細については、69 ページの「Solstice Enterprise Agents 要求のプロキシ処理」を参照してください。

グループの作成と管理

SNMP 内で多数のグループを作成すると、これらのグループの管理が非常に複雑になります。多数のグループを作成すると、これらのグループの問題の障害追跡も非常に困難になります。

注 – `snmpd.conf` ファイルを編集して、グループやビューを作成した場合、ストレージの型は固定されます。`snmpvacm` コマンドを使用せず、`snmpd.conf` ファイルを編集した場合、グループのエントリが固定されます。エントリを削除するには、これらを `snmpd.conf` ファイルから削除する必要があります。

---

46 ページの「VACM によるアクセス制御」 のグループの作成および管理の例を参照してください。

◆ ◆ ◆ 第 5 章

その他のエージェントからの移行

この章では、その他の管理エージェントから システム管理エージェントへプロセスやタスクの処理を移行する方法について説明します。アプリケーションの移行については、『Solaris System Management Agent Developer's Guide 』を参照してください。Solaris OS 内で使用しているその他のエージェントから SMA へのアプリケーションの移行は、緊急ではありません。ただし、Solstice Enterprise Agents ソフトウェアの場合は例外です。

この章で扱うトピックは次のとおりです。

Solstice Enterprise Agents ソフトウェアからの移行

将来の Solaris リリースでは、Solstice Enterprise Agents ソフトウェアのサポートは打ち切られる予定です。Solstice Enterprise Agents ソフトウェアマスターエージェントは、`/usr/lib/snmp/` の `snmpdx` です。この機能は、システム管理エージェントマスターエージェント `snmpd` で置き換えられます。このエージェントは `/usr/sfw/sbin/` にあります。このため、開発者により作成されたすべての Solstice Enterprise Agents サブエージェントを、ある時点でシステム管理エージェントに移行する必要があります。

SMA が `seaProxy` モジュールを読み込むように設定されている場合、SMA と同時に Solstice Enterprise Agents ソフトウェアおよび関連するサブエージェントを実行できます。このモジュールのこの用途については、69 ページの「Solstice Enterprise Agents 要求のプロキシ処理」を参照してください。

Solstice Enterprise Agents ソフトウェアには、MIB-II と `sun.mib` を実装するサブエージェント、`mibiisa` が付属しています。システム管理エージェントでは、`mibiisa` の機能は、システム管理エージェントの MIB-II 部分により実装されています。

注 – 今回の Solaris リリースでは、Solstice Enterprise Agents `mibiisa` サブエージェントは無効になっています。`mibiisa` 宛のすべての SNMP 要求は、システム管理エージェント内の MIB-II 実装によって処理されます。

▼ ブート時にシステム管理エージェントが初期化されるのを防ぐには

デフォルトの設定では、Solaris ソフトウェアのブート時、SMA は `snmpdx` の起動後に起動します。エージェントを SMA へ移行しない場合は、SMA を停止し、起動スクリプトを編集します。この編集により、リブート時にシステム管理エージェントが自動的に起動することがなくなります。

1 `snmpd.conf` ファイルを開きます。

このファイルは、`/etc/sma/snmp/snmpd.conf` にあります。

2 `snmpd.conf` ファイルを編集します。

参照用として、ファイル内に手順が記載されています。

```
####################################
# SECTION: Admins who want to disable the snmpd daemon from
# starting at boot time.
# Change DISABLE=NO to DISABLE=YES
# DO NOT DELETE
# DO NOT UNCOMMENT
# DISABLE=NO
#
# end ADMIN
```

DISABLE フラグの NO を YES に変更して、`snmpd` デーモンがブート時に起動しないようにします。

```
####################################
# SECTION: Admins who want to disable the snmpd daemon from
# starting at boot time.
# Change DISABLE=NO to DISABLE=YES
# DO NOT DELETE
# DO NOT UNCOMMENT
# DISABLE=YES
#
# end ADMIN
```

Solstice Enterprise Agents 要求のプロキシ処理

SMA が seaProxy モジュールを読み込むように設定されている場合、Solstice Enterprise Agents ソフトウェアを使って開発した SNMP サブエージェントを SMA に移行する必要はありません。seaProxy モジュールを使用すると、Solstice Enterprise Agents ソフトウェアおよび関連するサブエージェントを SMA と同時に実行できるようになります。

注 – システム管理エージェントは、SMA からの着信要求を Solstice Enterprise Agents ソフトウェアにプロキシできるように特別にカスタマイズされています。この点が、システム管理エージェントと、SMA の ベースになっている標準 Net-SNMP 実装バージョン 5.0.9 との違いです。

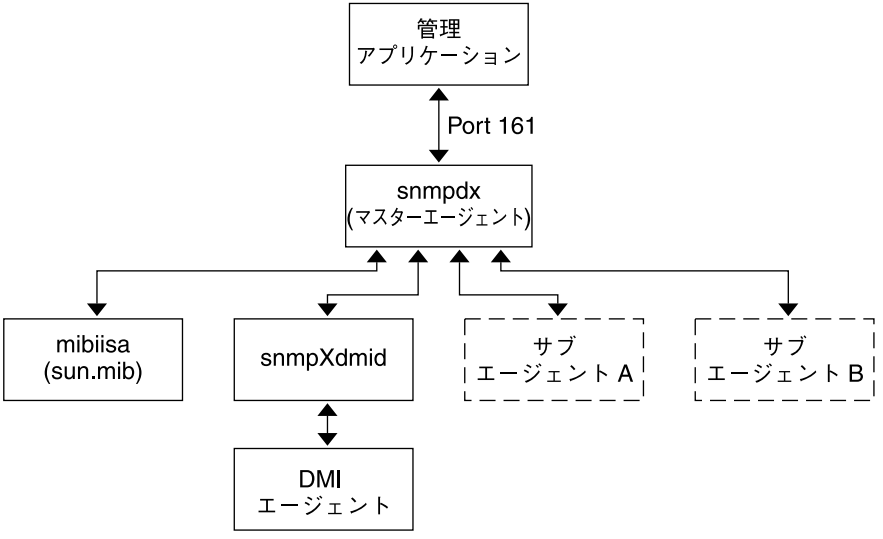

図 5–1 SEA ソフトウェアでの要求および応答のルーティング

システム管理エージェントのインストール後、もともと Solstice Enterprise Agents ソフトウェアのみで処理されていた要求の処理が変更されます。

- 以前は snmpdx に直接渡されていた MIB-II への要求が、システム管理エージェントマスターエージェントである snmpd によって処理される
- seaExtensions モジュールが読み込まれている場合、sun.mib (mibiisa Solstice Enterprise Agents サブエージェント MIB) への要求が、この seaExtensions モジュールで処理される。このモジュールが記述される MIB は SUN-SEA-EXTENSIONS-MIB
- seaProxy モジュールが読み込まれている場合、snmpdx.mib (Solstice Enterprise Agents ソフトウェアマスターエージェント MIB) への要求が、この seaProxy モジュールで処理される。このモジュールが記述される MIB は SUN-SEA-PROXY-MIB。 seaProxy モジュールの詳細については、70 ページの「seaProxy モジュールの有効化」を参照

ブート時のシステム管理エージェントの初期化を無効にしている場合、これらの記述は当てはまりません。詳細については、68 ページの「ブート時にシステム管理エージェントが初期化されるのを防ぐには」を参照してください。

proxy トークンを使用して、特定の OID の着信要求をすべて別のホストにプロキシすることができます。このプロキシ文については、snmpd.conf(4) のマニュアルページを参照してください。

seaProxy モジュールの有効化

Solstice Enterprise Agents ソフトウェア宛ての着信要求がポート 161 に着信すると、SMA によって受信されます。要求のプロキシが存在する場合、要求は snmpdx デーモンに渡されます。この要求は、snmpdx デーモンから Solstice Enterprise Agents ソフトウェアサブエージェントに渡されます。seaProxy モジュールは、snmpd.conf 内にない「動的プロキシ」を生成します。動的プロキシは、静的および動的な Solstice Enterprise Agents サブエージェント登録に基づいています。seaProxy モジュールは、Solstice Enterprise Agents サブエージェント登録の詳細情報を使って、動的プロキシを生成します。

x86 プラットフォーム上で実行中のシステムで、システム管理エージェント付属の seaProxy モジュールを有効にするには、/etc/sma/snmp/snmpd.conf ファイルに次の行が含まれていることを確認します。

```
dlmod seaProxy /usr/sfw/lib/libseaProxy.so
```

SPARC プラットフォーム上で実行中のシステムで、システム管理エージェント付属の seaProxy モジュールを有効にするには、/etc/sma/snmp/snmpd.conf ファイルに次の行が含まれていることを確認します。

```
dlmod seaProxy /usr/sfw/lib/sparcv9/libseaProxy.so
```

seaProxy モジュールの読み込み時、seaProxy モジュールはただちに Solstice Enterprise Agents サブエージェントからの情報収集を開始します。このため、特に snmpd デーモンは、snmpdx デーモンのあとに起動する必要があります。snmpd デーモンが snmpdx デーモンより先に起動する場合、SMA は Solstice Enterprise Agents ソフトウェアサブエージェント登録テーブルを再度読み込むことになります。snmpdx デーモンを snmpd デーモンより先に実行できるのは、たとえば、snmpd デーモンを停止して再起動した場合などです。

seaProxy モジュールは、ソフトウェアサブエージェント登録テーブル内の情報を使って、登録済みの Solstice Enterprise Agents ソフトウェアサブエージェントのプロキシを生成します。

seaProxy モジュールは、mibiisa サブエージェントのプロキシは生成しません。

着信要求のプロキシ文

この節では、システム管理エージェントから Solstice Enterprise Agents ソフトウェアに要求を渡すためのプロキシ文について説明します。

動的プロキシの生成後、システム管理エージェントプロキシ機構は、これらの要求を snmpdx へ転送する処理を行います。seaProxy モジュールは、snmpdx で登録する必要があるすべての Solstice Enterprise Agents サブエージェントに対して動的プロキシを生成します。SMA でも Solstice Enterprise Agents サブエージェントを使用できるのはこのためです。snmpdx を含む Solstice Enterprise Agents ソフトウェアのサポートは、切り替え期間を経て終了する予定です。Solstice Enterprise Agents で実装したサブエージェントを、なるべく早くシステム管理エージェントへ移行してください。

Solstice Enterprise Agents ソフトウェアからシステム管理エージェントへの移行は、AgentX サブエージェントを使って行われます。特定の Solstice Enterprise Agents モジュールをシステム管理エージェントへ移行する方法については、『Solaris System Management Agent Developer's Guide』を参照してください。このマニュアルには、モジュールの移行方法とシステム管理エージェントに付属しているデモモジュールの解説が記載されています。これらのデモモジュールの 1 つが、Solstice Enterprise Agents モジュールの移行プロセスを示しています。

システム管理エージェントと Solstice Enterprise Agents ソフトウェアを並行して実行する場合、SMA マスターエージェント snmpd はポート 161 を使用する必要があります。ブート時に、SMA サービスは匿名ポートを取得します。このサービスは、Solstice Enterprise Agents 構成ファイル /etc/snmp/conf/snmpd.conf 内のポートエントリを使って、このポート上で snmpdx が実行されるように設定します。変更後、/etc/snmp/conf/snmpdx.reg ファイルの最後の数行に、新しいポート番号が追加されます。

この例では、新しいポート番号は 16161 です。/etc/snmp/conf/snmpdx.reg ファイルの最後の数行には、その他の情報も含まれています。

```
agents =
{
  {  name = "relay-agent"
     subtrees = { sun.2.15 }
     timeout = 900000000
     port = 16161
  }
}
```

DMI サブエージェントなど、Solstice Enterprise Agents サブエージェントの起動時には、非公開コミュニティー文字列により、ポート 161 へ要求が送信されます。この非公開コミュニティー文字列は、起動時に読み取られるシステム管理エージェント構成ファイル内で定義する必要があります。そうしないと、Solstice Enterprise Agents サブエージェントが正しく登録されず、終了します。

SMA は、着信要求の OID 用にプロキシ文が生成されたかどうかをチェックします。SMA がこのチェックを行うのは、Solstice Enterprise Agents サブエージェントが要求内に格納している非公開コミュニティー文字列が、起動時に読み取られた SMA 構成ファイルに定義されている場合です。これらの文字列が検証されると、SMA は着信要求のポートを、この節で説明したように構成されたポートへ変更します。この例では、構成されたポートは 16161 です。

注 – seaProxy モジュールを有効にした場合は、SMA マスターエージェント snmpd の再起動後、Solstice Enterprise Agents ソフトウェアマスターエージェント snmpdx を再起動する必要はありません

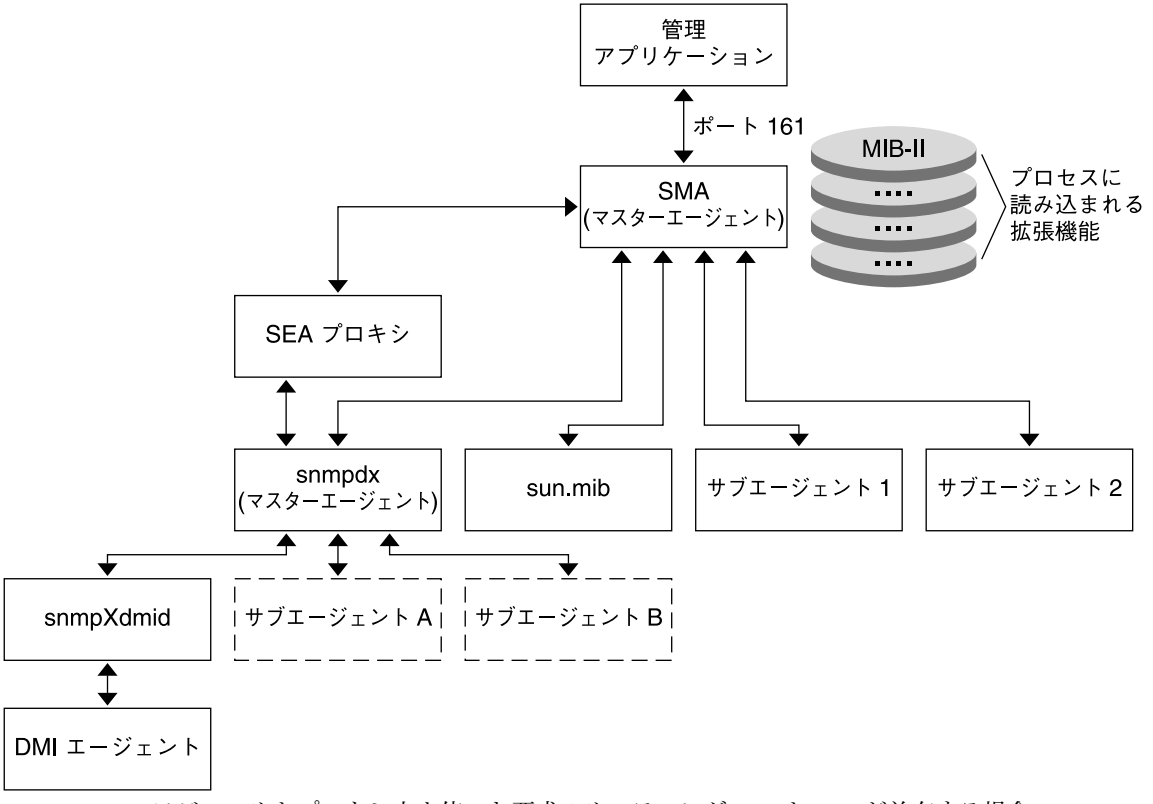

図 5–2 seaProxy モジュールとプロキシ文を使った要求のルーティング (SEA と SMA が並存する場合)

Sun Fire Management Agent からの移行

Sun SNMP Management Agent for Sun Fire™ and Netra™ Systems は、次のサーバーをサポートするスタンドアロン型 SNMP エージェントです。

- Sun Fire v210 サーバー
- Sun Fire v240 サーバー
- Sun Fire v250 サーバー
- Sun Fire v440 サーバー
- Netra 240 サーバー
- Netra 440 サーバー

Sun SNMP Management Agent for Sun Fire and Netra Systems は、ハードウェアインベントリおよび環境モニターに対する SNMP ベースのアクセスを提供します。詳細については、『SNMP Management Agent for the Sun Fire and Netra Systems』を参照してください。上記のいずれかのサーバーで Solaris 10 オペレーティングシステムを実行している場合、Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA へ移行する必要があります。

システム管理エージェントの場合と同様、Sun SNMP Management Agent for Sun Fire and Netra Systems は、`snmpd` という名前のデーモンも使用する SNMP エージェントです。これら 2 つの異なった `snmpd` デーモンが両方とも実行中の場合は、Sun SNMP Management Agent for Sun Fire and Netra Systems によって使用されている `snmpd` デーモンを停止するとき、停止しようとしているのが該当するデーモンであることを確認してください。システム管理エージェントの `snmpd` デーモンは、`/usr/sfw/sbin/snmpd` にあります。

Sun SNMP Management Agent for Sun Fire and Netra Systems からシステム管理エージェントへ移行する必要があります。ポート 161、ポート 162 では、Sun SNMP Management Agent for Sun Fire and Netra Systems を実行できません。

`masfcnv` 移行スクリプト

この節では、Sun SNMP Management Agent for Sun Fire and Netra Systems の構成を SMA に移行する手順について説明します。この手順では `masfcnv` スクリプトを使用します。このスクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA エージェントへの移行専用です。

./masfcnv 移行スクリプトは /usr/sfw/lib/sma_snmp にあります。./masfcnv 移行スクリプトは、次の機能を実行します。

- スクリプトは、USM (SNMP) ユーザー名とパスワードを移行する。スクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA に移行する USM ユーザー名が、SMA 内の既存のユーザー名と重複しないことを確認する。重複が存在する場合、そのユーザーを SMA のユーザーと同一ユーザーとして取り扱うかどうかを決定する必要がある。SMA 内の USM については、42 ページの「USM による認証とメッセージプライバシ」を参照

  そのユーザーに関連付けられている Sun SNMP Management Agent for Sun Fire and Netra Systems 鍵を移行するかどうかを決定する必要がある。移行しない場合は、そのユーザーの既存のシステム管理エージェント鍵を引き続き使用する。Sun SNMP Management Agent for Sun Fire and Netra Systems は MD5 ベースの鍵のみをサポートする。SMA は、SHA や SNMP 要求の暗号化 (DES) など、その他の認証スキーマもサポートする。したがって、移行後のユーザーは、必要な鍵が構成されるまで、これらの追加機能を使用できない。しかし、MD5 認証ベースのアクセスは許可される。認証と暗号化の詳細については、43 ページの「認証プロトコルアルゴリズム」を参照
- スクリプトは、/usr/sfw/lib/sma_snmp/ にある snmpd.conf テンプレートファイルを使用する。このテンプレートファイルを使って、新しい snmpd.conf エージェント構成ファイルが作成される。この新しい snmpd.conf エージェント構成ファイルは、Sun SNMP Management Agent for Sun Fire and Netra Systems 専用である。この新しい snmpd.conf エージェント構成ファイルは、/etc/opt/SUNWmasf/conf/ にインストールされる。Sun SNMP Management Agent for Sun Fire and Netra Systems は、この新しい snmpd.conf エージェント構成ファイルを使って、SMA の主要構成ファイルを変更する。Sun SNMP Management Agent for Sun Fire and Netra Systems はまた、エージェント構成ファイルを使って、/var/sma_snmp/snmpd.conf の SMA 持続的記憶領域ファイルを変更する

  SMA 構成ファイルの詳細については、28 ページの「構成ファイルと構成スクリプト」を参照
- このスクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems 構成ファイルをデフォルトの構成で置き換える。このデフォルト構成は、Sun SNMP Management Agent for Sun Fire and Netra Systems を AgentX サブエージェントとして設定する
- スクリプトは、変更された構成ファイルのバックアップを作成する。構成ファイルのバックアップを作成するには、ファイル名に拡張子 .bak.*n* を付ける (*n* はオプションの番号)
- このスクリプトは、/etc/init.d 内の既存の Sun SNMP Management Agent for Sun Fire and Netra Systems 起動スクリプトを新しいスクリプトで置き換える
- このスクリプトは、VACM 構成を移行する。SUN MIB によって使用されている OID 空間に関連した Sun SNMP Management Agent for Sun Fire and Netra Systems 構成は、自動的に移行される。VACM 構成はその他の OID に関連付けることができる。たとえば、VACM 情報を MIB-II のシステム分岐に関連付けることができる。VACM 情報がその他の OID に関連付けられている場合、移行が必要かどうかを確認する必要がある。VACM の詳細については、46 ページの「VACM によるアクセス制御」を参照

- このスクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA へトラップの宛先を移行する。もともと両方のエージェント用に構成されていたエンティティーが、移行後の構成で重複エントリになることはない
- このスクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA へコミュニティー文字列を移行する。両方のエージェントに同一の文字列が構成されている場合は、ユーザーに通知する

移行後、SMA は、標準ポート161/162 上でSNMP アクセスを提供します。その他のポートが構成されている場合、SMA はそのアクセスも提供します。SMA は、以前に Sun SNMP Management Agent for Sun Fire and Netra Systems が使用していたポートでの SNMP アクセスも提供します。すべてのポートは、同じ OID セットへのアクセスを提供します。これらの OID には、Sun SNMP Management Agent for Sun Fire and Netra Systems の場合と同様に、SUN-PLATFORM-MIB で使用される OID が含まれます。ユーザーごとにデータの可視性を制限するその他のアクセス制御を構成することもできます。

Sun SNMP Management Agent for Sun Fire and Netra Systems からユーザー名とパスワードを移行する場合、SMA で使用される `engineID` は、以前 Sun SNMP Management Agent for Sun Fire and Netra Systems で使用されていたものと同じでなければなりません。SNMPv3 で使用される USM は、`engineID` を認証に使用される鍵に埋め込みます。SMA が Sun SNMP Management Agent for Sun Fire and Netra Systems とは異なる `engineID` を使用するように設定した場合、使用する `engineID` を特定する必要があります。Sun SNMP Management Agent for Sun Fire and Netra Systems でもともと使用されていたのと異なる `engineID` を使用する場合、移行されたユーザーのパスワードをリセットします。USM の詳細については、42 ページの「USM による認証とメッセージプライバシ」を参照してください。

`masfcnv` スクリプトの詳細については、`masfcnv(1M)` のマニュアルページを参照してください。

注 – どのような場合でも、Sun SNMP Management Agent for Sun Fire and Netra Systems エージェントは Solstice Enterprise Agents の実行可能ファイル `snmpdx` とは別個に実行されます。Sun SNMP Management Agent for Sun Fire and Netra Systems エージェントを停止しても、Solstice Enterprise Agents ソフトウェアが自動的に停止することはありません。Solstice Enterprise Agents ソフトウェアからシステム管理エージェントに移行する必要があります。詳細については、67 ページの「Solstice Enterprise Agents ソフトウェアからの移行」を参照してください。

▼ Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA へ移行するには

1 スーパーユーザーで、システム管理エージェントと `masfd` エージェントを両方とも停止します。

```
# svcadm disable svc:/application/management/sma:default
# /etc/init.d/masfd stop
```

SMA のサブエージェントとして構成されているその他のエージェントも停止する必要があります。これらは、移行処理の完了後に再起動する必要があります。

2 テスト移行を実施し、移行スクリプトの実行結果を確認します。

テスト移行は、システム管理エージェントの構成に大幅な変更を加えた場合に便利です。

```
# cd /usr/sfw/lib/sma_snmp
# ./masfcnv --dry-run -i -p enable --select-community=agent
```

予行が正常に実行された場合は、指定の SMA 構成ファイルが標準出力へ書き出されます。処理を続行する前に、この出力の内容を確認してください。`./masfcnv` 移行スクリプトにより、Sun SNMP Management Agent for Sun Fire and Netra Systems の構成が SMA へ移行されます。構成の競合が発生した場合は、`masfcnv(1M)` のマニュアルページで解決方法を参照してください。

3 移行スクリプトを実行します。

```
# cd /usr/sfw/lib/sma_snmp
# ./masfcnv -i -p enable --select-community=agent
```

4 スーパーユーザーで、システム管理エージェントと **Sun SNMP Management Agent for Sun Fire and Netra Systems** の両方を再起動します。

```
# svcadm enable svc:/application/management/sma:default
# /etc/init.d/masfd start
```

Sun SNMP Management Agent for Sun Fire and Netra Systems が再構成され、システム管理エージェントのサブエージェントとして実行されるようになります。システム管理エージェントのサブエージェントとして構成されているその他のエージェントも、移行処理の完了後に再起動する必要があります。

Sun SNMP Management Agent for Sun Fire and Netra Systems から SMA への移行後、Sun Fire ハードウェア計測は、SMA 経由で SNMP アプリケーションにアクセスできるようになります。SMA は、以前に Sun SNMP Management Agent for Sun Fire and Netra Systems によって使用されていたポートを使用します。

◆ ◆ ◆ 付 録 A

ツールとマニュアルページ

この付録では、システム管理エージェントで使用可能な各種ツールおよびマニュアルページについて説明します。

ツールとユーティリティーの構成ファイル

システム管理エージェントでは、標準の Net-SNMP 実装の場合と同様、`snmp.conf` 構成ファイルを使って、使用可能なツールとユーティリティーの構成を行います。`snmp.conf` 構成ファイルは `/etc/sma/snmp/` に格納されています。

`snmp.conf` 構成ファイルを変更する前に、28 ページの「構成ファイルと構成スクリプト」を参照してください。また、`snmp_config(4)` および `snmp.conf(4)` のマニュアルページも、この順序で参照してください。

`snmp.conf` 構成ファイルは、`snmp.conf` のマニュアルページに記載されている指示子をサポートします。このファイルに重要な情報を格納する場合は、該当するユーザーだけがファイルを読み取れるようにアクセス権を設定してください。

マニュアルページ

この節では、システム管理エージェントに関連するすべてのマニュアルページを示します。これらのマニュアルページは、内容別の表に示します。

- 表 A–1
- 表 A–2
- 表 A–3
- 表 A–4
- 表 A–5

次の表には、一般的な SNMP 情報に関するマニュアルページを一覧します。

表 A–1 一般的な SNMP トピックに関するマニュアルページ

マニュアルページ	説明
netsnmp(5)	Solaris ソフトウェア付属の Net-SNMP 実装の概要を示す。sma_snmp(5) のマニュアルページからも参照できる
snmpcmd(1M)	Net-SNMP コマンドの共通オプションについて説明する
snmp_variables(4)	Net-SNMP コマンドに変数名を指定する際の所定の書式について説明する

次の表には、Net-SNMP コマンドツールに関するマニュアルページの一覧を示します。

表 A–2 SNMP ツールのマニュアルページ

マニュアルページ	ツールの説明
mib2c(1M)	mib2c ツールは、MIB 定義ファイル内のノードを使って、MIB モジュールの基盤となる 2 つの C コードテンプレートファイルを生成する
snmpbulkget(1M)	snmpbulkget ユーティリティーは、SNMP GETBULK 操作によりネットワークマネージャーに情報を送信する SNMP アプリケーションである
snmpbulkwalk(1M)	snmpbulkwalk ユーティリティーは、SNMP GETBULK 要求を通して、ネットワークエンティティーの情報ツリーを効率的に照会する SNMP アプリケーションである
snmpget(1M)	snmpget ユーティリティーは、SNMP GET 要求を通してネットワークエンティティーの情報を照会する SNMP アプリケーションである
snmpgetnext(1M)	snmpgetnext ユーティリティーは、SNMP GETNEXT 要求を通してネットワークエンティティーの情報を照会する SNMP アプリケーションである
snmpinform(1M)	snmpinform コマンドは、SNMP TRAP 操作によりネットワークマネージャーに情報を送信する SNMP アプリケーションである、snmptrap ユーティリティーを呼び出す
snmpnetstat(1M)	snmpnetstat コマンドは、SNMP プロトコルを使ってリモートシステムから取得した各種ネットワーク関連情報の値を記号で表示する
snmpset(1M)	snmpset ユーティリティーは、SNMP SET 要求を通してネットワークエンティティーの情報を設定する SNMP アプリケーションである

表 A–2 SNMP ツールのマニュアルページ　(続き)

マニュアルページ	ツールの説明
snmptrap(1M)	snmptrap ユーティリティーは、SNMP TRAP 操作によりネットワークマネージャーに情報を送信する SNMP アプリケーションである
snmpusm(1M)	snmpusm ユーティリティーは、SNMP エージェントの USM (ユーザーに基づくセキュリティーモデル) テーブルの簡単な保守に使用する SNMP アプリケーションである
snmpvacm(1M)	snmpvacm ユーティリティーは、SNMP エージェントの VACM (ビューに基づくアクセス制御モデル) テーブルの保守に使用する SNMP アプリケーションである
snmpwalk(1M)	snmpwalk ユーティリティーは、SNMP GETNEXT 要求を通してネットワークエンティティーの情報ツリーを照会する SNMP アプリケーションである
snmpdf(1M)	snmpdf コマンドは、df コマンドのネットワーク版である。snmpdf は、HOST-RESOURCES-MIB の hrStorageTable または UCD-SNMP-MIB の dskTable を調べて、リモートマシンのディスク容量をチェックする
snmpdelta(1M)	snmpdelta ユーティリティーは、指定された整数値の OID を監視する。このユーティリティーは、時間の経過による変化を報告する
snmptable(1M)	snmptable ユーティリティーは、SNMP GETNEXT または GETBULK 要求を繰り返し使用して、ネットワークエンティティーの情報を照会する SNMP アプリケーションである
snmptest(1M)	snmptest ユーティリティーは、ネットワークエンティティーの情報を監視および管理できる柔軟性の高い SNMP アプリケーションである。このユーティリティーは、コマンド行インタプリタを使って、さまざまな型の SNMP 要求をターゲットエージェントに送信できる
snmptranslate(1M)	snmptranslate ユーティリティーは、1 つ以上の SNMP オブジェクト識別子の値を記号やテキスト形式から数値形式へ変換するアプリケーションである。このアプリケーションはまた、1 つ以上の SNMP オブジェクト識別子の値を数値形式から記号やテキスト形式へも変換する
snmpstatus(1M)	snmpstatus コマンドは、ネットワークエンティティーから複数の重要な統計情報を取得する SNMP アプリケーションである

次の表には、Net-SNMP エージェントが使用する構成ファイルに関連するマニュアルページの一覧を示します。

表 A–3 SNMP 構成ファイルのマニュアルページ

マニュアルページ	説明
snmp_config(4)	システム管理エージェント付属の Net-SNMP 構成ファイルの概要を示す
snmp.conf(4)	snmp.conf ファイルは、snmpget や snmpwalk などの Net-SNMP アプリケーションの動作を定義する
snmpd.conf(4)	snmpd.conf ファイルは、Net-SNMP エージェントの動作を定義する
snmptrapd.conf(4)	snmptrapd.conf ファイルは、Net-SNMP トラップ受信デーモン snmptrapd がトラップを受信したときの動作を定義する
snmpconf(1M)	snmpconf ユーティリティーは、ユーザーに質問するスクリプトである。スクリプトは、ユーザーの応答に基づいて snmpd.conf 構成ファイルを作成する

次の表には、Net-SNMP に関連付けられたデーモンのマニュアルページの一覧を示します。

表 A–4 SNMP デーモンのマニュアルページ

マニュアルページ	説明
snmpd(1M)	snmpd デーモンは SNMP エージェントである。ポートにバインドされ、SNMP 管理ソフトウェアからの要求を待つ
snmptrapd(1M)	snmptrapd デーモンは、SNMP TRAP および INFORM メッセージを受信し、ログに記録する SNMP アプリケーションである

表 A–5 移行スクリプトのマニュアルページ

マニュアルページ	説明
masfcnv(1M)	masfcnv 移行スクリプトは、Sun SNMP Management Agent for Sun Fire and Netra Systems 用の既存の構成ファイルセットを SMA へ移行する場合に役立つ

用語集

Agent Extensibility Protocol (AgentX)
マスター SNMP エージェントと通信可能なサブエージェントプロトコル。

DAQ
データ収集。

DES
Data Encryption Standard (データ暗号化規格) の略。

MD5
RFC 1321 に定義されているメッセージダイジェスト関数。

MIB II
SNMP で管理可能なオブジェクトの仮想ファイルストアに関する現在の標準定義。

mib2c
MIB をコンパイルし、MIB 実装の構文テンプレートを生成するユーティリティー。

net-snmp
SMA の元になったオープンソース版。SMA は net-snmp の基本機能を使用し、SNMP プロトコルバージョン 1、2、および 3 をサポートする。

PDU
Protocol Data Unit (プロトコルデータユニット) の略。このユニットは、SNMP メッセージの型を定義する。PDU には、制御フィールドと配列が含まれる。配列要素は、それぞれペアになっている。制御フィールドはメッセージ型に依存する。配列内の各ペアの最初の要素は管理データを示す。配列内の各ペアの 2 番目の要素はこの管理データの値を示す。

SHA1
Secure Hash Algorithm - Version 1.0 の略。SHA は暗号化メッセージダイジェストアルゴリズムである。

USM
User-based Security Model (ユーザーに基づくセキュリティーモデル) の略。SNMP メッセージレベルセキュリティーを提供する標準。RFC 3414 () に説明がある。この RFC 文書には、USM の構成パラメータをリモートで監視および管理するための MIB についても記載されている。

VACM
View-Based Access Control Model (ビューに基づくアクセス制御モデル) の略。管理情報へのアクセスを制御するための標準。RFC 3415 () に説明がある。この RFC 文書には、VACM の構成パラメータをリモートで管理するための MIB についても記載されている。

エージェント　SNMP プロトコルを実装するソフトウェアプログラム。通常、管理対象デバイス上で実行される。マネージャーからの要求に対してサービスの提供も行う。SNMP で管理できない一部のネットワークノードのプロキシとして利用することもできる。

オブジェクト識別子 (OID)　すべての管理対象オブジェクト (デバイスまたはデバイスの特性) は、名前、構文、およびエンコーディングを持つ。この名前が、オブジェクトを一意に識別するオブジェクト識別子 (OID) です。OID は、整数をピリオドで区切った形式で表されます。たとえば、1.3.6.1.2.1.1.1.0 は、管理サブツリーのシステムグループ内のシステムの説明を指定します。

管理情報構造 (SMI)　業界で広く受け入れられている、論理アクセスを可能にするオブジェクト名の編成方法。SMI によると、管理対象オブジェクトはそれぞれ、名前、構文、およびエンコーディングを備えていなければならない。この名前が、オブジェクトを一意に識別するオブジェクト識別子 (OID) です。構文は、整数型、8 ビット文字列型などのデータ型を定義する。エンコーディングは、管理対象オブジェクトに関連付けられた情報をマシン間で伝送するために直列化する方法を示す。

管理情報ベース (MIB)　管理対象オブジェクトの仮想情報ストア。MIB は、デバイス内の管理対象オブジェクトのプロパティーを管理対象として定義する。

構成トークン　識別子、キーワード、定数、句読点、空白文字などのトークンがある。

コンテキスト　SNMP エンティティーからアクセス可能な管理対象オブジェクトの集合。管理対象オブジェクトのサブセットの名前。

サブエージェント　マスターエージェントと通信するエージェント。

システム管理エージェント (SMA)　オープンソースの Net-SNMP をベースにして、Sun で一部変更を加え、ツールとラッパーを追加した管理エージェント。

トラップ　管理対象デバイス上で発生した例外について記述されたマネージャー宛てのメッセージ。

プロキシエージェント　非 SNMP (外部) ネットワークデバイスの代わりに機能するエージェント。管理ステーションがプロキシエージェントに接続し、外部デバイスの識別情報を渡す。プロキシエージェントは、管理ステーションから受け取ったプロトコルインタラクションを外部デバイスがサポートする形式に変換する。

マスターエージェント　指定されたポート上で実行されるエージェント。

マネージャー　管理対象デバイスまたはシステムからデータにアクセスするクライアントアプリケーション。

レガシーサブエージェント　「プロキシエージェント」を参照。

索引

ろ