Solaris™ Security Toolkit 4.2
管理マニュアル

Sun Microsystems, Inc.
www.sun.com

Part No. 819-3789-10
2005 年 7 月, Revision A

コメントの送付: http://www.sun.com/hwdocs/feedback

原典: *Solaris Security Toolkit 4.2 Administration Guide*
Part No: 819-1402-10
Revision A

目次

図目次

表目次

コード例

はじめに

このマニュアルには、Solaris™ Security Toolkit ソフトウェアを理解し、使用するために必要な参考情報が記載されています。このマニュアルの主な対象読者は、Solaris™ Operating System (OS) バージョン 8、9、および 10 のセキュリティーを確保するために Solaris Security Toolkit ソフトウェアを使用するユーザーです。具体的には、管理者、コンサルタント、および Sun のシステムの新規配備または配備済みシステムのセキュリティー確保を担当するユーザーです。記載されている説明は、このソフトウェアを JumpStart™ モードまたはスタンドアロンモードのいずれかで使用する場合に適用されます。

お読みになる前に

このマニュアルの読者は、Solaris™ OS の Sun 認定システム管理者または Sun 認定ネットワーク管理者であることが必要です。また、標準ネットワークプロトコルおよびトポロジについて理解していることも必要です。

このマニュアルは、セキュリティーについてさまざまなレベルの知識や経験を持つユーザーに役立つように作成されているため、それぞれの知識や経験に合わせて適宜活用してください。

マニュアルの構成

このマニュアルはユーザーガイドです。各章には、このソフトウェアを使用してシステムのセキュリティーを確保する際の情報、操作説明、およびガイドラインが記載されています。このマニュアルは、以下の章で構成されています。

第 **1** 章では、Solaris Security Toolkit ソフトウェアの設計および目的について説明します。主要コンポーネント、機能、利点、サポートしているプラットフォームについて解説します。

第 **2** 章では、システムのセキュリティーを確保する方法について説明します。Solaris Security Toolkit を使用してシステムのセキュリティーを確保する場合は、あらかじめこのソフトウェアのプロセスを適用できます。

第 **3** 章では、Solaris Security Toolkit ソフトウェアおよびその他のセキュリティー関連ソフトウェアのダウンロード、インストール、および実行手順について説明します。

第 **4** 章では、セキュリティーの強化に伴い Solaris Security Toolkit ソフトウェアによって行われた変更を元に戻すための情報および手順について説明します。

第 **5** 章では、Solaris Security Toolkit ソフトウェアを使用するための JumpStart サーバーの構成および管理方法について説明します。

第 **6** 章では、Solaris Security Toolkit ソフトウェアを使用してシステムのセキュリティーを監査 (検証) する方法について説明します。セキュリティー強化後、確立されたセキュリティープロファイルを管理するには、この章で説明する情報と手順を使用します。

第 **7** 章では、前述の章に記載の情報と知識を利用して、実際に新規システムをインストールして、セキュリティーを確保する方法について説明します。

UNIX コマンド

このマニュアルには、システムの停止、システムの起動、およびデバイスの構成などに使用する基本的な UNIX® コマンドと操作手順に関する説明は含まれていない可能性があります。これらについては、以下を参照してください。

- 使用しているシステムに付属のソフトウェアマニュアル
- 下記にある Solaris™ オペレーティングシステムのマニュアル

  `http://docs.sun.com`

シェルプロンプトについて

シェル	プロンプト
UNIX の C シェル	*machine_name*%
UNIX の Bourne シェルと Korn シェル	$
スーパーユーザー (シェルの種類を問わない)	#

書体と記号について

書体または記号*	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。	.login ファイルを編集します。 ls -a を実行します。 % You have mail.
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表します。	**マシン名% su** Password:
AaBbCc123	コマンド行の可変部分。実際の名前や値と置き換えてください。	rm *filename* と入力します。
『 』	参照する書名を示します。	『Solaris ユーザーマニュアル』
「 」	参照する章、節、または、強調する語を示します。	第 6 章「データの管理」を参照。 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。	% **grep '^#define ** **XV_VERSION_STRING'**

* 使用しているブラウザにより、これらの設定と異なって表示される場合があります。

ハードウェアモデルに使用されている一般的な用語

Sun Fire™ ハイエンドシステムとは、次のモデル番号を指します。

- E25K
- E20K
- 15K
- 12K

Sun Fire ミッドレンジシステムとは、次のモデル番号を指します。

- E6900
- E4900
- 6800
- 4810
- 4800
- 3800

Sun Fire エントリレベルミッドレンジシステムとは、次のモデル番号を指します。

- E2900
- Netra 1280
- V1280
- V890
- V880
- V490
- V480

サポートされるハードウェアシステム

Solaris Security Toolkit 4.2 ソフトウェアは、Solaris 10 OS 上で稼働する SPARC® システム (64 ビットのみ) と x86/x64 システムをサポートします。Solaris Security Toolkit 4.2 ソフトウェアは、Solaris 8 および 9 で稼働する SPARC 32 ビットシステム (Ultra 2 Creator 3D など) はサポートしません。

サポートされる Solaris OS のバージョン

Sun では、Solaris Security Toolkit ソフトウェアを Solaris 8、Solaris 9、または Solaris 10 オペレーティングシステムで使用する場合にのみサポートを提供しています。

注 – Solaris Security Toolkit 4.2 ソフトウェアの場合、Solaris 10 を使用できるのは Sun Fire ハイエンドシステムのドメイン上だけであり、システムコントローラ (SC) 上では使用できません。

Solaris Security Toolkit ソフトウェアは Solaris 2.5.1、Solaris 2.6、および Solaris 7 オペレーティングシステムで使用することもできますが、これらのオペレーティングシステムで使用する場合、Sun ではサポートを提供していません。

Solaris Security Toolkit ソフトウェアは、インストールされている Solaris オペレーティングシステムのバージョンを自動的に検出し、そのバージョンに合わせて適切なタスクを実行します。

このマニュアル全体の例では、スクリプトが OS のバージョンをチェックする場合、スクリプトは、Solaris OS のバージョンである 2.*x*、7、8、9、または 10 ではなく、SunOS™ のバージョンである 5.*x* をチェックします。表 P-1 に、SunOS のバージョンと Solaris OS のバージョンの関係を示します。

表 P-1 SunOS のバージョンと Solaris OS のバージョンの相関関係

SunOS のバージョン	Solaris OS のバージョン
5.5.1	2.5.1
5.6	2.6
5.7	7
5.8	8
5.9	9
5.10	10

サポートされる SMS のバージョン

System Management Services (SMS) を使用して Sun Fire ハイエンドシステム上のシステムコントローラ (SC) を稼働させている場合は、すべての Solaris 8 および Solaris 9 OS バージョンで SMS バージョン 1.4、1.4.1、および 1.5 と Solaris Security Toolkit 4.2 ソフトウェアの併用がサポートされます。Solaris 10 OS 上で Solaris Security Toolkit 4.2 ソフトウェアによりサポートされる SMS のバージョンはありません。

注 – Solaris Security Toolkit 4.2 ソフトウェアの場合、Solaris 10 を使用できるのはドメイン上だけであり、システムコントローラ (SC) 上では使用できません。

関連マニュアル

オンラインのマニュアルは次の URL で参照できます。

```
http://www.sun.com/products-n-solutions/hardware/docs/
Software/enterprise_computing/systems_management/sst/index.html
```

用途	タイトル	Part No.	形式	場所
補正情報	Solaris Security Toolkit 4.2 ご使用にあたって	819-1504-10	PDF HTML	オンライン
リファレンス	Solaris Security Toolkit 4.2 リファレンスマニュアル	819-1503-10	PDF HTML	オンライン
マニュアルページ	Solaris Security Toolkit 4.2 マニュアルページガイド	819-1505-10	PDF	オンライン

マニュアル、サポート、およびトレーニング

Sun のサービス	URL	説明
マニュアル	http://jp.sun.com/documentation/	PDF と HTML マニュアルをダウンロードする、印刷マニュアルを注文する
サポートおよびトレーニング	http://jp.sun.com/supportraining/	テクニカルサポートを受ける、パッチをダウンロードする、Sun のコースについて情報を入手する

Sun 以外の Web サイト

このマニュアルで紹介する Sun 以外の Web サイトが使用可能かどうかについては、Sun は責任を負いません。このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、広告、製品、またはその他の資料についても、Sun は保証しておらず、法的責任を負いません。また、このようなサイトやリソース上、またはこれらを経由して利用できるコンテンツ、商品、サービスの使用や、それらへの依存に関連して発生した実際の損害や損失、またはその申し立てについても、Sun は一切の責任を負いません。

コメントをお寄せください

マニュアルの品質改善のため、お客様からのご意見およびご要望をお待ちしております。コメントは下記よりお送りください。

http://www.sun.com/hwdocs/feedback

ご意見をお寄せいただく際には、下記のタイトルと Part No. を記載してください。

『Solaris Security Toolkit 4.2 管理マニュアル』, part number 819-3789-10

第1章

概要

この章では、Solaris Security Toolkit ソフトウェアの設計と目的について説明します。主要コンポーネント、機能、利点、およびサポートしているプラットフォームについて解説します。また、変更時や配備時におけるバージョン管理のガイドラインとともに、Solaris Security Toolkit ソフトウェアをカスタマイズする際の重要なガイドラインも記載しています。

この章では、以下の項目を説明します。

Solaris Security Toolkit ソフトウェアによるシステムのセキュリティーの確保

Solaris Security Toolkit ソフトウェアは JumpStart Architecture and Security Scripts (JASS) ツールキットとも呼ばれ、Solaris OS システムのセキュリティーを確保して維持するための、拡張可能でスケーラブルな、自動化されたメカニズムを提供します。Solaris Security Toolkit ソフトウェアを使用して、システムのセキュリティーを強化し、また監査することができます。

次に、このガイドで使用されている重要な用語を示します。

- 強化 – Solaris OS の構成を変更してシステムのセキュリティーを向上させること。
- 監査 – システムの構成が事前に設定されたセキュリティープロファイルに従っているかどうかを調べること。

注 – 監査という用語は、システムのセキュリティー状態を定義済みセキュリティープロファイルと比較して検証する、Solaris Security Toolkit ソフトウェアの自動プロセスを指します。このマニュアルでこの用語を使用する場合、監査を実行したあとでシステムのセキュリティーが完全に確保されていることを保証するものではありません。

- スコアリング – 監査の実行時に発見された障害の数を数えること。どのような障害も検出されていない場合、この数は `0` になります。Solaris Security Toolkit は、障害が検出されるたびにこの数 (脆弱性値とも言う) を `1` ずつ増やします。

Solaris Security Toolkit ソフトウェアのインストールには、次の 2 つのモードがあります (これらの概要はこの節で後述)。

- 2 ページの「JumpStart モード」
- 3 ページの「スタンドアロンモード」

システムのインストール方法に関係なく、Solaris Security Toolkit ソフトウェアを使用して、システムのセキュリティーを強化および最小化できます。次に、Solaris Security Toolkit を定期的に使用することにより、セキュリティーを確保したシステムのセキュリティープロファイルが偶然あるいは故意に変更されていないかを監査します。

JumpStart モード

システムのインストールと構成は、可能な限り (理想的には完全に) 自動化するべきです。これには、OS のインストールと構成、ネットワークの構成、ユーザーアカウント、アプリケーション、セキュリティー強化などが含まれます。Solaris OS のインストールを自動化するために利用できるテクノロジの 1 つは JumpStart ソフトウェアです。JumpStart ソフトウェアを使用すると、ユーザー操作をほとんど (またはまったく) 行わずに、ネットワーク経由でシステムをインストールできます。Solaris Security Toolkit ソフトウェアには、JumpStart ソフトウェアベースのインストールにおいて、Solaris OS システムの強化に関連するタスクの大半を自動実行するフレームワークとスクリプトが用意されています。JumpStart ベースのインストールを容易にする JumpStart Enterprise Toolkit (JET) は、次の Sun ソフトウェアダウンロードサイトで入手できます。JET には、Solaris Security Toolkit によるセキュリティー強化をサポートするモジュールが含まれています。

```
http://www.sun.com/download/
```

JumpStart テクノロジについての詳細は、Sun BluePrints™ マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

スタンドアロンモード

また、Solaris Security Toolkit ソフトウェアにはスタンドアロンモードがあります。このモードでは、配備済みのシステムで JumpStart モードとまったく同じ強化機能を実行することができます。いずれのモードを使用する場合でも、セキュリティーの変更はシステムのセキュリティー要件に合わせてカスタマイズする必要があります。

システムのインストール方法に関係なく、Solaris Security Toolkit ソフトウェアを使用して、システムのセキュリティーを強化できます。次に、Solaris Security Toolkit を定期的に使用することにより、セキュリティーを確保したシステムの構成が偶然あるいは故意に変更されていないかを監査します。

ソフトウェアコンポーネントについて

この節では、Solaris Security Toolkit ソフトウェアコンポーネントの構造の概要を説明します。Solaris Security Toolkit ソフトウェアはファイルおよびディレクトリの集まりです。図 1-1 はソフトウェアコンポーネントの構造を示しています。

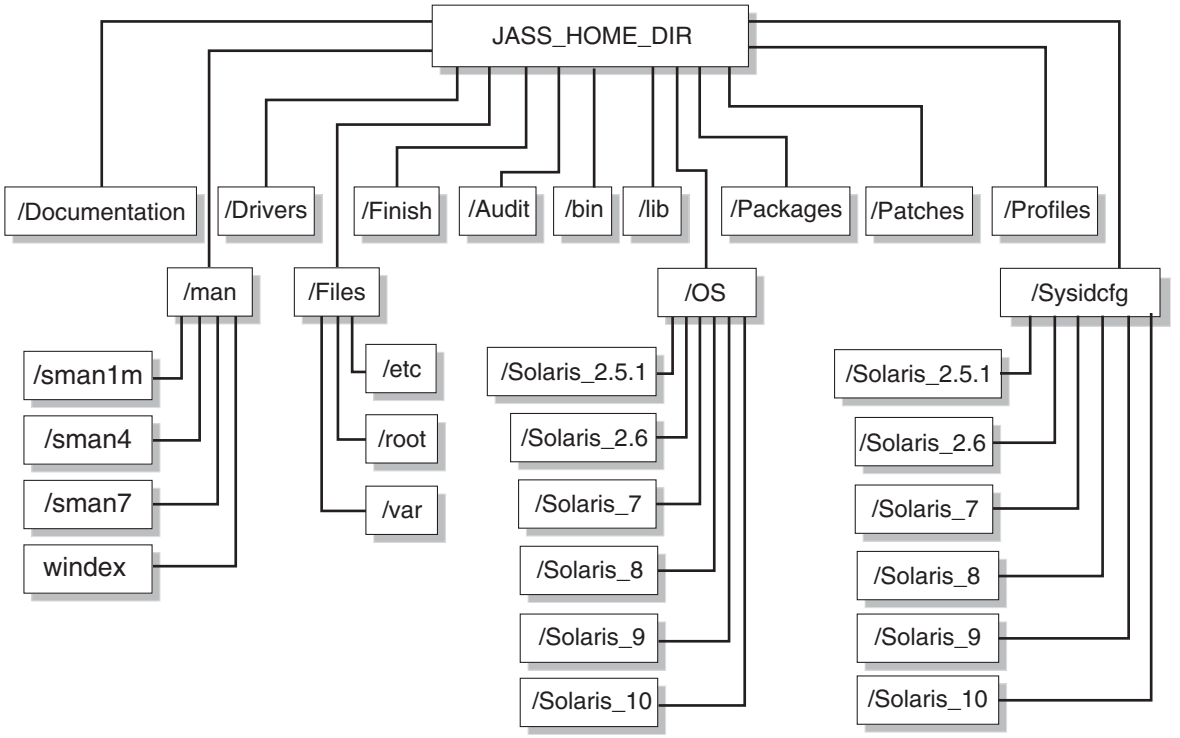

図 1-1 ソフトウェアコンポーネントの構造

次のプログラム (コマンドファイル) は、`/bin` ディレクトリにあります。

- `add-client` – JumpStart 環境にクライアントを追加するための JumpStart ヘルパープログラム
- `rm-client` – JumpStart 環境からクライアントを削除するための JumpStart ヘルパープログラム
- `make-jass-pkg` – カスタマイズした Solaris Security Toolkit 構成の内部配布を簡単にするため、Solaris Security Toolkit ディレクトリの内容から Solaris OS パッケージを作成するコマンド
- `jass-check-sum` – Solaris Security Toolkit の実行中に作成されたチェックサムに基づいて、Solaris Security Toolkit ソフトウェアにより修正済みのファイルが変更されていないかどうかを確認するためのコマンド
- `jass-execute` – Solaris Security Toolkit ソフトウェアのほとんどの機能を実行するコマンド

ディレクトリ

Solaris Security Toolkit アーキテクチャーのコンポーネントは以下のディレクトリで構成されています。

- /Audit
- /bin
- /Documentation
- /Drivers
- /Files
- /Finish
- /lib
- /man
- /OS
- /Packages
- /Patches
- /Profiles
- /Sysidcfg

各ディレクトリについて、この節で説明します。関連がある場合は、各スクリプト、構成ファイル、またはサブディレクトリをリストするとともに、詳細情報を記載した参照先の章も示しています。

Solaris Security Toolkit ディレクトリ構造は Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

Audit ディレクトリ

このディレクトリには、システムが定義済みのセキュリティープロファイルに適合しているかどうかを評価する監査スクリプト、または監査スクリプトのセットが格納されています。このディレクトリ内のスクリプトは以下のカテゴリに分かれています。

- 無効化 (disable)
- 有効化 (enable)
- インストール (install)
- 最小化 (minimize)
- 印刷 (print)
- 削除 (remove)
- 設定 (set)
- 更新 (update)

各カテゴリに含まれるスクリプトの詳細なリスト、および各スクリプトの説明については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

Documentation ディレクトリ

このディレクトリには、README、EOL_NOTICE、INSTALL ファイルなど、ユーザー向け情報が記載されたテキストファイルが格納されています。

man ディレクトリ

このディレクトリには、コマンド、関数、およびドライバに関するマニュアルページのセクションがサブディレクトリとして格納されています。また、このディレクトリには、コマンドの索引である windex ファイルもユーザーの便宜を考えて格納されています。

これらのマニュアルページについての詳細は、実際のマニュアルページまたは『Solaris Security Toolkit 4.2 マニュアルページガイド』を参照してください。

Drivers ディレクトリ

このディレクトリには、Solaris Security Toolkit ソフトウェアの実行時に実行およびインストールされるファイルを指定した構成情報ファイルが格納されています。具体的には、ドライバ、スクリプト、および構成ファイルが含まれています。

以下は Drivers ディレクトリに格納されているドライバおよびスクリプトの例です。

- audit_{private|public}.funcs
- common_{log|misc}.funcs
- {config|hardening|secure}.driver
- driver.{init|run}
- driver_{private|public}.funcs
- finish.init
- server-{config|hardening|secure}.driver
- suncluster3x-{config|hardening|secure}.driver
- sunfire_15k_sc-{config|hardening|secure}.driver
- undo.{funcs|init|run}
- user.init.SAMPLE
- user.run.SAMPLE

Solaris Security Toolkit に含まれるドライバにはすべて、次に示す 3 つのファイルが存在します。

- *name*-{config|hardening|secure}.driver

これらの 3 つのファイルは、上記のリストでは sunfire_15k_sc-{config|hardening|secure}.driver のように括弧内に示されています。これらのファイルは詳細説明用としてリストしたものです。ドライバを実行するときは、secure.driver または *name*-secure.driver だけを使用してください。このドライバによって、関連するドライバが自動的に呼び出されます。

Solaris Security Toolkit アーキテクチャーには構成情報が含まれているため、実際のスクリプト自体は変更せずに、異なる環境でドライバ、終了スクリプト、および監査スクリプトを使用することができます。終了スクリプトおよび監査スクリプトで使用される変数はすべて構成ファイルで維持されています。これらの構成ファイルはドライバによってインポートされ、ドライバから呼び出されたときに、終了スクリプトと監査スクリプトで変数を使用できるようになります。

Solaris Security Toolkit ソフトウェアにはメインの構成ファイルが 4 つあり、すべて `Drivers` ディレクトリに格納されています。

- `driver.init`
- `finish.init`
- `user.init`
- `user.run`

`user.run` ファイルには、後継バージョンまたは機能拡張されたバージョンの Solaris Security Toolkit 関数の配置場所を記すことができます。これらの関数が存在する場合、自動的に使用されます。

注意 – 変数定義の変更は、`user.init` 構成ファイル内だけで行なってください。`driver.init` 構成ファイルおよび `finish.init` 構成ファイル内の定義は決して変更しないでください。

ドライバによって呼び出される終了スクリプトは `Finish` ディレクトリにあり、監査スクリプトは `Audit` ディレクトリにあります。ドライバでインストールされるファイルは `Files` ディレクトリから読み込まれます。終了スクリプトの詳細については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』の第 4 章を参照してください。監査スクリプトの詳細については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』の第 5 章を参照してください。

図 1-2 はドライバの制御フローを示しています。

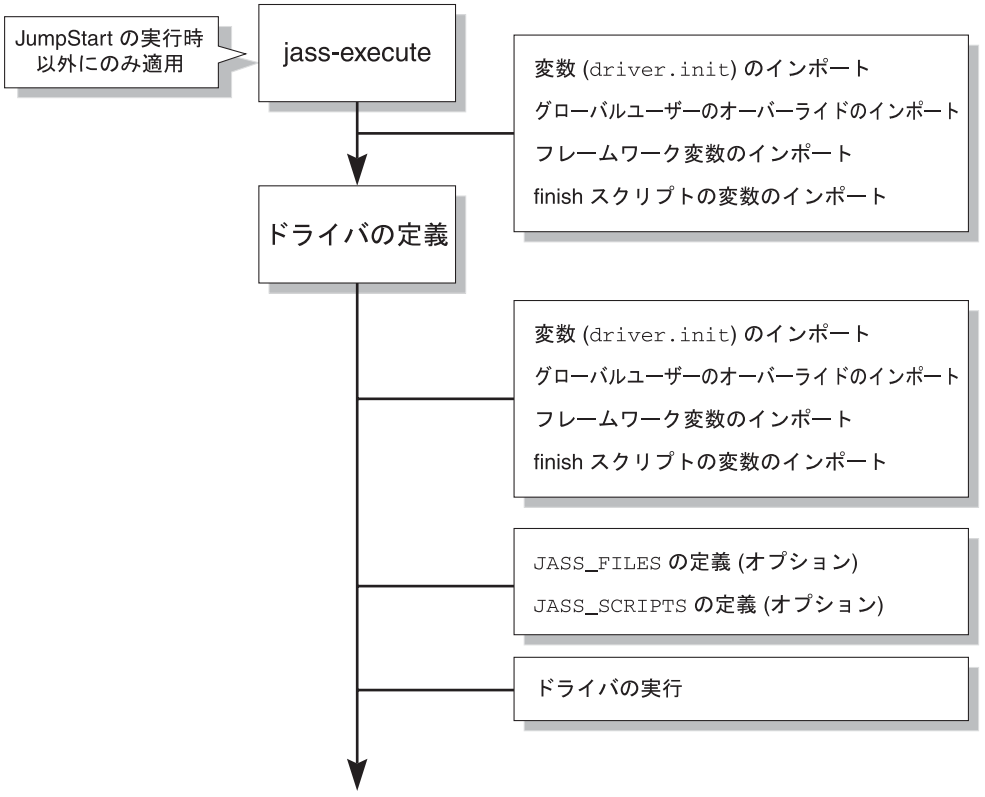

図 1-2 ドライバの制御フロー

1. ドライバによる `jass-execute` コマンドの使用は、JumpStart を使用しない実行にだけ適用されます。JumpStart の実行は、`jass-execute` コマンドを呼び出さず、ドライバを直接呼び出します。
2. ドライバによって、変数を明示的に設定できます。
3. ドライバは、各種の `.init` ファイルから環境変数をすべてインポートします。
4. ドライバは、`JASS_FILES` 環境変数と `JASS_SCRIPTS` 環境変数を定義します。これらの定義はオプションであり、いずれか 1 つ、または両方を定義することができます。いずれも定義しないことも可能です。

   環境変数 `JASS_FILES` と `JASS_SCRIPTS` の定義についての詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』の第 7 章を参照してください。
5. ドライバは `driver.run` を呼び出し、`JASS_FILE` および `JASS_SCRIPTS` 環境変数で定義されているタスクを実行します。

6. (オプション) ドライバは、finish.init または user.init 内のシステムデフォルトを上書きするために使用できる特定のドライバ動作を定義します。コード例 1-1 に示すドライバは、JASS_PASS_HISTORY を明示的に 4 に設定しています。

コード例 1-1 は、ドライバの制御フローコードを示しています。

コード例 1-1 ドライバの制御フローコード

```
DIR="`/bin/dirname $0`"
JASS_PASS_HISTORY="4"
export DIR
. ${DIR}/driver.init

JASS_FILES="
                          /etc/cron.d/cron.allow
                          /etc/default/ftpd
                          /etc/default/telnetd
"

JASS_SCRIPTS="
                          install-at-allow.fin
                          remove-unneeded-accounts.fin
"
. ${DIR}/driver.run
```

1. このコード例では、ドライバが開始ディレクトリを認識できるように、DIR 環境変数が設定およびエクスポートされます。
2. このドライバは、JASS_PASS_HISTORY 環境変数を明示的に 4 に設定します。
3. このドライバは、driver.init から読み取りを開始して、各種の .init ファイルを読み取ります。
4. JASS_HOME_DIR/Files ディレクトリからクライアントにコピーされるファイルを含むように、JASS_FILES 環境変数が定義されます。
5. Solaris Security Toolkit ソフトウェアによって実行される終了スクリプトで JASS_SCRIPTS 環境変数が定義されます。
6. driver.run ドライバを呼び出して、強化タスクが実行されます。driver.run は、JASS_FILES で定義されているファイルをコピーし、JASS_SCRIPTS で指定されているスクリプトを実行します。

Files ディレクトリ

このディレクトリは、JumpStart クライアントにコピーされるファイルを格納する目的で、JASS_FILES 環境変数と driver.run スクリプトによって使用されます。

このディレクトリには以下のファイルがあります。

- /.cshrc
- /.profile
- /etc/default/sendmail
- /etc/dt/config/Xaccess
- /ftpd/banner.msg
- /etc/hosts.allow
- /etc/hosts.allow-15k_sc
- /etc/hosts.allow-server
- /etc/hosts.allow-suncluster
- /etc/hosts.deny
- /etc/init.d/klmmod
- /etc/init.d/nddconfig
- /etc/init.d/set-tmp-permissions
- /etc/init.d/sms_arpconfig
- /etc/init.d/swapadd
- /etc/issue
- /etc/motd
- /etc/opt/ipf/ipf.conf
- /etc/opt/ipf/ipf.conf-15k_sc
- /etc/opt/ipf/ipf.conf-server
- /etc/security/audit_class+5.10
- /etc/security/audit_class+5.8
- /etc/security/audit_class+5.9
- /etc/security/audit_control
- /etc/security/audit_event+5.10
- /etc/security/audit_event+5.8
- /etc/security/audit_event+5.9
- /etc/sms_domain_arp
- /etc/sms_sc_arp
- /etc/syslog.conf
- /root/.cshrc
- /root/.profile
- /var/opt/SUNWjass/BART/rules
- /var/opt/SUNWjass/BART/rules-secure

Finish ディレクトリ

このディレクトリには、実行中にシステムの変更および更新を行う終了スクリプトが格納されています。このディレクトリ内のスクリプトは以下のカテゴリに分かれています。

- 無効化 (disable)
- 有効化 (enable)
- インストール (install)
- 最小化 (minimize)
- 印刷 (print)

- 削除 (remove)
- 設定 (set)
- 更新 (update)

各カテゴリに含まれるスクリプトの詳細なリスト、および各スクリプトの説明については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

OS ディレクトリ

このディレクトリには、Solaris OS イメージのみが格納されています。これらのイメージは、JumpStart ソフトウェアのインストールプロセスでクライアントインストールの Solaris OS ソースとして使用されます。このディレクトリの名前が次の Solaris Secutiry Toolkit OS 命名規則に従っている場合、`add_client` スクリプトは、このディレクトリに含まれる Solaris OS バージョンを引数として受け入れます。

Solaris OS イメージの読み込みと変更についての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

標準のインストール命名規則は次のとおりです。

Solaris OS

Solaris OS では次の命名規則を使用します。

`Solaris_`*os version_4 digit year–2 digit month of CD release*

たとえば、2005 年 3 月リリースの Solaris 10 Operating Environment CD の場合、ディレクトリ名は `Solaris_10_2005-03` になります。Solaris OS のアップデートとリリースを区別することにより、テストおよび配備に関する情報を厳密に管理することができます。

Solaris OS (x86/x64 プラットフォーム版)

Solaris OS (x86/x64 プラットフォーム版) では、次のディレクトリ命名規則を使用します。

`Solaris_`*os version_4 digit year–2 digit month of CD release*`_ia`

たとえば、Solaris OS (x86/x64 プラットフォーム版) のリリース日が 2005 年 3 月の場合、ディレクトリ名は `Solaris_10_2005-03_ia` のようになります。

Packages ディレクトリ

このディレクトリには、終了スクリプトによるインストールと監査スクリプトによる検証が可能なソフトウェアパッケージが格納されています。たとえば、Open Secure Shell ソフトウェアパッケージは、Packages ディレクトリに格納できます。これにより、必要に応じて正しい終了スクリプトを使用してこのソフトウェアをインストールできるようになります。

Solaris Security Toolkit ソフトウェアに含まれるいくつかの終了スクリプトと監査スクリプトは、ソフトウェアのインストール、基本的な構成、および検証の各機能を実行します。次に、Packages ディレクトリからソフトウェアをインストールして検証するスクリプトを示します。

- install-fix-modes.{fin|aud}
- install-jass.{fin|aud}
- install-md5.{fin|aud}
- install-openssh.{fin|aud}

Patches ディレクトリ

このディレクトリは、Solaris OS に対応した推奨およびセキュリティパッチクラスタを格納するために使用します。必要なパッチをダウンロードして、このディレクトリに圧縮解除します。

パッチを保存してこのディレクトリに圧縮解除することにより、インストールが効率化されます。パッチをこのディレクトリに圧縮解除すると、Solaris Security Toolkit ソフトウェアのパッチインストールスクリプトによってインストールが自動的に行われます。ユーザーがインストールごとに手動でパッチクラスタを圧縮解除する必要はありません。

使用している Solaris OS バージョンごとにサブディレクトリを作成します。たとえば、Patches ディレクトリ内に 9_Recommended ディレクトリと 10_Recommended ディレクトリが存在する場合があります。

Solaris Security Toolkit ソフトウェアは、Solaris OS (x86/x64 プラットフォーム版) パッチクラスタをサポートしています。このパッチクラスタで使用される命名規則は、SunSolve OnLine℠ サービスで利用できるものと同じです。

形式は <release>_x86_Recommended です。Solaris 10 OS の Solaris OS (x86/x64 プラットフォーム版) パッチクラスタの場合、ディレクトリ名は 10_x86_Recommended になります。

Profiles ディレクトリ

このディレクトリには、すべての JumpStart プロファイルが格納されています。プロファイルには、インストールする Solaris OS クラスタ (Core、End User、Developer、または Entire Distribution)、ディスクのレイアウト、およびインストールの種類 (スタンドアロンなど) を決定するために JumpStart ソフトウェアによって使用される構成情報が含まれます。

JumpStart プロファイルは rules ファイルでリストおよび使用され、特定のシステムまたはシステムグループの構築方法を定義します。

Sysidcfg ディレクトリ

Profiles ディレクトリと同様に、Sysidcfg ディレクトリに格納されているファイルは JumpStart モードのインストールでのみ使用されます。これらのファイルは、必要なインストール情報を提供することで Solaris OS のインストールを自動化します。各 OS 固有の情報が別々のディレクトリツリーで保存されています。

Solaris OS ごとに別々のディレクトリがあります。リリースごとに、Solaris_*OS Version* という名前のディレクトリが存在します。Solaris Security Toolkit ソフトウェアには、Solaris OS バージョン 2.5.1 ～ 10 に対応した sysidcfg サンプルファイルが含まれています。

sysidcfg サンプルファイルは、ネットワークごと、ホストごとなど、ほかの種類に拡張できます。Solaris Security Toolkit ソフトウェアは、任意の sysidcfg ファイルをサポートします。

sysidcfg ファイルについての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

データリポジトリ

データリポジトリ (JASS_REPOSITORY) ディレクトリは JASS_HOME_DIR ディレクトリ構造内には存在しませんが、Solaris Security Toolkit の操作を元に戻す機能の実行をサポートしており、各実行内容に関するデータを保存します。また、このソフトウェアによって変更されたファイルの一覧を維持し、実行ログに表示されるデータを保存します。このディレクトリは、/var/opt/SUNWjass/runs/*timestamp* 内にあります。

バージョンの管理

Solaris Security Toolkit ソフトウェアで使用されるすべてのファイルおよびスクリプトのバージョンを管理することは、2 つの理由で非常に重要です。

1. この環境の目的の 1 つは、システムのインストールを再現することにあります。インストール中に使用されたすべてのファイルバージョンのスナップショットがなければ、この目的は達成できません。
2. これらのスクリプトは多くの組織にとってきわめて重要なセキュリティー機能を実行するため、テストが完了している必要な変更だけが実装されるよう細心の注意を払う必要があります。

Source Code Control System (SCCS) バージョン管理パッケージが Solaris OS `SUNWsprot` パッケージで提供されています。フリーウェア、または市販の他のバージョン管理ソフトウェアを使用してバージョン情報を管理することもできます。いずれのバージョン管理製品を使用する場合でも、将来のシステムの再構築に備えて、アップデート情報を管理するとともに、バージョン情報の取得プロセスを実装しておくことが必要です。

ファイルの内容が変更されているかどうかを判断するには、バージョン管理に加えて、完全性管理ソリューションを使用します。システムに対して特権を持つユーザーであれば、バージョン管理システムを回避することは可能ですが、遠隔システム上で完全性データベースを運用する完全性管理システムは簡単に回避できません。ローカルシステムに格納されているデータベースは不正に改ざんされる恐れがあるため、完全性管理ソリューションは一元管理するのが最良の方法です。

Solaris Security Toolkit ソフトウェアの構成およびカスタマイズ

Solaris Security Toolkit ソフトウェアには、Sun BluePrints マニュアル『Enterprise Security: Solaris Operating Environment Security Journal, Solaris Operating Environment Versions 2.5.1, 2.6, 7, and 8』および Sun BluePrints OnLine のセキュリティー関連記事に記載されているすべてのセキュリティーガイドラインを実行するためのスクリプト、フレームワーク関数、および変数のデフォルト値が組み込まれています。これらの値はすべてのシステムに適合するわけではありません。使用しているシステムのセキュリティー要件に適合するように、Solaris Security Toolkit ソフトウェアをカスタマイズする必要があります。

Solaris Security Toolkit ソフトウェアの最も優れた特性の 1 つは、環境、システム、およびセキュリティー要件に合わせて簡単にカスタマイズできることです。Solaris Security Toolkit ソフトウェアをカスタマイズするには、ドライバ、終了スクリプト、監査スクリプト、フレームワーク関数、環境変数、およびファイルテンプレートを通して動作を調整します。

ほとんどの場合、Solaris Security Toolkit コードを変更する必要はありません。Solaris Security Toolkit ソフトウェアを環境に合わせて使用する際に、コードを変更せざるを得ない場合は、`user.run` 内でコードを一意の関数名にコピーし、15 ページの「ガイドライン」で記述されているとおり変更履歴を簡単に追跡できるようにします。

このマニュアルでは、Solaris Security Toolkit ソフトウェアのカスタマイズに関するガイドラインと手順を説明しています。ドライバのカスタマイズに役立つ情報を取得するには、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。カスタマイズでは、ファイルや変数を変更および作成します。

このマニュアルでは、Solaris Security Toolkit ソフトウェアのカスタマイズ例も記載しています。例ではいくつかのカスタマイズ方法を取り上げていますが、他にも多くの方法があります。

Solaris Security Toolkit ソフトウェアをカスタマイズする前に明確に理解しておくべき情報について、以下の節で説明します。情報は多数の配備例から収集された共通のユーザー経験に基づくものであり、一般的な間違いを防ぐために役立ちます。

ポリシーおよび条件

Solaris Security Toolkit ソフトウェアをカスタマイズおよび配備する場合、適切な計画を立てることにより、カスタマイズ後のプラットフォーム構成が適正なものとなり、組織の期待に添うものになります。

計画を立てる段階で、セキュリティーのポリシーと基準、業界の規制とガイドライン、ベンダーの推奨する実践方法など、さまざまな情報を入手します。

こうした情報に加えて、カスタマイズ後の構成によって、プラットフォームに備わっている所定のビジネス機能に影響が出ないように、アプリケーションと動作条件を検討する必要があります。

ガイドライン

Solaris Security Toolkit ソフトウェアをカスタマイズする際は、以下のガイドラインを考慮してください。これらのガイドラインを理解し遵守することによって、配備のプロセスが大幅に簡素化され、効果的なものになります。

- 一般的な規則として、Solaris Security Toolkit ソフトウェアに含まれている元のファイル (ドライバ、スクリプト、ファイルなど) を変更しないでください。元のファイルの変更内容が新しいバージョンのファイルによって上書きされることがあるため、元のファイルを変更すると、Solaris Security Toolkit ソフトウェアを新しいバージョンにアップグレードできなくなります。カスタマイズの内容はすべて失われ、システムの構成が不適切に変更される可能性があります。

  ファイルをカスタマイズする場合は、まずファイルをコピーします。元のファイルはそのままにして、コピーを変更します。このガイドラインには 1 つだけ次の例外があります。

  - `sysidcfg` ファイル
- Solaris Security Toolkit 4.2 ソフトウェアには、`Files` ディレクトリ内のテンプレートにキーワード接尾辞を使用するという新しい機能があります。この方法を利用すると、Solaris Security Toolkit 4.2 ソフトウェアに付属している各デフォルトテンプレートを変更する必要がなくなります。可能なかぎり接尾辞を使用してください。
- ドライバまたはスクリプトのコピーに、元のファイルと区別できるような名前を付けてください。スクリプトの目的を表すプレフィックスやキーワードを使用します。会社の名前あるいは株式記号などのプレフィックス、部署を識別するもの、あるいはプラットフォームやアプリケーションの種類なども名前に使用できます。表 1-1は名前の例を示しています。

表 1-1 カスタムファイルの命名規則

カスタムファイル	命名規則
`abccorp-secure.driver`	会社のプレフィックス
`abcc-nj-secure.driver`	会社の株式記号、場所
`abbcorp-nj-webserver.driver`	会社、場所、アプリケーションの種類
`abc-nj-trading-webserver.driver`	会社、場所、組織、アプリケーションの種類

- 以下の Solaris Security Toolkit ファイルがシステムに適合しているか確認してください。これらのファイルをカスタマイズするには、元のファイルをコピーし、コピーの名前を `user.init` および `user.run` に変更して、コピーの内容を変更または追加します。

`Drivers/user.init.SAMPLE`	グローバルパラメタのカスタマイズに使用します
`Drivers/user.run.SAMPLE`	グローバル関数のカスタマイズに使用します

注 – pkgrm コマンドで SUNWjass を削除しても、user.init および user.run ファイルが作成されている場合、これらのファイルは削除されません。この動作は、本来の Solaris Security Toolkit には含まれていないファイルがディレクトリ構造に追加されたときにも発生します。

注 – Solaris Security Toolkit 4.2 ソフトウェアでは、pkgrm コマンドに新機能が追加されています。このリリースの pdgrm コマンドは、最初の処理としてディストリビューションに含まれるすべてのファイルの整合性をチェックします。異なるファイルが見つかると、pkgrm コマンドは正しいファイルを配置するか、あるいは変更されたファイルを削除するようにシステム管理者に伝えるエラーメッセージを表示して終了します。

第2章

システムのセキュリティーの確保：手法の適用

この章では、システムのセキュリティーを確保するための方法について説明します。Solaris Security Toolkit を使用してシステムのセキュリティーを確保する場合は、あらかじめこのソフトウェアのプロセスを適用できます。

この章では、以下の項目を説明します。

計画と準備

Solaris Security Toolkit ソフトウェアを使用してシステムのセキュリティーを確保するには、適切な計画が不可欠です。計画段階では、組織のセキュリティーポリシーと基準、システムのアプリケーションと動作条件に基づいて、Solaris Security Toolkit プロファイルを開発します。この段階では以下の作業を実行します。

このマニュアルでは触れていませんが、このほかにも、リスクとセキュリティー問題の把握、インフラストラクチャーとそのセキュリティー要件の把握、アカウンタビリティー、ログ、および使用状況監査の検討なども計画段階に含まれる作業です。

リスクと利益の検討

システムを強化する際は、Solaris Security Toolkit ソフトウェアを実行後にシステムが正常に動作するように、特別な注意が必要です。また、システムの停止時間を最短に抑えるため、プロセスを最適化することが重要です。

注 – 配備済みのシステムでセキュリティー確保を実行する場合は、システムを再構築し、インストール時にセキュリティーを強化してから、システムの動作に必要なすべてのソフトウェアを再度読み込む方が効果的なことがあります。

この節では、システムのセキュリティー確保を行う前に明確に把握しておくべき事柄について説明します。リスクと利益を比較検討し、組織にとって適切なアクションを決定してください。

1. システムのサービスおよびアプリケーションの条件を把握します。

   Solaris Security Toolkit ソフトウェアを実行する前に、システムで実行されているサービスおよびアプリケーションを識別する必要があります。Solaris Security Toolkit ソフトウェアを正しく構成するために、サービスおよびアプリケーションの依存関係をすべて列挙してください。この作業を行わないと、サービスが無効になったり、必要なサービスを起動できなくなったりする可能性があります。Solaris Security Toolkit ソフトウェアで行われた変更は、ほとんどの場合、元に戻すことができます。しかし、インストール前に正しいプロファイルを開発しておくことで、ソフトウェアの実行に関連する潜在的なシテスム停止時間を制限できます。

2. システムをオフラインにして再起動する必要があることを考慮します。

   Solaris Security Toolkit ソフトウェアで行われた変更を有効にするには、システムを再起動しなければなりません。システムの重要度、システムが提供しているサービス、および保守ウィンドウの可用性によっては、ソフトウェアを実行できない場合があります。システムの停止時間とセキュリティーを強化しなかった場合のリスクとを慎重に比較検討して、決定する必要があります。

3. 機能性を検証するために、システムを数回再起動しなければならないことがあります。

   可能なかぎり、システムをミッションクリティカルな設定で実装する前に、すべての変更はまず非実働システムで行います。しかし、これは常に可能であるとは限りません。ハードウェアまたはソフトウェアの不足のため、対象となる環境を有効にミラー化できない場合もあります。Solaris Security Toolkit ソフトウェアによる強化処理の前後でテストを実施する必要があります。システムを強化した後でも、障害追跡の必要な未特定の依存関係がある可能性があります。この問題は、ほとんどの場合、この章で説明する方法で簡単に解決されます。Solaris Security Toolkit ソフトウェアの実行後に機能性の問題が検出された場合、Solaris Security Toolkit ソフトウェアの効果を元に戻すか、またはシステムのセキュリティー構成をさらに変更して失われた機能を有効にするために、システムを数回再起動しなければならないことがあります。

4. プラットフォームのセキュリティー対策は、単に強化または監査だけではありません。

   セキュリティーを強化するためにシステム構成の変更を検討するときは、セキュリティーの強化または監査が、システム、サービス、およびデータを保護するための対策の一部にすぎないことを認識しておくことが重要です。ほかに講じるべき対策および手段はこのマニュアルの対象範囲外ですが、アカウント管理、特権管理、ファイルシステムとデータの完全性、ホストベースのアクセス制御、侵入の検出、脆弱性のスキャンと分析、アプリケーションのセキュリティーに関連した問題なども考慮する必要があります。

5. システムは、すでに悪用可能な脆弱性を持っているか、あるいは悪用されている可能性があります。

   強化対象のプラットフォームが、すでに攻撃者によって脆弱性を悪用されている可能性があります。Solaris Security Toolkit ソフトウェアでは、すでに悪用されている脆弱性を保護することはできません。脆弱性が悪用されている場合には、次の処理を行なってください。

   a. システムを再インストールします。

   b. Solaris Security Toolkit ソフトウェアをインストールします。

   c. Solaris Security Toolkit ソフトウェアを使用してセキュリティーを強化します。

セキュリティーポリシー、基準、および関連ドキュメントの確認

システムのセキュリティー確保を行うにあたっては、プラットフォームのセキュリティーに関する組織のセキュリティーポリシー、基準、およびガイドラインを最初に把握しておくことが必要です。こうしたドキュメントには、組織内のすべてのシステムが準拠すべき要件および実践方法が記載されているため、Solaris Security Toolkit のプロファイルの土台となります。組織にドキュメントがない場合は、自分で作成することにより、Solaris Security Toolkit ソフトウェアをカスタマイズする能力を向上させることができます。

注 – これらのドキュメントを探すときは、ベストプラクティスまたはその他のドキュメントにリストされている可能性があることに留意してください。

セキュリティーポリシーについての詳細は、Sun BluePrints OnLine 掲載記事『Developing a Security Policy』を参照してください。この文書は、キュリティーポリシーが組織のセキュリティー計画で果たす役割を十分に理解するのに利用できます。

ポリシーステートメントは Solaris Security Toolkit のプロファイルの構成方法に直接影響します。以下に 2 つの例を示します。

例 1

- ポリシー – 組織は、強力なユーザー認証と送信データの暗号化をサポートしている管理プロトコルを使用する必要がある。
- プロファイルが受ける影響 – Telnet、FTP (File Transfer Protocol)、SNMPv1 (Simple Network Management Protocol version 1) などのクリアテキストプロトコルは使用できません。Solaris Security Toolkit の `secure.driver` はこれらのサービスを無効にするため、追加の構成は不要です。

注 – Kerberos などの拡張機能を使用すれば、Telnet と FTP は強力な認証と暗号化をサポートするように構成できます。しかし、それらのデフォルト構成は、追加されたセキュリティーレベルをサポートしません。

例 2

ポリシー – すべてのユーザーはパスワードを 30 日ごとに変更しなければならない。

プロファイルが受ける影響 – Solaris Security Toolkit ソフトウェアがパスワードの有効期限を指定するように構成できます。Solaris Security Toolkit ソフトウェアの `secure.driver` は、パスワードの有効期限を最大 8 週間 (56 日間) に設定します。ポリシーに適合するように、Solaris Security Toolkit ソフトウェアのプロファイルを変更する必要があります。『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

Solaris Security Toolkit ソフトウェアの `secure.driver` をシステム上で実行すると、パスワードの有効期限が有効になります。ただし、既存のユーザーについては、ユーザーがパスワードを変更するまではこの有効化が既存のユーザーに影響することはありません。既存のユーザーに対してパスワードの有効期限を有効にするには、各ユーザーアカウントで `passwd`(1) コマンドを呼び出します。既存ユーザーのパスワードを強制的に変更するには、`passwd -f` コマンドを使用できます。`passwd`(1) コマンドの詳細は、Solaris 10 OS Reference Collection を参照してください。

アプリケーションおよびサービス要件の決定

この作業を行うと、システムを強化した後でもサービスが正常に動作します。この作業は以下の手順で行います。

- 23 ページの「アプリケーションおよびサービスインベントリの識別」
- 23 ページの「サービス要件の決定」

アプリケーションおよびサービスインベントリの識別

アプリケーション、サービス、操作または管理機能のインベントリを作成します。インベントリの作成は、システムで実際に使用されているソフトウェアを調べるために必要です。システムには多くの場合、使用されないソフトウェアや、ビジネス機能をサポートしていないソフトウェアが含まれています。

システムの構成は最小限にする必要があります。つまり、ビジネス機能のサポートに不要なソフトウェアはインストールしないようにします。システム上に不要なソフトウェアがあれば、それだけ攻撃者がシステムを悪用する機会が多くなります。また、システム上のソフトウェア数が多ければ、適用しなければならないパッチ数が増えることを意味します。Solaris OS の最小化についての詳細は、Sun BluePrints OnLine の掲載記事『Minimizing the Solaris Operating Environment for Security』を参照してください。Sun Fire システムドメインの最小化についての詳細は、Sun BluePrints OnLine 掲載文書『Part I: Minimizing Domains for Sun Fire V1280, 6800, 12K, and 15K Systems』と『Part II: Minimizing Domains for Sun Fire V1280, 6800, 12K, and 15K Systems』を参照してください。

ソフトウェアのインベントリを作成するときは、システムに常駐しているアプリケーションの他に、管理、監視、およびバックアップソフトウェアなどの基幹コンポーネントも含めてください。

サービス要件の決定

アプリケーションとサービスのインベントリを作成したら、セキュリティーの強化によって影響されるコンポーネントの依存関係があるかどうかを調べます。第三者のアプリケーションの多くは、Solaris OS で提供されるサービスを直接には使用しません。Solaris OS で提供されるサービスを直接使用するアプリケーションについては、以下の節を参照してください。

- 23 ページの「共有ライブラリ」
- 26 ページの「構成ファイル」
- 27 ページの「サービスフレームワーク」

注 – この節に挙げている例はすべて、Solaris 9 OS のものです。

共有ライブラリ

アプリケーションをサポートするために必要なライブラリを把握しなければなりません。この情報は環境をデバッグする際に最も役立ちますが、システムを強化する際の準備段階でも必要です。システムの状態が不明な場合は、できるだけ多くの情報を収集して、ソフトウェアの依存関係などを把握してください。

アプリケーションで使用されるライブラリを決定するには、インストールしている Solaris OS のバージョンに応じて、次の 3 つの方法を使用できます。この節では、各方法のコード例を示します。

- 方法 **1** - アプリケーションバイナリやライブラリなど、ファイルシステムオブジェクトの情報を取得する (コード例 2-1)。
- 方法 **2** - 稼動中のプロセスについての情報を収集し、稼動中のアプリケーションを分析する (コード例 2-2)。
- 方法 **3** - 動的に読み込まれるアプリケーションを識別し、プログラムの起動時にそのプログラムのトレースを行う (コード例 2-3)。

方法 **1**

ファイルシステムオブジェクトについての情報を取得するには、`/usr/bin/ldd` コマンドを使用します。

たとえば、ドメインネームシステム (DNS) サーバーソフトウェアのサポートに必要なライブラリを判断します。

コード例 2-1 ファイルシステムオブジェクトについての情報の取得

```
# ldd /usr/sbin/in.named
libresolv.so.2 => /usr/lib/libresolv.so.2
libsocket.so.1 => /usr/lib/libsocket.so.1
libnsl.so.1 =>    /usr/lib/libnsl.so.1
libc.so.1 =>      /usr/lib/libc.so.1
libdl.so.1 =>    /usr/lib/libdl.so.1
libmp.so.2 =>    /usr/lib/libmp.so.2
/usr/platform/SUNW,Ultra-5_10/lib/libc_psr.so.1
```

方法 **2**

実行中のプロセスから情報を取得するには、/usr/proc/bin/pldd コマンド (Solaris OS バージョン 8、9、および 10 で利用可能) を使用します。

コード例 2-2　　実行中のプロセスからの情報の取得

```
# pldd 20307
20307:  /usr/sbin/in.named
/usr/lib/libresolv.so.2
/usr/lib/libsocket.so.1
/usr/lib/libnsl.so.1
/usr/lib/libc.so.1
/usr/lib/libdl.so.1
/usr/lib/libmp.so.2
/usr/platform/sun4u/lib/libc_psr.so.1
/usr/lib/dns/dnssafe.so.1
/usr/lib/dns/cylink.so.1
```

方法 **3**

pldd コマンドは、アプリケーションによって動的に読み込まれる共有ライブラリ、およびアプリケーションがリンクされている共有ライブラリを表示します。この情報は次の truss コマンドでも取得できます。

注 – 説明を簡潔にするため、次の出力は途中までしか表示していません。

コード例 2-3　　動的に読み込まれるアプリケーションの識別

```
# truss -f -topen,open64 /usr/sbin/in.named
20357:  open("/usr/lib/libresolv.so.2", O_RDONLY)       = 3
20357:  open("/usr/lib/libsocket.so.1", O_RDONLY)       = 3
20357:  open("/usr/lib/libnsl.so.1", O_RDONLY)          = 3
20357:  open("/usr/lib/libc.so.1", O_RDONLY)            = 3
20357:  open("/usr/lib/libdl.so.1", O_RDONLY)           = 3
20357:  open("/usr/lib/libmp.so.2", O_RDONLY)           = 3
20357:  open("/usr/lib/nss_files.so.1", O_RDONLY)       = 4
```

コード例 2-3　　動的に読み込まれるアプリケーションの識別 (続き)

```
# truss -f -topen,open64 /usr/sbin/in.named
20357:  open("/usr/lib/nss_files.so.1", O_RDONLY)        = 4
20357:  open("/usr/lib/dns/dnssafe.so.1", O_RDONLY)      = 4
20357:  open("/usr/lib/dns/cylink.so.1", O_RDONLY)       = 4
20357:  open("/usr/lib/dns/sparcv9/cylink.so.1", O_RDONLY) = 4
```

このバージョンの出力には、プロセス識別子、システムコール (この場合は open) とその引数、およびシステムコールの戻り値が含まれています。戻り値から、システムコールが共有ライブラリを見つけて開くことができたことを判断できます。

共有ライブラリのリストが表示されたら、次のコマンドを使用して、共有ライブラリが属している Solaris OS パッケージを決定します。

```
# grep "/usr/lib/dns/cylink.so.1" /var/sadm/install/contents
/usr/lib/dns/cylink.so.1 f none 0755 root bin 63532 24346 \
  1018126408 SUNWcsl
```

結果の出力では、この共有ライブラリが SUNWcsl (Core、Shared Libs) パッケージに属していることが示されます。このプロセスはアプリケーションまたはサービスをサポートするために必要なパッケージの識別に役立つため、プラットフォームの最小化を実行する際に特に有用です。

構成ファイル

サービス要件は構成ファイルを通して収集することもできます。サービスを無効にするために、構成ファイルの名前が変更されたり削除されていることがあります。したがって、このプロセスは、システムの強化方法に直接的に影響します。詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

構成ファイルが使用されているかどうかを判断するには、truss コマンドを使用します。

注 – 説明を簡潔にするため、次の出力は途中までしか表示していません。

コード例 2-4 構成ファイルが使用されているかどうかの判定

```
# truss -f -topen,open64 /usr/sbin/in.named 2>&1 | \
grep -v "/usr/lib/.*.so.*"
20384:  open("/etc/resolv.conf", O_RDONLY)              = 3
20384:  open("/dev/conslog", O_WRONLY)                  = 3
20384:  open("/usr/share/lib/zoneinfo/US/Eastern", O_RDONLY) = 4
20384:  open("/var/run/syslog_door", O_RDONLY)          = 4
20384:  open("/etc/nsswitch.conf", O_RDONLY)            = 4
20384:  open("/etc/services", O_RDONLY)                 = 4
20384:  open("/etc/protocols", O_RDONLY)                = 4
20384:  open("/etc/named.conf", O_RDONLY)               = 4
20384:  open("named.ca", O_RDONLY)                      = 5
20384:  open("named.local", O_RDONLY)                   = 5
20384:  open("db.192.168.1", O_RDONLY)                  = 5
20384:  open("db.internal.net", O_RDONLY)               = 5
```

この例では、DNS サービスは `/etc/named.conf` などの構成ファイルを使用しています。前の例と同様に、戻り値がエラーを示す場合は、問題がある可能性があります。セキュリティー強化を実行する前後で結果を注意深く文書化することで、検証プロセス全体の効率を向上させることができます。

サービスフレームワーク

このカテゴリには、大規模で複雑なアプリケーションを構築するためのフレームワークまたはメタサービスが含まれます。このカテゴリに一般に含まれるフレームワークには、次のようなタイプがあります。

- ネーミングサービス (NIS (Network Information Services)、NIS+、LDAP (Lightweight Directory Access Protocol) など)
- 認証サービス (Kerberos、LDAP など)
- ユーティリティーサービス (遠隔手続き呼び出し (RPC) 機能で使用されるポートマッパーなど)

アプリケーションがこうしたサービスに依存しているかどうかは不明な場合もあります。アプリケーションの構成時に、Kerberos 領域に追加するなどの特殊な操作が必要なときは、依存関係がはっきりしています。しかし、アプリケーションの依存関係が追加の作業を必要とせず、実際の依存関係がベンダーによって文書化されていないこともあります。

RPC ポートマッパーはその例です。Solaris Security Toolkit ソフトウェアの `secure.driver` は、RPC ポートマッパーを無効にします。このため、このサービスに依存しているほかのサービスが予期しない動作をすることがあります。過去の経験から、例外処理に対してアプリケーションのコードがうまく記述されているかどうかによって、サービスは中断、停止、または異常終了します。アプリケーションが RPC ポートマッパーを使用しているかどうかを判断するには、`rpcinfo` コマンドを使用します。次に例を示します。

コード例 2-5　　RPC を使用しているアプリケーションの特定

```
# rpcinfo -p
100000    3   tcp    111  rpcbind
100000    4   udp    111  rpcbind
100000    2   udp    111  rpcbind
100024    1   udp  32777  status
100024    1   tcp  32772  status
100133    1   udp  32777
100133    1   tcp  32772
100021    1   udp   4045  nlockmgr
100021    2   udp   4045  nlockmgr
100021    3   udp   4045  nlockmgr
100021    4   udp   4045  nlockmgr
100021    1   tcp   4045  nlockmgr
```

サービス列には、`/etc/rpc` ファイルまたはネーミングサービス (LDAP など) から取得した情報が表示されます。

Sun 以外の製品で多く見られるように、このファイルにサービスのエントリがない場合、サービスフィールドには何も表示されません。この場合は、他のアプリケーションで登録されているアプリケーションの識別が難しくなります。

たとえば、rusers コマンドを考えてみましょう。このコマンドは RPC ポートマッピングサービスに依存しています。RPC ポートマッパーが実行されていないと、rusers コマンドは停止したように見えます。最終的には、次のエラーメッセージでタイムアウトになります。

```
# rusers -a localhost
localhost: RPC: Rpcbind failure
```

この問題は、プログラムがサービスと通信できないために発生します。しかし、/etc/init.d/rpc から RPC ポートマッピングサービスを起動すると、プログラムはすぐに実行して結果を返します。

別の例として、RPC ポートマッピングサービスは実行しているが、rusers サービスを実行するよう構成されていない場合を考えてみます。この場合の応答はまったく異なり、検証は比較的容易に行えます。

コード例 2-6　　rusers サービスの検証

```
# rusers -a localhost
localhost: RPC: Program not registered
# grep rusers /etc/rpc
rusersd         100002  rusers
# rpcinfo -p | grep rusers
<出力は生成されない>
```

rpcinfo コマンドで rusers サービスのエントリが表示されないことから、このサービスが有効になっていないと予測できます。この予測を検証するには、/etc/inet/inetd.conf のサービスエントリを探します。

```
# grep rusers /etc/inet/inetd.conf
# rusersd/2-3   tli     rpc/datagram_v,circuit_v     wait root
/usr/lib/netsvc/rusers/rpc.rusersd     rpc.rusersd
```

サービス行の先頭にあるコメント記号 (#) は、rusers サービスが無効になっていることを示します。このサービスを有効にするには、以下のとおり、行のコメントを解除し、SIGHUP 信号を /usr/sbin/inetd プロセスに送信します。

```
# pkill -HUP inetd
```

注 – pkill コマンドは Solaris OS バージョン 7 ～ 10 でのみ利用できます。他のバージョンの場合、ps コマンドと kill コマンドで、それぞれ、プロセスを見つけて信号を送信します。

アプリケーションが RPC 機能を使用しているかどうかを判断するには、前述の ldd コマンドを使用する方法もあります。

コード例 2-7 RPC を使用しているアプリケーションを特定する別の方法

```
# ldd /usr/lib/netsvc/rusers/rpc.rusersd
libnsl.so.1 =>    /usr/lib/libnsl.so.1
librpcsvc.so.1 =>         /usr/lib/librpcsvc.so.1
libc.so.1 =>      /usr/lib/libc.so.1
libdl.so.1 =>     /usr/lib/libdl.so.1
libmp.so.2 =>     /usr/lib/libmp.so.2
/usr/platform/SUNW,Ultra-250/lib/libc_psr.so.1
```

librpcsvc.so.1 のエントリは、ファイル名とともに、このサービスが RPC ポートマッピングサービスに依存していることを示します。

RPC ポートマッパーのほかに、FTP、SNMP、NFS (Network File System) などの一般的な OS サービスにアプリケーションが依存している場合もあります。同様の方法でこれらのサービスをデバッグし、ビジネス機能をサポートするのにこれらのサービスが実際に必要かどうかを判断することができます。たとえば、netstat コマンドを以下のとおり実行します。

```
# netstat -a | egrep "ESTABLISHED|TIME_WAIT"
```

このコマンドは、現在使用されているサービス、または最近使用されたサービスのリストを返します。以下に例を示します。

表 2-1 最近使用されたサービスの表示

```
localhost.32827      localhost.32828      49152      0 49152      0
ESTABLISHED
localhost.35044      localhost.32784      49152      0 49152      0
ESTABLISHED
localhost.32784      localhost.35044      49152      0 49152      0
ESTABLISHED
```

表 2-1　最近使用されたサービスの表示 (続き)

```
localhost.35047      localhost.35046      49152      0 49152      0
ESTABLISHED

localhost.35046      localhost.35047      49152      0 49152      0
ESTABLISHED

filefly.ssh 192.168.0.3.2969    17615     1 50320     0 ESTABLISHED
```

この例では、多くのサービスが使用されていますが、どのポートがどのサービスまたはアプリケーションによって所有されているかは不明です。この情報を取得するには、pfiles(1) コマンド (Solaris OS バージョン 8、9、10 で利用可能) を使用します。pfiles コマンドは、各プロセスで開かれているすべてのファイルの情報を表示します。

コード例 2-8　サービスまたはアプリケーションによって所有されているポートの特定

```
# for pid in `ps -aeo pid | grep -v PID`; do
> pfiles ${pid} | egrep "^${pid}:|sockname:"
> done
```

lsof (list open file) コマンドを使用すると、依存関係をより効果的かつ効率的に判断することができます。

lsof ソースコードは、次のサイトからダウンロードしてください。

ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/

lsof バイナリは、次のサイトからダウンロードしてください。

http://www.sunfreeware.com

lsof コマンドは、どのプロセスがどのファイルおよびポートを使用しているかを調べます。たとえば、前の例のポート 35047 を使用しているプロセスを特定するには、以下のコマンドを使用します。

コード例 2-9　ファイルおよびポートを使用しているプロセスの特定

```
# ./lsof -i | grep 35047
ttsession   600 root 9u  IPv4 0x3000b4d47e8     0t1  TCP
localhost:35047->localhost:35046 (ESTABLISHED)

dtexec     5614 root 9u  IPv4 0x3000b4d59e8     0t0  TCP
localhost:35046->localhost:35047 (ESTABLISHED)
```

lsof の出力は、ポート 35047 が dtexec プロセスと ttsession プロセスとの通信に使用されていることを示します。

lsof プログラムにより、ファイルシステムまたはネットワークの使用を必要とするシステム間またはアプリケーション間の依存関係を高速で調べることができる場合があります。この節で説明する情報のほとんどは、lsof プログラムの各オプションを使用して取得することができます。

注 – ここで説明した方法では、めったに使用されないサービスの依存関係は見つからないことがあります。これらの方法に加えて、Sun のドキュメントとベンダー提供のドキュメントを参照してください。

Solaris Security Toolkit プロファイルの開発および実装

計画および準備段階が終了したら、セキュリティープロファイルを開発して実装します。セキュリティープロファイルは、サイト固有のセキュリティーポリシーを実装するための、関連する構成ドライバ、強化ドライバ、およびセキュアドライバから構成されます。これには、*name*-{config|hardening|secure}.driver、スクリプト、ファイルなどがあります。

Solaris Security Toolkit ソフトウェアに用意されているセキュリティープロファイルのいずれかをカスタマイズするか、または独自のセキュリティープロファイルを開発します。組織のポリシー、基準、アプリケーション要件は、たとえわずかでも各組織で異なります。

セキュリティープロファイルをカスタマイズするには、終了スクリプト、監査スクリプト、環境変数、フレームワーク、およびファイルテンプレートを通してそのアクションを調整します。

詳細は、以下の章を参照してください。

- ソフトウェアのカスタマイズに関するガイドラインについては、第 1 章、14 ページの「Solaris Security Toolkit ソフトウェアの構成およびカスタマイズ」を参照してください。
- セキュリティープロファイルを作成する手順例については、第 7 章、112 ページの「セキュリティープロファイルの作成」を参照してください。
- ドライバのカスタマイズについての詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

  スクリプト、フレームワーク関数、環境変数、ファイルについては、必要に応じて、『Solaris Security Toolkit 4.2 リファレンスマニュアル』のほかの章を参照してください。カスタマイズする主要な環境変数は JASS_FILES と JASS_SCRIPTS の 2 つです。

多数のプラットフォームに共通する基準を設定し、かつプラットフォーム間の相違を維持するには、ネストまたは階層セキュリティープロファイルと呼ばれる方法を使用します。詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。カスタマイズしたセキュリティープロファイルを組織のポリシー、基準、および要件と比較し、変更が不適切でないか確認してください。

ソフトウェアのインストール

Solaris Security Toolkit ソフトウェアのインストール手順は、配備済みのシステムでも新しいシステムでも同じです。インストール手順についての詳細は、第 3 章を参照してください。

配備済みのシステムで、このプロセスをより簡単に短時間で行うことができる特殊な場合があります。こうした場合は、強化プロセスではなく、インストール前後の作業で対処します。

インストール前の作業の実行

配備済みシステムの強化を実行する前に、次の 2 つの重要な作業について検討し、計画します。

- バックアップ
- 検証

これらの作業は、配備済みシステムの状態を確認し、強化の前に構成の潜在的な問題を解決するのに役立ちます。

データのバックアップ

この作業は、データが誤って消去された場合に備えるために行います。問題が発生した場合、システムの構成およびデータが何らかの形でアーカイブされている必要があります。次の作業を行う必要があります。

- システムをバックアップする
- バックアップメディアの読み取りが可能か確認する
- コンテンツの復元が可能か確認する

これらの作業は、システムの構成に大幅な変更を加える前に行なってください。

システムの安定性の検証

検証作業はバックアップと同様に重要です。セキュリティーの強化などでシステムの構成を変更する前に、検証を行なってシステムが安定した作業状態であることを確認します。この検証は次の手順で行います。

- 再起動
- アプリケーションまたはサービスのテストの成功

定義済みのテストおよび承認プランの使用を推奨しますが、プランが常に利用できるとはかぎりません。プランを利用できない場合は、実際の使用方法に基づいてシステムをテストします。このプロセスでは、実行中の構成が、保存されている構成と実際に一致していることを確認します。

システムの起動時やアプリケーションの起動時に、エラーメッセージまたは警告が表示されないか調べます。修正できないエラーが発生した場合はログファイルに記録し、それがセキュリティーの強化中に発生する問題の潜在的な原因にならないようにします。ログファイルを確認するときは、以下のシステム、サービス、およびアプリケーションのログを表示します。

- `/var/adm/messages`
- `/var/adm/sulog`
- `/var/log/syslog`
- `/var/cron/log`

システムを再起動しても、エラーまたは警告メッセージ、あるいは未知のエラーまたは警告が表示されなければ、この作業は完了です (既知のエラーや警告はすべて文書化されている)。システムを再起動して安定した作業状態にしてください。検証プロセスにおいて実行中のシステム構成と保存されている構成との相違を発見した場合は、組織の変更管理ポリシーおよびプロセスを再評価し、原因を特定します。

インストール後の作業の実行

インストール後の作業はインストール前の作業の拡張であり、セキュリティーの強化によってシステムまたはアプリケーションに新たな障害が発生していないことを確認するために行います。主に、システムおよびアプリケーションのログファイルを表示して確認します。システムのセキュリティー強化およびそれに続く再起動後に生成されたログファイルが、セキュリティーの強化前に取得されたログファイルと同じでなければなりません。起動されたサービスが減少したためにメッセージ数が少ないことがありますが、最も重要なのは、新しいエラーまたは警告メッセージが表示されていないことです。

ログファイルの確認に加えて、機能性のテストを行います。アプリケーションが異常終了してもログエントリが生成されない場合もあります。検証についての詳細は、次の節を参照してください。

アプリケーションおよびサービスの機能性の検証

システムのセキュリティーを確保するプロセスの最後に、システムが提供するアプリケーションとサービスが正しく機能することを検証します。あわせて、セキュリティーポリシーの要件がセキュリティープロファイルで正常に実装されたことを検証します。この作業は、すべての問題が検出されて即座に修正されるように、プラットフォームの強化直後に完璧に実行します。この作業は、セキュリティープロファイルのインストールの検証と、アプリケーションおよびサービスの機能性の検証という 2 つの作業に分かれています。

セキュリティープロファイルのインストールの検証

Solaris Security Toolkit ソフトウェアがセキュリティープロファイルを正常にインストールしたことを検証するには、インストールログファイル `jass-install-log.txt` を確認します。このファイルは、`/var/opt/SUWWjass/runs` の下の各強化処理または監査処理 (処理の開始時間) 固有のディレクトリにインストールされます。

注 – Solaris Security Toolkit ソフトウェアがシステムに対して行った操作を確認するには、このログファイルを参照してください。実行のたびに、その開始時刻に基づいて、ディレクトリ内に新しいログファイルが保存されます。

プロファイルのインストールに加えて、システムのセキュリティー構成も評価します。検査は手動で行うか、またはツールを使用して自動的に行います。

アプリケーションおよびサービスの機能性の検証

アプリケーションおよびサービスを検証するには、定義済みのテストおよび承認プランを実行します。このプランは、システムまたはアプリケーションのさまざまなコンポーネントを検査して、利用可能な作業状態であることを確認します。このような計画を用意できない場合は、システムの使用方法に応じて妥当な方法でテストします。このプロセスでは、セキュリティーの強化によってアプリケーションまたはサービスの機能が影響を受けていないことを確認します。

システムのセキュリティー強化後にアプリケーションまたはサービスの誤動作を検出した場合は、アプリケーションログファイルで問題を確認します。一般的に、`truss` コマンドを使用してアプリケーションの問題箇所を見つけることができます。不具合な箇所がわかったら、問題を絞り込んで、Solaris Security Toolkit ソフトウェアで行われた変更を特定します。

システムのセキュリティーの維持

多くの組織に共通している誤りは、セキュリティーに対してインストール時にのみ注意を払い、以降はまったく、あるいはほとんど確認を行わないことです。セキュリティーの維持は常時実行されているプロセスであり、定期的な確認および見直しが必要です。

システムのセキュリティー構成は、時間がたつにつれて他人に知られようになるため、セキュリティーを維持するには用心しなければなりません。たとえば、システムの脆弱性などは次第に知られるようになります。

次に、システムセキュリティーの維持に関する基本的なガイドラインを示します。

- パッチをインストールするときは、その前後でシステムのセキュリティー状態を確認してください。また、最新のパッチを使用してシステムを絶えず更新することも重要です。

  Solaris OS パッチのインストールには追加のソフトウェアパッケージが含まれる場合があり、それによって既存のシステム構成が上書きされることがあります。Solaris Security Toolkit ソフトウェアはパッチを適用する場合に役立ちます。システム上で繰り返し実行できるため、パッチをインストールした後でシステムのセキュリティーを確保することができます。パッチをインストールした後で、適切なドライバを使用してソフトウェアを実行し、定義されているセキュリティーポリシーにシステム構成を一致させます。使用している Solaris Security Toolkit ソフトウェアのバージョンが、パッチで追加された新しい機能をサポートしていない可能性もあるため、さらにシステムのセキュリティーを手動で確認してください。

- システムのセキュリティーを常時監視し、未承認の動作が行われないようにします。システムアカウント、パスワード、およびアクセスパターンを確認します。システムで現在行われている動作についての有効な情報が示されている場合があります。

- 一元管理する `syslog` リポジトリを配備および維持し、`syslog` メッセージを収集して分析します。これらのログから有用な情報が得られることがあります。

- 包括的な脆弱性および監査戦略を構築し、システムの構成を監視および管理します。システムのセキュリティーを常に維持していく上で、この要件は特に重要です。

- 最新バージョンの Solaris Security Toolkit ソフトウェアでシステムを定期的に更新します。

Solaris Security Toolkit ソフトウェアには、インストール時に使用できるデフォルトのセキュリティープロファイルが用意されています。

第3章

セキュリティーソフトウェアのアップグレード、インストール、および実行

この章では、Solaris Security Toolkit ソフトウェアおよびその他のセキュリティー関連ソフトウェアをダウンロードし、アップグレードまたはインストールを行なって実行する手順について説明します。スタンドアロンモード、または JumpStart モードのいずれかに環境を構成する手順、およびサポートを入手するための手順を含みます。

ソフトウェアのアップグレードまたはインストールを行い、構成して実行するには、この節で説明する手順およびプロセスに従ってください。ここでは、追加のセキュリティーソフトウェアのダウンロード手順、実行例、およびガイドラインについても記載しています。

Solaris Security Toolkit ソフトウェアはスタンドアロン製品ですが、ダウンロード可能なほかのセキュリティーソフトウェアと組み合わせて使用するともっとも効果的です。このようなセキュリティソフトウェアには、SunSolve OnLine から入手できる最新の推奨およびセキュリティパッチクラスタ、Solaris OS 用 Secure Shell ソフトウェア (Solaris OS で提供されていない場合)、Solaris OS および Sun 以外のソフトウェアのアクセス権を強化するためのアクセス権および所有権変更ソフトウェア、Sun のファイルと実行ファイルの完全性を検証するための完全性検証バイナリが含まれます。

この章では、次の作業について説明します。

計画とインストールの事前作業の実施

Solaris Security Toolkit ソフトウェアを使用してシステムのセキュリティーを確保するには、適切な計画が不可欠です。ソフトウェアをインストールする前の計画についての詳細は、第 2 章を参照してください。

ソフトウェアを配備済みのシステムにインストールする場合、インストール前の作業についての詳細は、33 ページの「インストール前の作業の実行」を参照してください。

ソフトウェアの依存関係

Solaris Security Toolkit 4.2 ソフトウェアは、`SUNWloc` パッケージに依存しています。このパッケージが存在しない場合、Solaris Security Toolkit は正常に動作しません。

サポートされている Solaris オペレーティングシステムのバージョンについては、xxi ページの「サポートされる Solaris OS のバージョン」を参照してください。

サポートされている System Management Services (SMS) ソフトウェアのバージョンについては、xxii ページの「サポートされる SMS のバージョン」を参照してください。

モードの決定

セキュリティー保護がなされていないシステムが侵入者によって攻撃される時間を制限するために、OS のインストール中またはインストール直後にシステムを強化します。Solaris Security Toolkit ソフトウェアを使用してシステムのセキュリティーを強化する前に、環境に合わせてソフトウェアを適切に構成する必要があります。

Solaris Security Toolkit ソフトウェアではモジュラーフレームワークが提供されています。JumpStart を使用していない場合、Solaris Security Toolkit ソフトウェアのフレームワークの柔軟性により、JumpStart をあとで使用するための準備を行うことができます。JumpStart を使用している場合は、Solaris Security Toolkit ソフトウェアの機能を利用して既存の JumpStart アーキテクチャーに統合することができます。

次に、スタンドアロンモードと JumpStart モードについて説明します。

スタンドアロンモード

スタンドアロンモードでは、Solaris Security Toolkit ソフトウェアは Solaris OS シェルプロンプトから直接実行します。このモードでは、セキュリティーの変更または更新が必要なシステムで、Solaris Security Toolkit ソフトウェアを使用することができます。システムを停止して OS を新規にインストールする必要はありません。しかし、可能なかぎり、オペレーティングシステムを初めからインストールし直してセキュリティーを確保することを推奨します。

スタンドアロンモードは、パッチまたは Sun 以外のソフトウェアをインストールしたあとでシステムを強化する場合に特に便利です。Solaris Security Toolkit ソフトウェアをシステム上で繰り返し実行しても、問題は発生しません。Solaris Security Toolkit ソフトウェアによって修正されたファイルがパッチによって上書きまたは変更されることがあるため、ソフトウェアを再実行すると、パッチのインストールによって取り消されたセキュリティー設定を再度実装することができます。

注 – 実働環境では、パッチを現在の環境にインストールする前に、テストおよび開発環境で実行してください。

スタンドアロンモードは配備済みのシステムのセキュリティーを即座に強化するための最良の方法の 1 つです。44 ページの「セキュリティーソフトウェアのダウンロード」で示されているダウンロードおよびインストール手順を実行するだけで、JumpStart を使用せずに Solaris Security Toolkit ソフトウェアをアーキテクチャーに統合することができます。

JumpStart モード

JumpStart テクノロジは、ネットワークベースで Solaris OS をインストールするための Sun のメカニズムであり、インストール中に Solaris Security Toolkit スクリプトを実行することができます。このマニュアルは、読者が JumpStart テクノロジについて熟知しており、既存の JumpStart 環境を利用できることを前提にしています。JumpStart テクノロジについての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

Solaris Security Toolkit 4.2 パッケージの場所は変更できます。適切なオプションを使用して `pkgadd` コマンドを実行することにより、どのディレクトリにでもこのパッケージをインストールできます。`JASS_HOME_DIR` が JumpStart サーバーの基本ディレクトリになります。

Solaris Security Toolkit ソフトウェアを JumpStart アーキテクチャーに統合するための手順は簡単です。JumpStart サーバーを構成する方法については、第 5 章を参照してください。

アップグレード手順

この節では、システムを Solaris Security Toolkit 4.0 または 4.1 ソフトウェアから Solaris Security Toolkit 4.2 ソフトウェアにアップグレードする方法について説明します。アップグレードは、Solaris OS のアップグレードとともに行うことも、Solaris OS のアップグレードとは別に行うこともできます。システムの強化は、Solaris オペレーティングシステム上で Solaris Security Toolkit ソフトウェアを使用して行います。手順は、バージョン 4.0、4.1 のどちらからアップグレードする場合も同じです。この節で説明する手順に従うことで、既存のカスタマイズの上書きが防止されます。このため、必ず説明されているとおりに進めてください。

注意 – 一度にインストールできるのは、1 つのバージョンの Solaris Security Toolkit だけです。

Solaris Security Toolkit 4.2 ソフトウェアでは、`pkgrm` コマンドに新機能が追加されています。このリリースの `pdgrm` コマンドは、最初の処理としてディストリビューションに含まれるすべてのファイルの整合性をチェックします。異なるファイルが見つかると、`pkgrm` コマンドは正しいファイルを配置するか、あるいは変更されたファイルを削除するようにシステム管理者に伝えるエラーメッセージを表示して終了します。

ドライバは、Solaris Security Toolkit がインストールされている `Drivers` サブディレクトリに入っています。ユーザーが作成したドライバもこのディレクトリに置かれます。`pkgrm` コマンドで `SUNWjass` を削除すると、Solaris Security Toolkit に付属していたドライバとユーザーが変更したドライバは削除されます。Solaris Security Toolkit に付属のドライバとは別の名前でカスタムドライバを追加してあった場合には、それらのカスタムドライバは (削除されず) 残ります。

注意 – ドライバを変更した場合には、アップグレードを行う前にそのドライバを保存する必要があります。Solaris Security Toolkit ソフトウェアのオリジナルファイルは変更しないでください。ドライバファイルを変更するのではなく、新しいファイルにドライバファイルをコピーしてその新しいファイルを変更してください。

▼ Solaris Security Toolkit ソフトウェアと Solaris オペレーティングシステムをアップグレードする

1. **システムをアップグレードするために最善の方法 (システムのバックアップまたは Solaris アップグレードの利用) に従います。**

2. Solaris Security Toolkit ソフトウェアの旧バージョンをアンインストールします。
3. Solaris Security Toolkit 4.2 ソフトウェアをインストールします。
4. 旧バージョンの Solaris Security Toolkit ドライバとユーザー指定のドライバを使用している、アップグレードされたシステムに対し、Solaris Security Toolkit 4.2 ソフトウェアを監査モードで実行します。

   ユーザー指定のドライバは、Drivers ディレクトリに置かれていなければなりません。このディレクトリに存在すれば、jass-execute または強化処理に指定できます。
5. 次に示す処理のどちらか一方を行います。

   a. エラーが発生しない場合は、手順 6 に進みます。

   b. 実行中にエラーが発生する場合 (インストールされていない実行制御スクリプトが変更されている場合や、サービスを FMRI を使用して制御する必要がある場合など) はそれらのエラーを修正し、エラーが発生しなくなるまで手順 4 と 5 を繰り返します。
6. カスタマイズしたドライバを secure.driver と比較し、カスタマイズしたドライバに新しい終了スクリプトまたは監査スクリプトを追加すべきかどうかを判断します。
7. 次に示す処理のどちらか一方を行います。

   a. スクリプトがすべてそろっていれば、手順 8 に進みます。

   b. 欠如しているスクリプトがあれば、それらを追加し、必要なスクリプトがすべて含まれるまで手順 4、5、6、および 7 を繰り返します。
8. Solaris Security Toolkit 4.2 を強化モードで実行します。
9. Solaris Security Toolkit 4.2 を監査モードで実行し、エラーがないことを確認します。
10. システムセキュリティーの構成と状態を調べ、セキュリティー要件を満たしているかどうかを確認します。
11. 次に示す処理のどちらか一方を行います。

    a. システムが要件を満たしている場合は、手順 12 に進みます。

    b. システムが要件を満たしてない場合は、使用されているドライバを更新し、手順8に戻ります。
12. システムを十分に検証し、必要なネットワークサービスを提供しているかを確認するとともに、すべてのアプリケーションが問題なく稼働しているかを確認します。
13. エラーが検出される場合は、使用されているドライバを更新し、手順 8 に戻ります。

以上の操作でアップグレードが完了します。

▼ Solaris Security Toolkit ソフトウェアだけをアップグレードする

1. 旧バージョンの Solaris Security Toolkit ソフトウェアをアンインストールします。
2. Solaris Security Toolkit 4.2 ソフトウェアをインストールします。
3. 42 ページの「Solaris Security Toolkit ソフトウェアと Solaris オペレーティングシステムをアップグレードする」の手順 4 に進みます。

Solaris OS のみのアップグレード

Solaris Security Toolkit 4.2 ソフトウェアはすでにインストール済みで、Solaris OS だけをアップグレードするという場合 (Solaris 8 OS から Solaris 10 OS へのアップグレードなど) は、Solaris Security Toolkit 4.2 ソフトウェアをアンインストールする必要はありません。Solaris OS アップグレードが完了した時点で、Solaris Security Toolkit 4.2 を監査モードで実行し、システムのセキュリティー構成を調べ、エラーが存在しないことを確認します。

セキュリティーソフトウェアのダウンロード

システムのセキュリティーを強化するには、まず、ソフトウェアセキュリティーパッケージを対象のシステムにダウンロードします。ここでは、以下の作業について説明します。

- 45 ページの「Solaris Security Toolkit ソフトウェアのダウンロード」
- 46 ページの「推奨パッチクラスタソフトウェアのダウンロード」
- 48 ページの「FixModes ソフトウェアのダウンロード」
- 49 ページの「OpenSSH ソフトウェアのダウンロード」
- 50 ページの「MD5 ソフトウェアのダウンロード」

注 – ここで説明するソフトウェアの中で、Solaris Security Toolkit ソフトウェア、推奨およびセキュリティーパッチクラスタ、FixModes、および message-digest 5 (MD5) アルゴリズムソフトウェアは必須です。OpenSSH の代わりに、さまざまなベンダーから提供されている市販の Secure Shell ソフトウェアを使用することもできます。Secure Shell ソフトウェアをすべてのシステムにインストールして使用します。Solaris 9 または Solaris 10 OS の場合は、OS に含まれている SSH (Secure Shell) を使用してください。Solaris 10 OS の場合は、MD5 チェックサム用に含まれている `/usr/bin/digest` コマンドを使用してください。

Solaris Security Toolkit ソフトウェアのダウンロード

Solaris Security Toolkit ソフトウェアは、Solaris OS パッケージ形式で配布されています。最初に、Solaris Security Toolkit ソフトウェアをダウンロードします。スタンドアロンモードの場合は Solaris Security Toolkit ソフトウェアを使用するサーバーに、JumpStart モードの場合は JumpStart サーバーにインストールします。

注 – 次の手順では、ファイル名のバージョン番号が記載されていません。常に最新バージョンのファイルをダウンロードしてください。

このマニュアルでは、これ以降、`JASS_HOME_DIR` 環境変数は Solaris Security Toolkit ソフトウェアのルートディレクトリを指します。デフォルトでは、このディレクトリは `/opt/SUNWjass` です。

▼ pkg バージョンをダウンロードする

1. **ソフトウェア配布ファイル (`SUNWjass-`*n.n*`.pkg.tar.Z`) をダウンロードします。**
   ソースファイルは以下の場所にあります。
   `http://www.sun.com/security/jass`

注 – ソフトウェアをダウンロードできない場合は、ブラウザに統合されている「Save As」オプションを使用してください。

2. uncompress コマンドを以下のとおり使用して、サーバー上のディレクトリにソフトウェア配布ファイルを圧縮解除します。

```
# uncompress SUNWjass-n.n.pkg.tar.Z
```

3. 次のように tar コマンドを実行し、ソフトウェア配布パッケージを解凍します。

```
# tar -xvf SUNWjass-n.n.pkg.tar
```

4. pkgadd コマンドを次のように使用して、ソフトウェア配布ファイルをサーバー上のディレクトリにインストールします。

```
# pkgadd -d SUNWjass-n.n.pkg SUNWjass
```

ここで、*n.n* はファイルの最新のバージョン番号です。

このコマンドを実行すると、/opt 内に SUNWjass ディレクトリが作成されます。すべての Solaris Security Toolkit ディレトリおよび関連ファイルがこのサブディレクトリに格納されます。

推奨パッチクラスタソフトウェアのダウンロード

Sun では、Solaris OS のパフォーマンス、安定性、機能性、およびセキュリティーの問題を修正するためのパッチをリリースしています。システムのセキュリティーを確保するには、最新のパッチクラスタをインストールすることが不可欠です。最新の Solaris OS 推奨およびセキュリティパッチクラスタをシステムにインストールするために、最新のパッチクラスタをダウンロードする方法を以下に説明します。

注 – パッチは、非実働システムで、または保守ウィンドウの表示中に評価およびテストしてからインストールしてください。

▼ 推奨パッチクラスタソフトウェアをダウンロードする

パッチクラスタをインストールする前に、個々のパッチの README ファイルおよび入手したその他の情報をご一読ください。インストールの前に知っておくと役立つ提案や情報が含まれている場合があります。

1. 次の SunSolve OnLine Web サイトから、最新のパッチクラスタをダウンロードします。

   http://sunsolve.sun.com

2. 右側のナビゲーションバー上にある「パッチやアップデート」リンクをクリックします。

3. 「推奨パッチクラスタ」リンクをクリックします。

4. 「Solaris 推奨・セキュリティパッチクラスタ」ボックスで適切な Solaris OS のバージョンを選択します。

   この例では、Solaris 10 OS を選択します。

5. HTTP または FTP ラジオボタンを使用して、ダウンロードオプションを選択します。

   ブラウザウィンドウに「ファイルのダウンロード」ダイアログボックスが表示されます。

6. ファイルをローカルシステムに保存します。

7. セキュリティー強化を実行するシステムにファイルを安全に移動します。

   セキュアコピーコマンド scp(1) を使用するか、セキュリティー保護されたファイル転送が可能なほかの方法を使用します。

   scp コマンドを以下のとおり使用します。

```
# scp 10_Recommended.zip target01:
```

8. ファイルを /opt/SUNWjass/Patches ディレクトリに移動して、圧縮解除します。

   次に例を示します。

コード例 3-1　パッチファイルの /opt/SUNWjass/Patches ディレクトリへの移動

```
# cd /opt/SUNWjass/Patches
# mv /directory in which file was saved/10_Recommended.zip .
# unzip 10_Recommended.zip
Archive:     10_Recommended.zip
   creating: 10_Recommended/
  inflating: 10_Recommended/CLUSTER_README
  inflating: 10_Recommended/copyright
  inflating: 10_Recommended/install_cluster
[. . .]
```

パッチクラスタソフトウェアは、ほかのセキュリティーパッケージをダウンロードして Solaris Security Toolkit ソフトウェアを実行したあとで、自動的にインストールされます。

注 – 推奨およびセキュリティーパッチクラスタを `/opt/SUNWjass/Patches` ディレクトリに保存しないと、Solaris Security Toolkit ソフトウェアの実行時に警告メッセージが表示されます。新しいリリースの OS の場合など、パッチクラスタを適用しないときは、このメッセージは無視してもかまいません。

FixModes ソフトウェアのダウンロード

FixModes は、Solaris OS のディレクトリおよびファイルのデフォルトのアクセス権を強化するソフトウェアパッケージです。こうしたアクセス権を強化することにより、全体的なセキュリティーを大幅に向上させることができます。アクセス権を制限すると、悪意のあるユーザーがシステムに対する特権を容易に取得できなくなります。

注 – Solaris 10 OS リリースでは、FixModes ソフトウェアによって変更されたオブジェクトのデフォルトのアクセス権が大幅に改善されています。このため、このリリースではこのソフトウェアは不要になりました。したがって、Solaris 10 OS を稼働させているシステムで `install-fixmodes` の終了スクリプトと監査スクリプトを使用することはできません。

▼ FixModes ソフトウェアをダウンロードする

1. **以下の場所から、プリコンパイル済みの FixModes バイナリをダウンロードします。**

   `http://www.sun.com/security/jass`

   FixModes ソフトウェアは Solaris OS システム用にフォーマットされた、プリコンパイル済みの圧縮されたパッケージバージョンで配布されます。ファイル名は `SUNBEfixm.pkg.Z` です。

2. **セキュリティー強化を実行するシステムにファイルを移動します。`scp` コマンドを使用するか、セキュリティーで保護されたファイル転送が可能な他の方法を使用します。**

   `scp` コマンドを以下のとおり使用します。

```
# scp SUNBEfixm.pkg.Z target01:
```

3. SUNBEfixm.pkg.Z ファイルを解凍し、Solaris Security Toolkit の Packages ディレクトリ (/opt/SUNWjass/Packages) に保存します。次のコマンドを使用します。

```
# uncompress SUNBEfixm.pkg.Z
# mv SUNBEfixm.pkg /opt/SUNWjass/Packages/
```

FixModes ソフトウェアは、他のセキュリティーパッケージをダウンロードして Solaris Security Toolkit ソフトウェアを実行した後で、自動的にインストールされます。

OpenSSH ソフトウェアのダウンロード

セキュリティーが確保される環境では、ユーザーとの対話セッションを保護するために、強力な認証および暗号化を使用する必要があります。少なくともネットワークアクセスは暗号化されなければなりません。

暗号化を実行するための最も一般的なツールは Secure Shell ソフトウェアです。このソフトウェアは Solaris OS に付属していますが、市販品やフリーウェアを利用することもできます。Solaris Security Toolkit ソフトウェアによるあらゆるセキュリティーの変更を実装するには、Secure Shell ソフトウェアが必要です。

注 – Solaris 9 または Solaris 10 OS を稼働させている場合は、OS に付属の Secure Shell バージョンを使用してください。このバージョンの Secure Shell は基本セキュリティーモジュール (BSM) などのほかの Solaris OS セキュリティー機能と統合されており、Sun のサポート組織によってサポートされています。

Solaris Security Toolkit ソフトウェアを実行すると、暗号化されないユーザーとの対話サービスおよびデーモンがシステム上ですべて無効になります。特に、in.telnetd、in.ftpd、in.rshd、in.rlogind などのデーモンが無効になります。

Secure Shell を使用すると、Telnet および FTP を使用する場合と同様にシステムにアクセスすることができます。

▼ OpenSSH ソフトウェアをダウンロードする

注 – サーバーで Solaris 9 または Solaris10 OS を稼働させている場合は、次に示す OpenSSH のインストール手順は省略し、OS に付属の Secure Shell ソフトウェアを使用できます。Solaris 10 OS を稼働させているシステムでは install-ssh の終了スクリプトと監査スクリプトを使用することはできません。

- **次の Sun BluePrints OnLine 掲載記事または Sun BluePrints マニュアルを入手し、ソフトウェアのダウンロード方法を参照してください。**
  - OpenSSH をコンパイルおよび配備する方法を記載した Sun BluePrints OnLine 掲載記事『Building and Deploying OpenSSH on the Solaris Operating Environment』は、次の場所から入手できます。

    `http://www.sun.com/blueprints`
  - Sun BluePrints 書籍『Secure Shell in the Enterprise』は、書店で購入できます。

OpenSSH ソフトウェアは、他のセキュリティーパッケージをダウンロードして Solaris Security Toolkit ソフトウェアを実行した後で、自動的にインストールされます。

注意 – セキュリティー強化を実行するシステムで、OpenSSH をコンパイルしたり、コンパイラをインストールしたりしないでください。同じ Solaris OS のバージョン、アーキテクチャー、モード (たとえば、Solaris 8 OS、Sun4U™ (`sun4u`)、および 64 ビット) を実行している別の Solaris OS システムを使用して、OpenSSH をコンパイルします。市販の SSH を使用する場合は、コンパイルは不要です。これは、侵入者によるコンパイラの使用を制限するためです。ただし、侵入者がプリコンパイル済みのツールをインストールすることは可能なため、コンパイラをローカルシステムにインストールしなかったからといって、攻撃者に対する防御が万全なわけではありません。

MD5 ソフトウェアのダウンロード

MD5 ソフトウェアは、セキュリティー強化を実行するシステムで MD5 デジタルフィンガープリントを生成します。デジタルフィンガープリントを生成し、それを Sun から提供されている正しいフィンガープリントと比較することで、未承認ユーザーによる変更や安全であるかのような装飾 (トロイの木馬化) がなされているシステムバイナリを検出します。攻撃者は、システムバイナリを変更することにより、システムへのバックドアアクセスが可能になります。姿を見せずにシステムの誤動作を発生させます。

注 – サーバーで Solaris 10 OS を稼働させている場合は、次に示す MD5 インストール手順は省略し、OS に含まれている `/usr/bin/digest` コマンドを使用できます。

▼ MD5 ソフトウェアをダウンロードする

注 – Solaris 10 システムに関するこの手順で説明しているように、Solaris Security Toolkit は MD5 ソフトウェアのインストールやインストールの監査は行いません。このリリースでは `digest`(1M) コマンドに MD5 機能が含まれるため、Solaris 10 OS を稼働させているシステムでは MD5 ソフトウェアは不要です。

---

1. **以下の Web サイトから、MD5 バイナリをダウンロードします。**

   `http://www.sun.com/security/jass`

   MD5 プログラムは圧縮されたパッケージバージョンのファイルで配布されます。

2. **セキュリティー強化を実行するシステムに `SUNBEmd5.pkg.Z` を移動します。`scp` コマンドを使用するか、セキュリティーで保護されたファイル転送が可能な他の方法を使用します。**

   `scp` コマンドを以下のとおり使用します。

```
# scp SUNBEmd5.pkg.Z target01:
```

3. **ファイルを解凍し、Solaris Security Toolkit の `Packages` ディレクトリ (`/opt/SUNWjass/Packages`) に移動します。以下のコマンドを使用します。**

```
# uncompress SUNBEmd5.pkg.Z
# mv SUNBEmd5.pkg /opt/SUNWjass/Packages/
```

   MD5 ソフトウェアを `/opt/SUNWjass/Packages` ディレクトリに保存した後で Solaris Security Toolkit ソフトウェアを実行すると、MD5 ソフトウェアがインストールされます。

   MD5 バイナリをインストールしたら、Solaris フィンガープリントデータベースを通して、システム上の実行ファイルの完全性を検証できます。Solaris フィンガープリントデータベースについての詳細は、Sun BluePrints OnLine 掲載記事『The Solaris Fingerprint Database — A Security Tool for Solaris Software and Files』を参照してください。

4. **(オプション) Solaris Fingerprint Database Companion および Solaris Fingerprint Database Sidekick ソフトウェアを以下の Sun BluePrint Web サイトからダウンロードしてインストールします。**

   `http://www.sun.com/blueprints/tools`

注 – 手順 4 はオプションとマークされていますが、どのオペレーティングシステムでもこの手順を実行すると非常に有益です。

これらのオプションのツールをインストールして、MD5 ソフトウェアとともに使用します。これらのツールはシステムバイナリを MD5 チェックサムデータベースと比較して検証するプロセスを簡単にします。これらのツールを定期的に使用して、セキュリティーで保護されたシステム上の Solaris OS バイナリおよびファイルの完全性を検証してください。

これらのツール、およびダウンロード方法についての詳細は、Sun BluePrints OnLine 掲載記事『The Solaris Fingerprint Database — A Security Tool for Solaris Software and Files』を参照してください。

ダウンロードしたセキュリティーツールは完全性を検証する必要があります。Solaris Security Toolkit ソフトウェアおよび追加のセキュリティーソフトウェアをインストールして実行する前に、MD5 チェックサムを使用して完全性を検査してください。MD5 チェックサムは Solaris Security Toolkit のダウンロードページから利用できます。

セキュリティープロファイルのカスタマイズ

Solaris Security Toolkit ソフトウェアには、さまざまなセキュリティープロファイルのテンプレートがドライバとして用意されています。ドライバで実装されるセキュリティープロファイルは、不要なサービスを無効にし、`secure.driver` によって無効になっているオプションのセキュリティー機能を有効にします。前の章で述べたとおり、デフォルトのセキュリティープロファイル、およびこれらのドライバで行なった変更が、すべてのシステムに適切なわけではありません。

Solaris Security Toolkit ソフトウェアを実行する前に、デフォルトのセキュリティープロファイルを環境に合わせてカスタマイズするか、または新しいセキュリティープロファイルを開発します。セキュリティープロファイルをカスタマイズするテクニックとガイドラインについては、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

ソフトウェアのインストールと実行

Solaris Security Toolkit ソフトウェアを実行する前に、以下の作業を完了しておくことが重要です。セキュリティー強化の大部分は、Solaris Security Toolkit ソフトウェアを実行すると自動的に行われます。

- セキュリティー強化を実行するシステム、または JumpStart サーバーに、追加のセキュリティーソフトウェアおよび Solaris Security Toolkit ソフトウェアをダウンロードします。44 ページの「セキュリティーソフトウェアのダウンロード」を参照してください。
- システムをスタンドアロンか JumpStart のいずれかのモードに構成します。40 ページの「モードの決定」を参照してください。
- 必要に応じ、環境に合わせて Solaris Security Toolkit ソフトウェアをカスタマイズしてください。
- Solaris Security Toolkit ソフトウェアおよび追加のセキュリティーソフトウェアをインストールして実行する前に、MD5 チェックサムを使用して完全性を検証します。

Solaris Security Toolkit ソフトウェアは、コマンド行または JumpStart サーバーから直接実行できます。

コマンド行のオプション、およびソフトウェアの実行に関するその他の情報については、次のいずれかを参照してください。

- 53 ページの「スタンドアロンモードでのソフトウェアの実行」
- 65 ページの「JumpStart モードでのソフトウェアの実行」

スタンドアロンモードでのソフトウェアの実行

コード例 3-2 は、スタンドアロンモードでコマンド行を使用する例を示しています。

コード例 3-2　　スタンドアロンモードでのコマンド行の使用例

```
# ./jass-execute -h

usage:

To apply this Toolkit to a system, using the syntax:
   jass-execute [-r root_directory -p os_version ]
      [ -q | -o output_file ] [ -m e-mail_address ]
      [ -V [3|4] ] [ -d ] driver

To undo a previous application of the Toolkit from a system:
   jass-execute -u [ -b | -f | -k ] [ -q | -o output_file ]
```

コード例 3-2　　スタンドアロンモードでのコマンド行の使用例 (続き)

```
       [ -m e-mail_address ] [ -V [3|4] ]

To audit a system against a pre-defined profile:
   jass-execute -a driver [ -V [0-4] ] [ -q | -o output_file ]
      [ -m e-mail_address ]

To remove saved files from a previous run of the Toolkit:
   jass-execute -c [ -q | -o output_file ]
      [ -m e-mail_address ] [ -V [3|4] ]

To display the history of Toolkit applications on a system:
   jass-execute -H

To display the last application of the Toolkit on a system:
   jass-execute -l

To display this help message:
   jass-execute -h
   jass-execute -?

To display version information for this program:
   jass-execute -v

#
```

表 3-1はコマンド行オプションの一覧と各オプションの説明です。

表 3-1　　jass-execute でのコマンド行オプションの使用

オプション	説明
-a *driver*	システムがセキュリティープロファイルに適合しているかどうかを判断します。 -b、-k、-f、-c、-d、-h、-H、-l、-p、-r、または -u オプションとの併用は避けてください。
-b	前回の強化後に手動で変更されたファイルをバックアップした後、システムを元の状態に復元します。 必ず -u オプションと併用してください。
-c	クリーンオプションを指定します。直前の Solaris Security Toolkit 実行で保存されたファイルを削除します。
-d *driver*	スタンドアロンモードで実行されるドライバを指定します。 -a、-b、-c、-f、-h、-H、-k、または -u オプションとの併用は避けてください。

表 3-1　jass-execute でのコマンド行オプションの使用 (続き)

オプション	説明
-f	強化後に手動でファイルが変更されていても、強化中に行われた変更を強制的にリセットします。 必ず -u オプションと併用してください。
-H	システム上の Solaris Security Toolkit ソフトウェアの履歴を表示します。
-h | -?	使用可能なオプションの概要を説明する jass-execute ヘルプメッセージを表示します。 単一で使用します。-h|-? に追加で指定されたオプションは無視されます。
-k	強化後にファイルに手動で行われた変更を保持します。 必ず -u オプションと併用してください。
-l	最後にシステムにインストールされた Solaris Security Toolkit アプリケーションを表示します。
-m *e-mail_address*	社内サポートに使用する電子メールアドレスを指定します。
-o *output_file*	出力ファイルのフルパスと出力ファイルを指定します。
-p *os_version*	Solaris の OS バージョンを指定します。形式は uname -r と同じです。 -r *root_directory* オプションと一緒に使用する必要があります。
-q	非出力モードを指定します。このコマンドの実行中、メッセージは表示されません。出力は JASS_REPOSITORY/ に格納されます。
-r *root_directory*	jass-execute の実行中に使用されるルートディレクトリを指定します。ルートディレクトリは / であり、Solaris Security Toolkit 環境変数 JASS_ROOT_DIR で定義されます。セキュリティー強化の対象の Solaris OS は / を介して指定できます。たとえば、別の OS ディレクトリのセキュリティーを強化するには、一時的に /mnt にマウントし、-r オプションで /mnt を指定します。 -p *os_version* オプションと併用する必要があります。

表 3-1　jass-execute でのコマンド行オプションの使用 (続き)

オプション	説明	
-u	例外が発生したとき、希望のアクションを入力するようにプロンプトで要求して、操作を元に戻します。 -a、-c、-d、-h、-l、-p、-r、または-H オプションとの併用は避けてください。	
-V *verbosity_level*	監査の詳細レベルを指定します。次の 5 つの詳細レベル (0 ～ 4) があります。	
	0	合格または不合格を示す単一行が表示されます。
	1	スクリプトごとに合格または不合格を示す行が 1 つ表示されるほか、スクリプト行がすべて表示されたそのあとに総合計スコアを示す行が 1 つ表示されます。
	2	スクリプトごとに、すべてのチェック結果が表示されます。
	3	バナーとヘッダーメッセージを示す全出力が複数行で表示されます。これはデフォルト設定です。
	4	複数行 (レベル 3 で表示される全データ) に加え、logDebug ログ関数によって生成された項目がすべて表示されます。これはデバッグ用のレベルです。
-v	このプログラムのバージョン情報を表示します。	

スタンドアロンモードにおいて jass-execute コマンドで使用できるオプションについての詳細は、以下を参照してください。

- 58 ページの「監査オプション」
- 58 ページの「クリーンオプション」
- 59 ページの「ヘルプオプションの表示」
- 61 ページの「ドライバオプション」
- 62 ページの「電子メール通知オプション」
- 62 ページの「実行履歴オプション」
- 62 ページの「最近の実行オプション」
- 63 ページの「出力ファイルオプション」
- 63 ページの「非出力オプション」
- 64 ページの「ルートディレクトリオプション」
- 64 ページの「元に戻すオプション」

使用できるドライバの詳細なリストについては、6 ページの「Drivers ディレクトリ」を参照してください。新しいバージョンのソフトウェアでは追加のドライバが含まれていることがあります。

▼ スタンドアロンモードでソフトウェアを実行する

1. secure.driver (または sunfire_15k_sc-secure.driver などの製品固有のスクリプト) を次のように実行します。

コード例 3-3　　スタンドアロンモードでのソフトウェアの実行

```
# ./jass-execute -d secure.driver

[NOTE] The following prompt can be disabled by setting
JASS_NOVICE_USER to 0.
[WARN] Depending on how the Solaris Security Toolkit is configured,
it is both possible and likely that by default all remote shell
and file transfer access to this system will be disabled upon
reboot effectively locking out any user without console access to
the system.

Are you sure that you want to continue? (YES/NO) [NO]
y

[NOTE] Executing driver, secure.driver

============================================================
secure.driver: Driver started.
============================================================

============================================================
Solaris Security Toolkit Version: 4.2.0
Node name:                        ufudu
Zone name:                        global
Host ID:                          8085816e
Host address:                     10.8.31.115
MAC address:                      8:0:20:85:81:6e
OS version:                       5.10
Date:                             Tue Jul 5 16:28:24 EST 2005
============================================================
[...]
```

使用できるドライバの詳細なリストについては、6 ページの「Drivers ディレクトリ」を参照してください。新しいバージョンのソフトウェアでは追加のドライバが含まれていることがあります。

2. **Solaris Security Toolkit ソフトウェアを実行した後、システムを再起動して変更を実装します。**

   セキュリティーの強化中に、クライアントの構成に対してさまざまな変更が行われます。たとえば、サービスの起動スクリプトを無効にする、サービスのオプションを無効にする、あるいはパッチを通して新しいバイナリまたはライブラリをインストールするなどの操作が行われます。これらの変更は、クライアントを再起動するまでは有効にならない場合があります。

3. **システムの再起動後に、変更が正しく完璧に行われていることを検証します。**

   66 ページの「システムの変更の検証」を参照してください。

4. **エラーが検出された場合、問題を修正して、Solaris Security Toolkit ソフトウェアをスタンドアロンモードで再度実行します。**

監査オプション

-a オプションを介して、Solaris Security Toolkit ソフトウェアは監査を実行し、システムがそのセキュリティープロファイルに適合しているかどうかを判断することができます。システムのファイルが変更されているかどうかだけでなく、以前に無効にされたプロセスが実行されているか、または削除されたソフトウェアパッケージが再インストールされているかどうかも検証されます。この機能についての詳細は、第 6 章を参照してください。

セキュリティープロファイルと比較してシステムを監査する場合のコマンド行形式

```
# jass-execute -a driver [ -V [0-4] ] [ -q | -o output-file ]
[ -m email-address ]
```

クリーンオプション

-c オプションは、直前の Solaris Security Toolkit 実行で保存されたファイルを削除します。クリーンオプションは、非出力 (-q)、出力 (-o)、メール (-m)、および詳細 (-V) オプションと併用できます。

コード例 3-4 に、-c オプションの使用と出力の例を示します。

コード例 3-4 -c オプションの出力例

```
# bin/jass-execute -c
Executing driver, clean.driver

Please select Solaris Security Toolkit runs to clean:
1.  July 15, 2005 at 11:41:02 (/var/opt/SUNWjass/run/20050715114102)
2.  July 15, 2005 at 11:44:03 (/var/opt/SUNWjass/run/20050715114403)
```

コード例 3-4　　-c オプションの出力例 (続き)

```
Choice ('q' to exit)?  2
[NOTE] Cleaning previous run from /var/opt/SUNWjass/run/20050715114403

================================================================================
clean.driver: Driver started.
================================================================================



================================================================================
Toolkit Version: 4.2.0
Node name:       sstzone
Zone name:       sstzone
Host ID:         80cb346c
Host address:    10.8.28.45
MAC address:     8:0:20:cb:34:6c
OS version:      5.10
Date:            Fri Jul 15 11:44:58 PDT 2005

================================================================================
clean.driver: Performing CLEANUP of /var/opt/SUNWjass/run/20050715114403.
================================================================================

================================================================================
clean.driver: Driver finished.
================================================================================

================================================================================
[SUMMARY] Results Summary for CLEAN run of clean.driver
[SUMMARY] The run completed with a total of 1 script run.
[SUMMARY] There were  Failures in   0 Scripts
[SUMMARY] There were  Errors   in   0 Scripts
[SUMMARY] There were  Warnings in   0 Scripts
[SUMMARY] There was a Note     in   1 Script
[SUMMARY] Notes Scripts listed in:
          /var/opt/SUNWjass/run/20050715114403/jass-clean-script-notes.txt
================================================================================
```

ヘルプオプションの表示

-h オプションは jass-execute のヘルプメッセージを表示します。利用可能なオプションについての概要が提供されます。

-h オプションでは次のような出力が生成されます。

コード例 3-5　　-h オプションの出力例

```
# ./jass-execute -h
To apply this Toolkit to a system, using the syntax:
   jass-execute [-r root_directory -p os_version ]
      [ -q | -o output_file ] [ -m e-mail_address ]
      [ -V [3|4] ] [ -d ] driver

To undo a previous application of the Toolkit from a system:
   jass-execute -u [ -b | -f | -k ] [ -q | -o output_file ]
      [ -m e-mail_address ] [ -V [3|4] ]

To audit a system against a pre-defined profile:
   jass-execute -a driver [ -V [0-4] ] [ -q | -o output_file ]
      [ -m e-mail_address ]

To remove saved files from a previous run of the Toolkit:
   jass-execute -c [ -q | -o output_file ]
      [ -m e-mail_address ] [ -V [3|4] ]

To display the history of Toolkit applications on a system:
   jass-execute -H

To display the last application of the Toolkit on a system:
   jass-execute -l

To display this help message:
   jass-execute -h
   jass-execute -?

To display version information for this program:
   jass-execute -v

Note that just the driver name should be specified when using the
'-d' or '-a' options. A path need not be specified as the script
is assumed to exist in the Drivers directory.

The '-u' undo option is mutually exclusive with the '-d' and '-a'
options. The default undo behavior is to ask the user what to do if
a file to be restored has been modified as the checksum is
incorrect.

The -u option can be combined with the '-k', '-b', or '-f' to
override the default interactive behavior. The use of one of these
options is required when run in quiet mode ('-q').

The '-k' option can be used to always keep the current file and
```

コード例 3-5　　-h オプションの出力例 (続き)

```
backup if checksum is incorrect. The 'b' can be used to backup the
current file and restore original if the checksum is incorrect.
The 'f' option will always overwrite the original if the checksum
is incorrect, without saving the modified original.
```

ドライバオプション

-d *driver*　オプションは、スタンドアロンモードで実行されるドライバを指定します。

-d オプションを使用してドライバを指定します。追加したスクリプトの名前の先頭に Drivers/ が付加されます。コマンド行にはスクリプト名だけを入力してください。

注 – -d オプションと -a、-b、-c、-f、-H、-h、-k、または -u オプションとの併用は避けてください。

-d *driver* オプションを使用して jass-execute を実行すると、以下のような出力が生成されます。

コード例 3-6　　-d *driver* オプションの出力例

```
# ./jass-execute -d secure.driver
[...]
[NOTE] Executing driver, secure.driver

================================================================
secure.driver: Driver started.
================================================================

================================================================
Solaris Security Toolkit Version: 4.2.0
Node name:                        ufudu
Zone name:                        global
Host ID:                          8085816e
Host address:                     10.8.31.115
MAC address:                      8:0:20:85:81:6e
OS version:                       5.10
Date:                             Tue Jul 5 16:28:24 EST 2005
================================================================
[...]
```

電子メール通知オプション

-m *e-mail_address* オプションを使用すると、スタンドアロンモードでの監査、クリーン、セキュリティー強化、および取り消し操作の出力を、実行終了時に自動的に電子メール送信できます。この電子メールレポートは、ほかのオプションを指定することによってシステム上で生成されるログや、Solaris Security Toolkit ソフトウェアによって作成されるローカルログに加えて生成されるものです。

sunfire_15k_sc-config.driver を呼び出し、電子メール通知オプションを使用して Solaris Security Toolkit ソフトウェアを実行する場合、次のように指定します。

```
# ./jass-execute -m root -d sunfire_15k_sc-config.driver
[...]
```

実行履歴オプション

-H オプションは、Solaris Security Toolkit ソフトウェアがシステムで何回実行されたかを示す簡単なメカニズムを提供します。取り消した実行も含めてすべての実行がリストされます。

-H オプションでは以下のような出力が生成されます。

コード例 3-7　　-H オプションの出力例

```
# ./jass-execute -H
Note: This information is only applicable for applications of
      the Solaris Security Toolkit starting with version 0.3.

The following is a listing of the applications of the Solaris
Security Toolkit on this system. This list is provided in
reverse chronological order:

1.   June 31, 2004 at 12:20:19 (20040631122019) (UNDONE)
2.   June 31, 2004 at 12:10:29 (20040631121029)
3.   June 31, 2004 at 12:04:15 (20040631120415)
```

出力は、Solaris Security Toolkit ソフトウェアがこのシステムで 3 回実行され、最後の実行は取り消されたことを示しています。

最近の実行オプション

-l オプションは、最近の実行を決定するためのメカニズムを提供します。これは -H オプションで常に最後にリストされる実行です。

-l オプションでは次のような出力が生成されます。

コード例 3-8　　-l オプションの出力例

```
# ./jass-execute -l

Note: This information is only applicable for applications of
      the Solaris Security Toolkit starting with version 4.2.0.

The last application of the Solaris Security Toolkit was:

1.   June 31, 2005 at 12:20:19 (20040631122019) (UNDONE)
```

出力ファイルオプション

-o *output_file* オプションを使用すると、jass-execute のコンソール出力が別の *output_file* に転送されます。*output_file* には絶対パス名を指定できます。

このオプションは、JASS_REPOSITORY ディレクトリで維持されるログには影響しません。このオプションは、低速の端末接続環境で実行する場合に特に便利です。*verbosity_level* の指定によっては、Solaris Security Toolkit の実行によって大量の出力が生成される可能性があります。

このオプションは、-a、-d、または -u オプションと併用できます。

-o オプションでは次のような出力が生成されます。

コード例 3-9　　-o オプションの出力例

```
# ./jass-execute -o /var/tmp/root/jass-output.txt -d secure.driver
[NOTE] Executing driver, secure.driver
[NOTE] Recording output to /var/tmp/root/jass-output.txt
```

非出力オプション

-q オプションを使用すると、セキュリティー強化の実行時に、コンソールからの Solaris Security Toolkit の出力を無効にします。

このオプションは、JASS_REPOSITORY ディレクトリ内に保持されるログには影響しません。-o オプションと同様、このオプションも、Solaris Security Toolkit ソフトウェアを cron ジョブを利用して実行する場合や、低速のネットワーク接続環境で実行する場合に特に便利です。

このオプションは、-a、-c、-d、または -u オプションと併用できます。

-q オプションでは次のような出力が生成されます。

コード例 3-10　　-q オプションの出力例

```
# ./jass-execute -q -d secure.driver
[NOTE] Executing driver, secure.driver
```

ルートディレクトリオプション

-r *root-directory* オプションは、jass-execute の実行中に使用されるルートディレクトリを指定します。-r オプションを使用するときは、-p オプションでプラットフォーム (OS) バージョンも指定する必要があります。-p オプションの形式は uname -r で作成されるものと同じです。

ルートディレクトリは / であり、Solaris Security Toolkit 環境変数 JASS_ROOT_DIR で定義されます。セキュリティー強化の対象の Solaris OS は / を介して指定できます。たとえば、別の OS ディレクトリのセキュリティーを強化するには、一時的に /mnt にマウントし、-r オプションで /mnt を指定します。すべてのスクリプトはその OS イメージに適用されます。

元に戻すオプション

-u オプションを介して、Solaris Security Toolkit ソフトウェアはシステムのセキュリティー強化中に行った変更を元に戻すことができます。それぞれの終了スクリプトを -u オプションで取り消すことができます。また、Solaris Security Toolkit の元に戻す機能は、実行ごとに生成されるチェックサムと密接に統合されています。この機能についての詳細は、第 4 章を参照してください。

-u オプションは次に示す 3 つのオプションと併用できます。

- -b (backup: バックアップ) オプション。前回の強化後に手動で変更されたファイルをバックアップしたあと、システムを元の状態に復元します。
- -f (force: 強制) オプション。強化後に手動でファイルが変更されていても、例外を確認することなく強化中に行われた変更を強制的にリセットします。
- -k (keep: 保持) オプション。最後の強化後にファイルに手動で行われた変更を保持します。

元に戻すコマンドのコマンド行形式

```
# jass-execute -u [ -b | -f | -k ] [ -q | -o output_file ]
        [ -m e-mail_address ] [ -V [3|4] ]
```

JumpStart モードでのソフトウェアの実行

JumpStart モードは、JumpStart サーバーの `rules` ファイルに挿入される Solaris Security Toolkit ドライバによって制御されます。

JumpStart モードを使用するように環境が構成されていない場合は、第 5 章を参照してください。

JumpStart テクノロジについての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

▼ JumpStart モードでソフトウェアを実行する

Solaris Security Toolkit ソフトウェアを JumpStart モードで実行するには、ソフトウェアを JumpStart 環境に統合し、JumpStart インストールに関連付けられている終了スクリプトの一部として呼び出します。Solaris Security Toolkit ソフトウェアを環境に統合する方法については、第 5 章を参照してください。

1. **必要なドライバの変更をすべて行った後、JumpStart インフラストラクチャーを使用してクライアントをインストールします。**

   この作業を実行するには、クライアントの `ok` プロンプトから次のコマンドを使用します。

   ```
   ok> boot net - install
   ```

   インストールが完了すると、システムは自動的に再起動します。

   システムは常に正しい構成でなければなりません。セキュリティーの強化中に、クライアントの構成に対してさまざまな変更が行われます。たとえば、サービスの起動スクリプトを無効にする、サービスのオプションを無効にする、あるいはパッチを通して新しいバイナリまたはライブラリをインストールするなどの操作が行われます。これらの変更は、クライアントを再起動するまでは有効にならない場合があります。

2. **システムが再起動されたら、変更が正しく完璧に行われていることを検証します。**

   66 ページの「システムの変更の検証」を参照してください。

3. **エラーが検出された場合は、それを修正してクライアントの OS を再インストールします。**

システムの変更の検証

システムの再起動後に以下の手順を行なって、変更が正しく完璧に行われていることを検証します。

サービスの QA 検査の実行

システムのセキュリティー強化を実行する際の重要な作業の 1 つは、システムを正常に動作させるために有効にしておくべき OS サービスを判断することです。アプリケーションによって直接使用されている Solaris OS サービスは、有効にしておかなければなりません。たとえば、システムにログインするための Secure Shell などがこれに含まれます。一方、Sun 以外のソフトウェア管理ツールのグラフィカルユーザーインタフェース (GUI) の RPC デーモンなどは、間接的に使用されるサービスです。

こうした要件の大部分は、Solaris Security Toolkit ソフトウェアを実行する前に決定しておく必要があります。22 ページの「アプリケーションおよびサービス要件の決定」を参照してください。ただし、唯一の確実な方法は、システムをインストールしてセキュリティー強化を実行したあとで品質保証 (QA) 検査を行い、必要な機能が備わっているかテストすることです。配備する新しいシステムのセキュリティーを強化したあと、それらのシステムに対して QA プランを実行する必要があります。同様に、配備後のシステムでセキュリティー強化を実行した場合も、テストを入念に実行します。必要な機能および予想される機能がすべて存在しているか確認してください。

QA プロセスで不一致が検出された場合は、以下の手順を実行します。

1. 第 2 章のガイドラインに基づいて、問題の箇所を見つけます。
2. アプリケーションが変更後の構成で実行されることを検証します。
3. Solaris Security Toolkit の実行によって行われた変更を元に戻します。
4. 問題の解決方法に基づいて、セキュリティープロファイル (ドライバ) を修正します。
5. Solaris Security Toolkit ソフトウェアを再度実行します。

最終的に、セキュリティープロファイルを実行してもシステムの必要な機能に影響を与えないことが確認されなければなりません。

構成のセキュリティー評価の実行

システムの機能性の検証に加えて、システムのセキュリティー構成を検証し、必要レベルのセキュリティーが確保されているかどうかを判断します。システムの強化または最小化の内容に応じて、確認する項目は異なる可能性があります。

少なくとも、以下の点についてシステムの構成を確認します。

- 必要な推奨およびセキュリティーパッチがすべてインストールされていることを確認します。
- 必要でかつ適切なプロセスだけが実行されていることと、それらのプロセスが正しい引数で実行されていることを確認します。
- 必要なデーモンだけが実行されていることと、それらのデーモンが正しい引数で実行されていることを確認します。
- システムの必要なポートだけが開いていることを確認します。検査は (`netstat -a` などで) ローカルに行い、Nmap などのポートスキャナでリモートでも行います。これによって、ネットワークインタフェースで利用可能なポートが判別できます。
- システムを最小化した場合、必要な Solaris OS パッケージだけがインストールされていることを確認します。

上記の確認事項は、新しく構築してセキュリティー確保を実行したシステムに対する最小限のものです。配備済みのシステムのセキュリティーを強化した場合は、基礎となる OS が不当に変更されていないか検証する必要があります。完全性を検査するには、システムのファイルシステムを読み取り専用モードでマウントし、既知の OS インスタンスから完全性検査ソフトウェアを実行するのが最良の方法です。検証には Sun BluePrints OnLine 掲載記事「The Solaris Fingerprint Database—A Security Tool for Solaris Software and Files」に記載されているツールが役立ちます。

セキュリティープロファイルの検証

システムのセキュリティーを確保し、必要なサービスおよび機能を検証した後で、監査機能を使用してセキュリティープロファイルが正しく完璧に適用されていることを確認します。この作業は 2 つの理由で非常に重要です。第一に、システムのセキュリティーが適切に強化されたことを確認するためです。第二に、システムに対して定義されたセキュリティープロファイルが Solaris Security Toolkit 構成に適切に反映されていることを確認するためです。システムの配備後の期間全体にわたり、構成情報に基づいてセキュリティープロファイルが維持されるので、この検査は不可欠です。

監査機能についての詳細は、第 6 章を参照してください。

インストール後の作業の実施

ソフトウェアを配備済みのシステムにインストールした場合、インストール後の作業についての詳細は、34 ページの「インストール後の作業の実行」を参照してください。

第4章

システムの変更のリセット

この章では、Solaris Security Toolkit ソフトウェアの強化中に実行された変更のリセット (元に戻す処理) の内容と手順について説明します。このオプションを使用すると、Solaris Security Toolkit の強化処理を実行する前の状態に、システムを自動的に戻すことができます。

この章では、以下の項目を説明します。

変更のログ作成とリセット方法について

Solaris Security Toolkit の強化処理では、その実行のたびに JASS_REPOSITORY に実行ディレクトリが作成されます。ディレクトリ名は、実行の開始日時に基づきます。画面への出力表示に加え、ディレクトリ内にファイルが作成されて、変更の追跡と操作ログの作成も行われます。

ディレクトリに保存されたファイルは、システムの変更を追跡します。このファイルがあることで、元に戻す機能が使用可能になります。

注意 – 管理者は JASS_REPOSITORY ディレクトリ内のファイルの内容を書き換えてはなりません。ファイルを変更すると、データの内容が壊れてしまい、元に戻す機能の使用時に、予期しないエラーが発生したりシステムが破損したりするおそれがあります。

Solaris Security Toolkit ソフトウェアでシステムを強化すると、JumpStart またはスタンドアロンモードのいずれのモードでも、JASS_REPOSITORY/jass-manifest.txt ファイルに変更のログが記録されます。このファイルには、変更をリセットする際に**元に戻す**機能で使用される操作が一覧表示されます。このファイルには、作成されたファイルや、コピー、移動、削除されたファイルなど、Solaris Security Toolkit ソフトウェアによって実行された強化処理についての情報が含まれます。また、ソフトウェアパッケージのインストールなど、より複雑な変更をリセットする際に必要な標準項目とカスタム項目も含まれる場合があります。強化処理を実行するたびに、jass-manifest.txt ファイルが個別に作成されます。

注 – Solaris Security Toolkit ソフトウェアの**元に戻す**機能は、マニフェストファイル内に項目がある変更のみリセットします。

元に戻す処理では、Solaris Security Toolkit 処理の実行中に作成された、JASS_REPOSITORY に保存されているマニフェストファイルが調べられます。ユーザーが使用する、バックアップ、強制、または保持オプションによっては、この処理によってバックアップファイルが元の場所に復元される場合があります。バックアップ、強制、および保持オプションの詳細は、次のページを参照してください。

- 74 ページの「バックアップオプション」
- 74 ページの「強制オプション」
- 75 ページの「保持オプション」

強化処理中にファイルがバックアップされていないと、JASS_SAVE_BACKUP 変数が user.init ファイル内で 0 として定義されるか、あるいは -c オプションが使用された場合、元に戻す機能は使用できません。詳細は、70 ページの「システムの変更を元に戻すための要件」を参照してください。

Solaris Security Toolkit の処理を元に戻しても、関連するディレクトリは削除されません。その代わり、JASS_REPOSITORY ディレクトリに jass-undo-log.txt と reverse-jass-manifest.txt の 2 つのファイルが作成されます。jass-execute -u を次回実行するときは、すでに元に戻されている処理は表示されません。強化処理は一度だけ元に戻すことができます。

システムの変更を元に戻すための要件

Solaris Security Toolkit ソフトウェアの元に戻す機能には、以下の制限事項と要件があります。

- Solaris Security Toolkit バージョン 0.3 ～ 4.2 では、スタンドアロンまたは JumpStart モードのいずれで開始された実行に対しても、元に戻す機能を使用できます。ただし、変更は、スタンドアロンモードでのみ元に戻すことができます。元に戻す機能は、JumpStart インストール時には使用できません。

- JumpStart またはスタンドアロンモードで、Solaris Security Toolkit オプションからバックアップファイルを作成しないを選択すると、元に戻す機能は使用できません。JASS_SAVE_BACKUP パラメタに 0 が設定されて、バックアップファイルのコピーの作成が無効になります。
- 実行内容は一度だけ元に戻すことができます。
- 新しい終了スクリプトを開発する場合は、必ず Solaris Security Toolkit フレームワーク機能を使用してください。この際、対応する監査スクリプトを作成し、add_to_manifest 機能を使用してマニフェストファイルにエントリを追加する必要があります。そうでないと、Solaris Security Toolkit でカスタム開発スクリプトが認識されません。
- いかなる場合も JASS_REPOSITORY ディレクトリの内容を変更しないでください。ファイルを変更すると、データの内容が壊れてしまい、元に戻す機能の使用時に、予期しないエラーが発生したりシステムが破損したりするおそれがあります。

変更を元に戻すスクリプトのカスタマイズ

Solaris Security Toolkit フレームワークでは、終了スクリプトの設計と構築が柔軟に行えます。このフレームワークを使用すれば、組織の必要性に応じて Solaris Security Toolkit ソフトウェアの機能を拡張できるほか、システムのライフサイクルに合わせてシステム構成を上手に管理することができます。

スクリプトをカスタマイズするときは、元に戻す機能が変更によってどのような影響を受けるかを理解しておくことが重要です。ヘルパー関数を使用すれば、マニフェストファイルに正しい変更が行われるので、スクリプトを簡単にカスタマイズできます。**元に戻す**機能は、マニフェストファイルの内容に基づいて強化処理をリセットします。通常、ヘルパー関数は、組織の必要性に応じてスクリプトをカスタマイズするために必要な要素を提供します。

ヘルパー関数のリストとその使用方法については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。システムコマンドの代わりに、これらのヘルパー関数を使用すれば、元に戻す処理でマニフェストファイル内の関連項目を参照できます。

場合によっては、ヘルパー関数を持たない関数を実行する必要があります。そのようなときは、add_to_manifest と呼ばれる特殊関数を使用します。この関数を使用すると、マニフェストファイルに手動で項目を追加でき、ヘルパー関数を呼び出す必要はありません。この特殊関数を使用するときは、システムと Solaris Security Toolkit リポジトリの完全性を保護するために注意が必要です。この特殊関数の使用

例としては、Sun の pkg 形式でないソフトウェアパッケージを追加する場合があります。この場合、強化処理で追加されたほかの形式のパッケージの削除方法を、**元に戻す**機能に指示する必要があります。

ヘルパー関数と特殊関数 add_to_manifest を使用すれば、Solaris Security Toolkit ソフトウェアでスクリプトのカスタマイズと元に戻す処理への変更の反映を簡単かつ、柔軟に行えます。

これらの関数を使用せずに終了スクリプトの動作を変更すると、その変更が Solaris Security Toolkit ソフトウェアに認識されません。そのため、マニフェストファイルで参照されない変更をすべて手動で元に戻す必要があります。

ほかの例を考えてみます。たとえば、システム上のファイルを変更するときは、最初にオリジナルバージョンのファイルを保存する必要があります。通常、ユーザーは /usr/bin/cp コマンドを実行することにより、この作業を Solaris Security Toolkit ソフトウェアのコンテキスト外で行います。ただし、Solaris Security Toolkit ソフトウェアのコンテキスト内では、このコマンドを直接使用すると、リスト項目を作成する必要性が Solaris Security Toolkit ソフトウェアに認識されません。cp コマンドを使用する代わりに、backup_file ヘルパー関数を使用します。この関数は、接尾辞 JASS_SUFFIX を付けてオリジナルファイルのコピーを保存し、ファイルのコピーが作成されたことを Solaris Security Toolkit ソフトウェアに知らせるリスト項目を追加します。また、この関数はファイルのチェックサムも計算します。ファイルのチェックサムは、元に戻す機能と jass-check-sum コマンドで使用されます。

手動で変更されたファイルのチェック

jass-execute -u コマンドは、強化処理の実行後に手動で変更されたファイルを自動的にチェックしますが、場合によっては、jass-check-sum コマンドを使用し、変更されたファイルを一覧表示した上で確認した方が便利なときもあります。

このコマンドを使用すると、JASS_REPOSITORY ディレクトリの内容を確認して、マニフェストファイルに示されるすべてのファイルにチェックサムを実行できるので、強化処理の際にチェックサムが記録されたあとに変更されたファイルを特定できます。このチェックを実行してから強制的な元に戻す処理を実行すれば、不要な障害追跡にかかる時間を節約できる可能性がある有用な情報が得られます。

以下に出力例を示します。

コード例 4-1　　手動で変更されたファイルの出力例

```
# ./jass-check-sum
File Name             Saved CkSum             Current CkSum
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

コード例 4-1　　手動で変更されたファイルの出力例 (続き)

```
/etc/inet/inetd.con    1643619259:6883          2801102257:6879
/etc/logadm.conf       2362963540:1042          640364414:1071
/etc/default/inetd     3677377803:719           2078997873:720
```

この出力は、強化処理の完了後に 3 つのファイルが変更されていることを示しています。

元に戻す機能でのオプションの使用

この節では、元に戻す機能を実行する際に使用できる `jass-execute -u` コマンドとオプションについて説明します。

注 – 元に戻す機能では、`-c`、`-d`、`-a`、`-h`、`-l`、または `-H` オプションを使用することはできません。非出力モードで元に戻す機能を実行する場合は、`-b`、`-k`、または `-f` オプションを指定する必要があります。

`jass-execute -u` コマンドは、**元に戻す**処理を実行する際の標準的な方法です。このコマンドは、前回の強化後に手動で変更されたファイルを検出します。Solaris Security Toolkit ソフトウェアで、強化後に手動で変更されたファイルが検出されると、次に示す応答のどれか 1 つを選択するように求められます。

1. 最新のファイルをバックアップしてからオリジナルファイル (強化前に存在していたファイル) を復元する。
2. 最新のファイルを保持し、オリジナルファイルを復元しない。
3. 手動で変更されたファイルを強制的に上書きして (この結果データが消失する可能性がある)、オリジナルファイルを復元する。
4. 常に最新のファイルをバックアップしてからオリジナルファイル (強化前に存在していたファイル) を復元する。
5. 常に最新のファイルを保持し、オリジナルファイルを復元しない。
6. 手動で変更されたファイルを常に強制的に上書きして (この結果データが消失する可能性がある)、オリジナルファイルを復元する。

前回の強化処理以降に変更されたファイルを元に戻すコマンドがどのように処理すべきかを定義する場合は、元に戻すコマンドを実行する際にバックアップ (`-b`)、保持 (`-k`)、または強制 (`-f`) オプションを指定してください。

表 4-1 に、元に戻すコマンドで使用できるコマンド行オプションを示します。各オプションについての詳細は、以降の節を参照してください。

表 4-1 元に戻すコマンドで使用するコマンド行オプション

オプション	説明
-b	前回の強化後に手動で変更されたファイルをバックアップしたあと、システムを元の状態に復元します。
-f	強化後に手動でファイルが変更されていても、強化中に行われた変更を強制的にリセットします。
-k	強化後にファイルに手動で行われた変更を保持します。
-m	出力を電子メールアドレスに送信します。
-o	出力をファイルに送信します。
-q	画面に出力を表示しません。非出力オプションとも呼ばれます。出力は JASS_REPOSITORY/jass-undo-log.txt に保存されます。
-V	元に戻す実行の詳細レベルを指定します。

バックアップオプション

-b オプションは、前回の強化後に手動で変更されたファイルを自動的にバックアップした後、ファイルを強化前の状態に戻します。手動の変更を実行するには、復元されるファイルとバックアップされるファイルを比較し、両者の違いを手動で調整する必要があります。このオプションを使用してファイルをバックアップすると、ファイルは以下のように表示されます。

```
/etc/motd.BACKUP.JASS_SUFFIX
```

強制オプション

-f オプションは、強化後に手動でファイルが変更されていても、強化中に行われた変更を強制的にリセットします。保存されているファイルのチェックサムと現在のバージョンのファイルは比較されません。そのため、強化後に手動でファイルを変更していると、変更内容が上書きされて、元に戻した後に変更内容が失われます。

元に戻す処理の完了後、変更を手動で再実装する必要がある場合もあります。また、変更の種類によっては、ファイル間の違いの調整が必要な場合もあります。

注 – これらの問題を回避するには、jass-check-sum コマンドを使用するか、前述した -b コマンド行オプションを使用します。

保持オプション

-k オプションは、強化処理の実行後にファイルに手動で行われた変更をすべて自動的に保持し、オリジナルファイルを復元しません。-k オプションは、ファイルの不一致を検出します。また、通知データを生成してそのログを記録します。オリジナルファイルによる上書きは行いません。リセットされる変更は、保存されたチェックサムが有効なものだけです。

このオプションには欠点があります。たとえば、終了スクリプトによって変更されたファイルのサブセットがあとで変更されると、システムが矛盾した状態になることがあります。

たとえば、remove-unneeded-accounts.fin 終了スクリプトを使用したとします。このスクリプトは、システム上の /etc/passwd および /etc/shadow ファイルを変更します。強化処理の終了後、ユーザーがパスワードを変更すると、/etc/shadow ファイルに関連付けられたチェックサムが Solaris Security Toolkit ソフトウェアによって保存された値と一致しなくなります。その結果、保持オプションを使用すると、/etc/passwd ファイルのみ元の状態にコピーし直されます。/etc/shadow ファイルは現在の状態のままです。したがって、2 つのファイルが一致しません。

出力ファイルオプション

-o */complete/path/to/output_file* オプションを使用すると、jass-execute のコンソール出力が別の *output_file* に転送されます。

このオプションは、JASS_REPOSITORY ディレクトリ内に保持されるログには影響しません。Solaris Security Toolkit の元に戻す処理は大量の出力を生成するので、このオプションは、低速の端末接続環境で実行する場合に特に有効です。

非出力オプション

注 – 非出力モードで元に戻す機能を実行する場合は、-b、-k、または -f オプションを指定する必要があります。

-q オプションを使用すると、Solaris Security Toolkit ソフトウェアは画面に出力を表示しなくなります。このオプションは、JASS_REPOSITORY ディレクトリ内に保持されるログには影響しません。-o オプションと同様、このオプションも、Solaris Security Toolkit ソフトウェアを cron ジョブを利用して実行する場合や、低速のネットワーク接続環境で実行する場合に特に便利です。

電子メール通知オプション

-m *e-mail_address* オプションを使用すると、完了した実行のコピーが Solaris Security Toolkit ソフトウェアによって、指定された電子メールアドレスに送信されます。他のオプションを使用すると、ログに加えて電子メールレポートも生成されます。

システムの変更を元に戻す

場合によっては、Solaris Security Toolkit の 1 つまたは複数の強化処理で行われた変更のリセットが必要になります。セキュリティー強化処理によって行われた変更がシステムに悪影響を与えているときは、変更を元に戻します。

たとえば、強化処理の実行後、SVM (Solaris Volume Manager) などの必要なサービスが無効になっていることが検出された場合は、次の処理を行います。

1. 強化処理を取り消します。
2. カスタマイズされたドライバを作成します。

   ドライバのカスタマイズ手順については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 4 章の「ドライバのカスタマイズ」を参照してください。
3. JASS_SVCS_ENABLE 環境変数を使用し、使用する SVM サービスを有効にします。

   JASS_SVCS_ENABLE の使用方法については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 7 章の「JASS_SVCS_ENABLE」を参照してください。
4. 強化処理を繰り返します。

この節では、1 つまたは複数の強化処理によって行われた変更を元に戻す方法について説明します。強化処理を効果的に元に戻すには、制限事項と要件があります。70 ページの「システムの変更を元に戻すための要件」を参照してください。

▼ Solaris Security Toolkit の実行を元に戻す

1. **システムをバックアップして再起動します。**

   元に戻す処理を実行する前に、システムをバックアップして再起動することで、システムが既知の作業状態に確実に戻る (または戻れる) ようにしておきます。

2. **jass-execute -u コマンドで使用するオプションを決めます。**

   73 ページの「元に戻す機能でのオプションの使用」を参照してください。

   以下の説明では、jass-execute -u コマンドを使用することを前提にします。

3. **標準的な -u オプションを使用して 1 つまたは複数の強化処理を元に戻すには、JASS_HOME_DIR/bin から次のコマンドを入力します。**

```
# ./jass-execute -u
```

   Solaris Security Toolkit ソフトウェアは、JASS_REPOSITORY 内のマニフェストファイルをすべて検索して、個々の強化処理に関する情報を収集します。マニフェストファイルが空または存在しない場合は、元に戻す変更は存在せず、実行する必要はないと見なされます。また、マニフェストファイルと同じディレクトリ内に jass-undo-log.txt と呼ばれるファイルが存在する場合も、処理はすでにリセット済みで、実行する必要はないと見なされます。収集プロセスが完了すると、結果が表示されます。以下に出力例を示します。

コード例 4-2　　元に戻す際に使用可能な処理の出力例

```
# ./jass-execute -u
[NOTE] Executing driver, undo.driver
Please select a JASS run to restore through:
1. January 24, 2003 at 13:57:27
(/var/opt/SUNWjass/run/20030124135727)
2. January 24, 2003 at 13:44:18
(/var/opt/SUNWjass/run/20030124134418)
3. January 24, 2003 at 13:42:45
(/var/opt/SUNWjass/run/20030124134245)
4. January 24, 2003 at 12:57:30
(/var/opt/SUNWjass/run/20030124125730)

Choice? (‘q’ to exit)?
```

   この例では、4 つの強化処理が検出されています。これらの強化処理によってシステムが変更されていますが、まだ元に戻されていません。強化処理のリストは、常に時系列の逆の順序で表示されます。リストの最初の項目が最新の強化処理です。

4. 出力を確認して元に戻す処理を決定したら、該当する番号を入力します。

どの項目を選択した場合でも、選択した値以下のインデックス番号の処理がすべて元に戻されます。つまり、最新の強化処理から選択した強化処理まで、変更の実行順序とは逆の順序で元に戻されます。前の例で考えてみると、処理 3 を選択した場合、最初に処理 1 の変更が元に戻され、次に処理 2 の変更が元に戻され、最後に処理 3 の変更が元に戻されます。

コード例 4-3 は、元に戻す処理で 2 つのマニフェストファイル項目を処理する際に生成される出力を示しています。

コード例 4-3 元に戻す処理で複数のマニフェストファイル項目を処理する場合の出力例

```
[...]

================================================================
undo.driver: Performing UNDO of
//var/opt/SUNWjass/run/20050715145837.
================================================================

[...]

================================================================
undo.driver: Undoing Finish Script: update-cron-allow.fin
================================================================

[NOTE] Undoing operation COPY.
cp -p /etc/cron.d/cron.allow.JASS.20050715145906
/etc/cron.d/cron.allow
rm -f /etc/cron.d/cron.allow.JASS.20050715145906

[NOTE] Removing a Solaris Security Toolkit-created file.
rm -f /etc/cron.d/cron.allow

[...]
```

この例では、コピー処理が元に戻され、強化処理で追加されたファイルが削除されます。元に戻す処理の出力には、システムを復元する際に使用される実際のコマンドが示されるので、システムの構成の障害追跡が必要な場合は、そのプロセスを明確に把握して参照できます。

前回の強化処理が成功した以降に変更されたファイルが Solaris Security Toolkit によって検査されてこの検査が正常に完了する場合、すべての処理とその対応マニフェストファイルが処理されて変更が取り消されるまで元に戻す処理は継続します。

Solaris Security Toolkit ソフトウェアは、JASS_REPOSITORY 内のマニフェストファイルをすべて検索して個々の強制処理に関する情報を収集することに加え、次の処理も行います。

a. 変更された各ファイルのチェックサムを比較する。

b. チェックサムファイルに不一致がある場合、通知を生成してログに記録する。

c. これらのファイルをどのように処理するかをユーザーに尋ねる。

5. **元に戻す処理で例外 (強化処理のあとに変更されたファイル) が検出された場合は、オプションの 1 つを入力します。**

注 – Solaris Security Toolkit ソフトウェアは特定の例外ファイルに対してなされたユーザーのバックアップ、保持、および強制選択を記憶するため、元に戻す処理でそのファイルが次に例外となる際にファイルに対する選択を行う必要がありません。

以下に、例外と例外を処理するためのオプションを示す出力例を示します。

コード例 4-4 例外を元に戻す場合の出力例

```
[...]

================================================================
undo.driver: Undoing Finish Script: enable-process-accounting.fin
================================================================

[NOTE] Undoing operation COPY.
[WARN] Checksum of current file does not match the saved value.
[WARN]    filename = /var/spool/cron/crontabs/adm
[WARN]    current  = db27341e3e1f0f27d371d2e13e6f47ce
[WARN]    saved    = a7f95face84325cddc23ec66d59374b0

Select your course of action:
1. Backup - Save the current file, BEFORE restoring original.
2. Keep   - Keep the current file, making NO changes.
3. Force  - Ignore manual changes, and OVERWRITE current file.

NOTE: The following additional options are applied to this and ALL
subsequent files:
4. ALWAYS Backup.
5. ALWAYS Keep.
6. ALWAYS Force.

Enter 1, 2, 3, 4, 5, or 6:
```

この例で項目 1 を選択すると、次の出力が表示されます。

コード例 4-5　　元に戻す処理でバックアップオプションを選択した場合の出力例

```
Enter 1, 2, 3, 4, 5, or 6: 1

[WARN] Creating backup copies of some files may cause unintended
effects.
[WARN] This is particularly true of /etc/hostname.[interface]
files as well as crontab files in /var/spool/cron/crontabs.

[NOTE] BACKUP specified, creating backup copy of
/var/spool/cron/crontabs/adm.
[NOTE] File to be backed up is from an undo operation.
[NOTE] Copying /var/spool/cron/crontabs/adm to
/var/spool/cron/crontabs/adm.BACKUP.JASS.20050715151817
cp -p /var/spool/cron/crontabs.JASS/adm.JASS.20050715151719
/var/spool/cron/crontabs/adm
rm -f /var/spool/cron/crontabs.JASS/adm.JASS.20050715151719

[NOTE] Undoing operation COPY.
cp -p /var/spool/cron/crontabs.JASS/root.JASS.20050715151717
/var/spool/cron/crontabs/root
rm -f /var/spool/cron/crontabs.JASS/root.JASS.20050715151717

[NOTE] Undoing operation MAKE DIRECTORY.
rmdir /var/spool/cron/crontabs.JASS

[NOTE] Undoing operation SYMBOLIC LINK.
rm -f /etc/rc3.d/S22acct

[NOTE] Undoing operation SYMBOLIC LINK.
rm -f /etc/rc0.d/K22acct
```

項目 4 を選択すると、次の出力が表示されます。

コード例 4-6　元に戻す処理で「常にバックアップ」オプションを選択した場合の出力例

```
Enter 1, 2, 3, 4, 5, or 6: 4
[NOTE] Always do BACKUP selected.  Overriding JASS_UNDO_TYPE with
BACKUP.

[WARN] Creating backup copies of some files may cause unintended
effects.
[WARN] This is particularly true of /etc/hostname.[interface]
files as well as crontab files in /var/spool/cron/crontabs.

[NOTE] BACKUP specified, creating backup copy of
/var/spool/cron/crontabs/adm.
[NOTE] File to be backed up is from an undo operation.
[NOTE] Copying /var/spool/cron/crontabs/adm to
/var/spool/cron/crontabs/adm.BACKUP.JASS.20050715152126
cp -p /var/spool/cron/crontabs.JASS/adm.JASS.20050715151953
/var/spool/cron/crontabs/adm
rm -f /var/spool/cron/crontabs.JASS/adm.JASS.20050715151953

[NOTE] Undoing operation COPY.
[WARN] Checksum of current file does not match the saved value.
[WARN]    filename = /var/spool/cron/crontabs/root
[WARN]    current  = 741af21a62ea7a9e7abe6ba04855aa76
[WARN]    saved    = bcf180f45c65ceff3bf61012cb2b4982
[WARN] Creating backup copies of some files may cause unintended
effects.
[WARN] This is particularly true of /etc/hostname.[interface]
files as well as crontab files in /var/spool/cron/crontabs.
[NOTE] BACKUP specified, creating backup copy of
/var/spool/cron/crontabs/root.
[NOTE] File to be backed up is from an undo operation.
[NOTE] Copying /var/spool/cron/crontabs/root to
/var/spool/cron/crontabs/root.BACKUP.JASS.20050715152127
cp -p /var/spool/cron/crontabs.JASS/root.JASS.20050715151951
/var/spool/cron/crontabs/root
rm -f /var/spool/cron/crontabs.JASS/root.JASS.20050715151951

[NOTE] Undoing operation MAKE DIRECTORY.
rmdir /var/spool/cron/crontabs.JASS

[NOTE] Undoing operation SYMBOLIC LINK.
rm -f /etc/rc3.d/S22acct

[NOTE] Undoing operation SYMBOLIC LINK.
rm -f /etc/rc0.d/K22acct
```

選択した元に戻す処理が Solaris Security Toolkit によって完了したところで、強化後に変更フラグが付いたファイルを確認するとともに、必要に応じてシステムの変更を行うことをお勧めします。ファイルの変更によってシステムが矛盾した状態になっている可能性があるため、それらの変更を手動で確認するまでは、システムを再起動しないでください。

注 – この例では、変更されたファイルは `/etc/.login.BACKUP.JASS.20050715151817` という新しい名前で保存されます。元に戻す処理が完了したら、そのファイルを `/etc/.login` と比較して、さらに調整が必要であるかどうかを判断します。

---

6. **例外を調整してから、操作を続行します。**

7. **例外を調整した後、システムを再起動します。**

   Solaris OS 構成に加えた変更を有効にするためには、システムを再起動する必要があります。

注意 – Solaris Security Toolkit を JumpStart モードで稼働させると、Solaris Security Toolkit は `root` パスワードを設定します。その後元に戻す操作を行うと、`root` パスワードはその以前の設定であるパスワードなしの状態に戻ります。これは、パスワードをまったく入力することなく誰でもルートアカウントにログインできることを意味します。元に戻す操作を行なったあとは、必ず `passwd(1)` コマンドを使用して `root` パスワードをリセットしてください。システムがこのような状態にある場合は、Solaris Security Toolkit 4.2 ソフトウェアによる警告メッセージも表示されます。

第5章

JumpStart サーバーの構成と管理

この章では、Solaris Security Toolkit ソフトウェアを使用するための JumpStart サーバーの構成と管理について説明します。JumpStart テクノロジは、ネットワークベースで Solaris OS をインストールするための Sun のメカニズムであり、インストール中に Solaris Security Toolkit ソフトウェアを実行することができます。

Solaris Security Toolkit の JumpStart モードは JumpStart テクノロジに基づいており、バージョン 2.1 以降の Solaris OS 製品で使用できます。JumpStart テクノロジでは、Solaris OS とシステムソフトウェアのインストールを完全に自動化して複雑な処理を管理するため、システムの妥当性と標準化が促進されます。JumpStart テクノロジは、システムの迅速なインストールと配備に関する要件に適合するための手段です。

JumpStart テクノロジには、特にシステムのセキュリティー上の利点があります。JumpStart テクノロジを Solaris Security Toolkit ソフトウェアで使用すると、Solaris OS の自動インストール中にシステムを安全な状態に保つことができます。これにより、システムのインストール時にシステムのセキュリティーの標準化とセキュリティー対策が行えます。JumpStart ベースのインストールを容易にし、Solaris Security Toolkit によるセキュリティー強化をサポートするモジュールが含まれている JumpStart Enterprise Toolkit (JET) は、次の Sun ソフトウェアダウンロードサイトで入手できます。

```
http://www.sun.com/download/
```

JumpStart テクノロジについての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

この章では、以下の項目を説明します。

JumpStart サーバーと環境の構成

JumpStart 環境で使用するには、/opt/SUNWjass 内 (pkg ダウンロードの場合) の Solaris Security Toolkit ソースを JumpStart サーバーの基本ディレクトリにインストールします。デフォルトのディレクトリは、JumpStart サーバー上の /jumpstart です。Solaris Security Toolkit ソースをコピーすると、JASS_HOME_DIR が JumpStart サーバーのベースディレクトリになります。

この節の説明は、JumpStart テクノロジを理解していること、および既存の JumpStart 環境があることを前提にしています。

Solaris Security Toolkit ソフトウェアと JumpStart アーキテクチャーの統合は、簡単な手順で行えます。

▼ JumpStart モード用に構成する

1. **Solaris Security Toolkit のソースを JumpStart サーバーのルートディレクトリにインストールします。**

   Solaris Security Toolkit は、次の例に示すように JASS_REPOSITORY (この場合は /jumpstart) にインストールされます。

```
# pwd
/opt/SUNWjass
# pkgadd -R /jumpstart -d . SUNWjass
```

   一般に、Solaris Security Toolkit ソフトウェアは、JumpStart サーバーの SI_CONFIG_DIR にインストールされ、このディレクトリは通常 JASS_HOME_DIR にもなります。

2. **Solaris 2.5.1 OS の sysidcfg ファイルに変更を加えると、JASS_HOME_DIR/Sysidcfg/Solaris_2.5.1 ディレクトリ内のファイルが変更されます。**

   Solaris 2.5.1 OS では、SI_CONFIG_DIR 内の sysidcfg ファイルのみサポートされ、ほかのサブディレクトリ内のこのファイルはサポートされません。したがって、このバージョンの Solaris を使用している場合は、JASS_HOME_DIR/Sysidcfg/Solaris_2.5.1 内のsysidcfg ファイルを直接使用できません。この制限事項に対処するために、Solaris Security Toolkit ソフトウェアには、JASS_HOME_DIR/Sysidcfg/Solaris_2.5.1/sysidcfg ファイルにリンクした SI_CONFIG_DIR/sysidcfg が用意されています。

3. 以下のコマンドを使用して、JASS_HOME_DIR/Drivers/user.init.SAMPLE を JASS_HOME_DIR/Drivers/user.init にコピーします。

```
# pwd
/jumpstart/opt/SUNWjass/Drivers
# cp user.init.SAMPLE user.init
```

4. JumpStart のインストール時に Solaris Security Toolkit パッケージを対象システムにインストールする場合は、user.init ファイルに定義されている JASS_PACKAGE_MOUNT ディレクトリにこのパッケージを置く必要があります。次に例を示します。

```
# cp /path/to/SUNWjass.pkg JASS_HOME_DIR/Packages
```

5. マルチホーム JumpStart サーバーで問題が発生した場合は、JASS_PACKAGE_MOUNT と JASS_PATCH_MOUNT の 2 つの項目を JASS_HOME_DIR/Patches および JASS_HOME_DIR/Packages ディレクトリへの正しいパスに変更します。

6. Solaris Security Toolkit ソフトウェアを SI_CONFIG_DIR のサブディレクトリ (SI_CONFIG_DIR/path/to/JASS など) にインストールする場合は、以下の行を user.init ファイルに追加します。

```
if [ -z "${JASS_HOME_DIR}" ]; then
       if [ "${JASS_STANDALONE}" = 0 ]; then
              JASS_HOME_DIR="${SI_CONFIG_DIR}/path/to/JASS"
       fi
fi
export JASS_HOME_DIR
```

7. Solaris Security Toolkit ドライバ (たとえば、デフォルトの secure.driver) を選択または作成します。
   - hardening.driver とconfig.driver に示されるスクリプトをすべて使用する場合は、Drivers/secure.driver のパスを rules ファイルに追加します。
   - 特定のスクリプトのみ使用する場合は、そのファイルのコピーを作成し、そのコピーを変更します。ドライバのコピーと変更の手順については、『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 4 章の「ドライバのカスタマイズ」を参照してください。

注意 – Solaris Security Toolkit ソフトウェアに含まれるオリジナルのスクリプトを変更しないでください。Solaris Security Toolkit ソフトウェアの新しいリリースに効率的に移行するには、オリジナルファイルとカスタムファイルを別々に保管します。

8. **ドライバが完了したら、`rules` ファイル内に正しい項目を作成します。**

   たとえば、以下のような項目を作成します。

```
hostname imbulu - Profiles/core.profile Drivers/secure-abc.driver
```

   Solaris Security Toolkit ソフトウェアを既存の JumpStart 環境に正常に移行するには、もう 1 つ別の変更が必要な場合もあります。

9. **Solaris Security Toolkit ソフトウェアの `sysidcfg` ファイルを使用して JumpStart クライアントのインストールを自動化する場合は、そのファイルの適合性を確認します。**

   `sysidcfg` ファイルの解析中に JumpStart サーバーでエラーが検出されると、ファイルの内容がすべて無視されます。

   この節で説明する構成手順をすべて完了したあとは、JumpStart テクノロジを使用してクライアント上に Solaris OS をインストールできるようになり、インストール処理中に OS の強化または最小化を正常に行うことができます。

JumpStart プロファイルテンプレートの使用

JumpStart プロファイルテンプレートは、JumpStart モードでのみ使用されるファイルです。プロファイルの必須項目およびオプションの内容については、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

JumpStart プロファイルテンプレートを、個々のサイトを変更する際のサンプルとして使用します。プロファイルを確認し、当該環境でどのような変更を行うかを決定します。

プロファイルのコピーを作成し、サイトに合わせて変更します。Solaris Security Toolkit ソフトウェアへのアップデートによって変更内容が上書きされることがあるため、オリジナルのプロファイルを変更しないでください。

Solaris Security Toolkit ソフトウェアには、以下の JumpStart プロファイルがあります。

- `core.profile`
- `end-user.profile`
- `developer.profile`
- `entire-distribution.profile`
- `oem.profile`

- minimal-SunFire_Domain*.profile

以下の項では、これらのプロファイルについて説明します。

core.profile

この JumpStart プロファイルは、最小の Solaris OS クラスタである SUNWCreq をインストールします。ディスクのパーティションにルートパーティションとスワップパーティションを含めるように指定されることを除き、構成の変更は行われません。

end-user.profile

この JumpStart プロファイルは、End User Solaris OS クラスタである SUNWCuser と、プロセスアカウントが適切に機能するために必要な 2 つの Solaris OS パッケージをインストールします。また、ルートパーティションとスワップパーティションのみ含めるようにディスクパーティションが定義されます。

developer.profile

この JumpStart プロファイルは、Developer Solaris OS クラスタである SUNWCprog と、プロセスアカウントが適切に機能するために必要な 2 つの Solaris OS パッケージをインストールします。core.profile 定義と同様、ほかに行われる構成定義は、Solaris OS クラスタに加え、ルートとスワップを含めるディスクパーティション化のみです。

entire-distribution.profile

この JumpStart プロファイルは、Entire Distribution Solaris OS クラスタである SUNWCall をインストールします。他のプロファイルと同様、ルートパーティションとスワップパーティションを含めるようにディスクパーティションが定義されます。

oem.profile

この JumpStart プロファイルは、OEM Solaris OS クラスタである SUNWCXall をインストールします。このクラスタは Entire Distribution クラスタのスーパーセットで、OEM から提供されるソフトウェアをインストールします。

minimal-SunFire_Domain*.profile

注 – これらのプロファイルの使用は、Solaris OS バージョン 8 または 9 が稼働しているシステム上だけにとどめてください。

次のプロファイルは、すべて Sun BluePrints OnLine 掲載記事『Minimizing Domains for Sun Fire V1280, 12K, and 15K Systems』に基づいています。以下の JumpStart プロファイルは、記事に記載されているプロファイルと同じです。

- minimal-SunFire_Domain-Apps-Solaris8.profile
- minimal-SunFire_Domain-Apps-Solaris9.profile
- minimal-SunFire_Domain-NoX-Solaris8.profile
- minimal-SunFire_Domain-NoX-Solaris9.profile
- minimal-SunFire_Domain-X-Solaris8.profile
- minimal-SunFire_Domain-X-Solaris9.profile

クライアントの追加と削除

JumpStart ソフトウェアを使用してネットワークベースのクライアントインストールができるようにサーバーを構成するには、add-client スクリプトと rm-client スクリプトを使用します。これらのスクリプトは、JASS_HOME_DIR/bin ディレクトリに置かれています。JumpStart モードは、JumpStart サーバーの rules ファイルに挿入される Solaris Security Toolkit ドライバによって制御されます。

JumpStart モードを使用するように環境が構成されていない場合は、84 ページの「JumpStart サーバーと環境の構成」を参照してください。

SPARC ベースシステムの場合、add-client コマンドを実行すると Solaris Security Toolkit に必要な JumpStart クライアントと構成情報がインストールされます。このコマンドは JumpStart サーバーから実行されます。

DHCP (Dynamic Host Configuration Protocol) クライアントを必要とする **x86/x64** システムの場合、Solaris (インストール) メディアに付属の add_install_client スクリプトを使用する必要があります。

add-client スクリプト

JumpStart サーバーからクライアントを簡単に追加するには、Solaris Security Toolkit ソフトウェアに付属するこのスクリプトを使用します。このコマンドとオプションについては以下の段落で説明しますが、基本的な JumpStart テクノロジについては説明

しません。JumpStart テクノロジについての詳細は、Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

add-client スクリプトは add_install_client コマンドのラッパーであり、次の引数を受け付けます。

add-client コマンドの構文:

```
# add-client -c client -i server -m client-class -o client-OS -s sysidcfg
```

表 5-1 に、add-client コマンドの有効な入力を示します。

表 5-1 JumpStart add-client コマンド

オプション	説明
-c *client*	JumpStart クライアントの解釈可能なホスト名です。
-h|-?	使用情報を表示します。他のオプションは指定しません。ほかにオプションを指定した場合、それらは無視されます。
-i *server*	JumpStart クライアントの JumpStart サーバーインタフェースの IP アドレスまたは解釈可能なホスト名です。値が指定されていない場合は、ローカルホストで使用可能なインタフェースのリストが表示されます。
-m *client-class*	JumpStart クライアントのマシンクラスです。この値は、uname -m コマンドの出力と同じ形式です。
-o *client-OS*	JASS_HOME_DIR/OS ディレクトリ内にある使用可能な Solaris OS のバージョンです。クライアントにインストールされます。値が指定されていない場合は、JASS_HOME_DIR/OS ディレクトリ内にある使用可能な Solaris OS バージョンのリストが表示されます。
-s *sysidcfg*	システムの識別と構成に使用する sysidcfg ファイルが格納される代替ディレクトリへのパス名 (オプション) です。デフォルトでは、この値は JASS_HOME_DIR/Sysidcfg/*Solaris_version*/ ディレクトリに設定されます (*Solaris-version* はユーザーが使用した必須引数 -o から抽出される)。オプションのパス名を指定する場合は、JASS_HOME_DIR ディレクトリの相対パス名を使用してください。sysidcfg ファイルのパスだけを指定してください。
-v	このプログラムのバージョン情報です。ほかにオプションを指定した場合、それらは無視されます。

デフォルトを使用して JumpStart クライアント eng1 を追加するには、次のように指定します。

```
# /opt/SUNWjass/bin/add-client -c eng1 -m sun4u
Selecting default operating system, Solaris_ver.
Selecting default system interface, IP_address.
cleaning up preexisting install client "eng1"
removing eng1 from bootparams
updating /etc/bootparams
```

Solaris 9 OS (12/03) と -s *sysidcfg* オプションを使用して JumpStart クライアント eng1 を JumpStart サーバー jumpserve1 に追加するには、次のように指定します。

```
# /opt/SUNWjass/bin/add-client -c eng1 -i jumpserve1 -m sun4u -o Solaris_9_2003-
12 -s Hosts/alpha
cleaning up preexisting install client "eng1"
removing eng1 from bootparams
updating /etc/bootparams
```

rm-client スクリプト

JumpStart サーバーからクライアントを簡単に削除するには、Solaris Security Toolkit ソフトウェアに付属するこのスクリプトを使用します。このコマンドとオプションについては以下の段落で説明しますが、基本的な JumpStart テクノロジについては説明しません。JumpStart テクノロジについての詳細は、『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

rm-client スクリプトは add-client と同様、rm_install_client のラッパーです。

使用例 : **rm-client** [-c] *client*

ここで、*client* は JumpStart クライアントの解釈可能なホスト名です。

表 5-2 に、rm-client コマンドの有効な入力を示します。

表 5-2 JumpStart rm-client コマンド

オプション	説明
-c *client*	JumpStart クライアントの解釈可能なホスト名です。
-h|-?	使用情報を表示します。他のオプションは指定しません。ほかにオプションを指定した場合、それらは無視されます。
-v	このプログラムのバージョン情報です。ほかにオプションを指定した場合、それらは無視されます。

JumpStart クライアント eng1 を削除するには、次の rm-client コマンドを使用します。

```
# ./rm-client -c eng1
removing eng1 from bootparams
```

第6章

システムのセキュリティーの監査

この章では、Solaris Security Toolkit ソフトウェアを使用してシステムのセキュリティーを監査 (検証) する方法について説明します。セキュリティー強化後、確立されたセキュリティープロファイルを管理するには、この章で説明する情報と手順を使用します。すでに配備済みのシステムの場合、この章で説明する情報を参考にセキュリティーを評価してから、強化することもできます。

注 – この章とマニュアルで使用する監査という用語は、システムのセキュリティー状態を定義済みのセキュリティープロファイルと比較して検証する、Solaris Security Toolkit ソフトウェアの自動プロセスを指します。このマニュアルでこの用語を使用する場合、監査を実行したあとでシステムのセキュリティーが完全に確保されていることを保証するものではありません。

この章では、以下の項目を説明します。

セキュリティーの管理

セキュリティーの管理は、定期的な確認と再検討が求められる処理です。システムのデフォルトのセキュリティー構成は時間がたつうちに他人に知られてしまいがちなため、セキュリティーを維持するには注意が必要です (セキュリティーの管理についての詳細は、36 ページの「システムのセキュリティーの維持」 を参照)。

Solaris Security Toolkit には、指定されたセキュリティープロファイルとの準拠レベルを調べることによって、システムのセキュリティー状態を自動的に監査する手段が用意されています。

注 – この手段は、`jass-execute -a` コマンドを使用するスタンドアロンモードでのみ使用でき、JumpStart インストールの実行中は使用できません。

システムのセキュリティー状態の定期的な監査を手動または自動的に行います (`cron` ジョブや `rc` スクリプトなどを使用)。たとえば、新規インストールのセキュリティーを強化した 5 日後に、Solaris Security Toolkit ソフトウェアの監査コマンド (`jass-execute -a` *driver-name*) を実行して、システムのセキュリティーがセキュリティープロファイルで定義された状態から変更されていないかどうかを調べます。

セキュリティーを監査する頻度は、セキュリティーポリシーと環境の重要度によって異なります。ユーザーによっては 1 時間ごとに監査を行う場合や、毎日行う場合、月に 1 回だけ行う場合があります。また、1 時間ごとに ミニスキャン (チェック数に制限がある) を行い、1 日に 1 回フルスキャン (可能なチェックをすべて行う) を行う場合もあります。定期的に行うほかの監査とは別に、システムを再起動するたびに、その直後にセキュリティー状態をしっかりとチェックすることをお勧めします。

配備済みシステムのセキュリティー状態を管理するには、重要なコンポーネントをすべて監査します。セキュリティー状態を定期的に監査していないと、エントロピや、適正なセキュリティー状態を無意識または意図的に変更する修正により、構成が時間とともに変動することがよくあります。定期的に確認していないと、このような変更が検出されず、適切な対策を講じることができません。その結果、システムのセキュリティーが低下し、脆弱性が高まります。

定期的な監査に加え、アップグレードやパッチのインストールなどのような重要なシステム構成変更のあとにも監査を行います。

強化前のセキュリティーの確認

配備済みシステムのセキュリティーを強化する前に、そのシステムのセキュリティー状態を確認しておくと役立つことがあります。たとえば、配備済みシステムの管理を他の担当者から引き継いだ場合、システムの状態を調べれば、セキュリティー状態を把握できます。また、必要に応じて、他のシステムと同じセキュリティープロファイルに準拠させることもできます。

セキュリティー監査のカスタマイズ

監査オプションを使用すると、システムの状態を柔軟かつ広範に評価できます。強化スクリプトと同様、監査スクリプトの処理内容もカスタマイズできます。たとえば、環境変数のカスタマイズや、フレームワーク関数とヘルパー関数のカスタマイズが行えます。また、新しいチェックの追加、監査フレームワークへの機能の追加なども行えます。

ほとんどの場合、環境に応じて監査処理をカスタマイズする際のテンプレートとして、標準および製品固有の監査スクリプトが最適です。この場合、ドライバ、終了スクリプト、環境変数、およびファイルテンプレートを使用して監査スクリプトの動作をカスタマイズします。これら変更にはほとんど手間がかからず、コードを変更する必要もありません。セキュリティー強化に関する変更はすべて、監査の実行時に Solaris Security Toolkit ソフトウェアに自動的に認識されます。

ユーザーによっては、まったく新しい独自の、またはサイト固有のドライバやスクリプトの作成が必要な場合もあります。新しいドライバやスクリプトをコーディングするときは、テンプレートとサンプルを手本にします。サイト固有のドライバ、終了スクリプト、変数、および関数は、監査オプションの使用時に自動的に Solaris Security Toolkit ソフトウェアに認識されません。たとえば、サイト固有の終了スクリプト `abcc-nj-install-foo.fin` を含むサイト固有のドライバ `abcc-nj-secure.driver` を追加する場合は、サイト固有の監査スクリプト `abcc-nj-install-foo.aud` を作成する必要があります。同様に、監査スクリプトのみで開始する場合は、対応する終了スクリプトを作成する必要があります。

場合によっては、Solaris Security Toolkit ソフトウェアに備わっていないチェックや機能の追加が必要です。このような場合、チェックや新機能を監査スクリプトに追加します (該当する終了スクリプトに変更を加えることもできる)。`user.run` ファイルを通してコードに追加や変更を加える場合は、バグや障害の原因にならないように特に慎重に行なってください。

新しいドライバ、スクリプト、変数、および関数をカスタマイズまたは作成する場合は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

たとえば、Solaris Security Toolkit ソフトウェアによってインストールされないパッチを追加する必要があるとします。この場合、標準または製品固有のテンプレートのいずれかを拡張することも、独自に作成することもできます。独自のテンプレートを作成する場合は、パッチの追加を実行する終了スクリプトを作成してから、パッチのインストールを確認する監査スクリプトを作成します。既存の終了スクリプト (`.fin`) と監査スクリプト (`.aud`) をテンプレートとして使用する場合は、一意の名前を持つ新しいファイルにこれらのスクリプトを両方ともコピーする必要があります。

セキュリティー監査の準備

この章で説明する手順とガイドラインを使用するには、セキュリティープロファイルが必要です。セキュリティープロファイルの開発および実装については、第 2 章を参照してください。

Solaris Security Toolkit には、各種のセキュリティープロファイルがドライバとして含まれています。前述のとおり、デフォルトのセキュリティープロファイルとそれらのプロファイルによって行われた変更がシステムに適さない場合もあります。一般に、実装されるセキュリティープロファイルは、「最高水準」のセキュリティーを設定します。つまり、不要なサービスを無効にするだけでなく、`secure.driver` によって無効になっている任意のセキュリティー機能を有効にします。

Solaris Security Toolkit ソフトウェアの多くのユーザーは、標準および製品固有のセキュリティープロファイルを各自の環境に適用できます。具体的には、必要なセキュリティー状態に最も近いセキュリティープロファイルを見つけ、それをシステムの評価と強化の両方に使用します。

セキュリティープロファイルテンプレートを確認し、環境に合わせてカスタマイズするか、新しいテンプレートを作成します。セキュリティープロファイルをカスタマイズするテクニックとガイドラインについては、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。こうすることで、組織の必要性に合わせて調整したセキュリティー状態を実現できるとともに、セキュリティー評価中に返される不正エラーを最小限に抑えることもできます。たとえば、Telnet を有効にする必要があれば、セキュリティー評価時に Telnet が脆弱性として見なされないようにセキュリティープロファイルをカスタマイズできます。したがって、認証と暗号化に Telnet と Kerberos を使用するサイトでは、Telnet を使用しても脆弱性とは見なされません。

オプションの使用と監査出力の制御

この節では、監査の実行時に使用できるオプションと出力制御用のオプションについて説明します。この節では、以下の項目を説明します。

コマンド行オプション

次のコマンド行形式は、セキュリティープロファイルに照らしてシステムの監査を行う方法を示しています。

```
# jass-execute -a driver [ -V [0-4]] [ -q | -o output_file ] [ -m e-mail_address ]
```

Solaris Security Toolkit ソフトウェアの監査コマンドを実行するときは、表 6-1 に示すオプションを使用できます。

表 6-1　監査コマンドで使用するコマンド行オプション

オプション	説明
-a *driver*	システムがセキュリティープロファイルに適合しているかどうかを判断します。
-m *e-mail_address*	社内サポートに使用する電子メールアドレスを指定します。
-o *output_file*	Solaris Security Toolkit の実行出力に使用するファイルの名前を指定します。
-q	非出力モードを指定します。このコマンドの実行中、メッセージは表示されません。出力は JASS_REPOSITORY/ に格納されます。
-V *verbosity_level*	監査の詳細レベル (0 ~ 4) を指定します。

jass-execute -a　コマンドで使用できるオプションについての詳細は、以下の節を参照してください。

- 97 ページの「ヘルプ表示オプション」
- 98 ページの「電子メール通知オプション」
- 99 ページの「出力ファイルオプション」
- 99 ページの「非出力オプション」
- 99 ページの「詳細オプション」

ヘルプ表示オプション

-h オプションを使用すると、使用可能なオプションの概要を説明する jass-execute ヘルプメッセージが表示されます。

-h オプションでは次のような出力が生成されます。

コード例 6-1　　-h オプションの出力例

```
# ./jass-execute -h

To apply this Toolkit to a system, using the syntax:
  jass-execute [-r root_directory -p os_version ]
  [ -q | -o output_file ] [ -m e-mail_address ]
  [ -V [3|4] ] [ -d ] driver

To undo a previous application of the Toolkit from a system:
  jass-execute -u [ -b | -f | -k ] [ -q | -o output_file ]
     [ -m e-mail_address ] [ -V [3|4] ]

To audit a system against a pre-defined profile:
  jass-execute -a driver [ -V [0-4] ] [ -q | -o output_file ]
     [ -m e-mail_address ]

To display the history of Toolkit applications on a system:
  jass-execute -H

To display the last application of the Toolkit on a system:
  jass-execute -l

To display this help message:
  jass-execute -h
  jass-execute -?

To display version information for this program:
  jass-execute -v
```

電子メール通知オプション

-m *email-address* オプションを使用すると、実行の完了時に Solaris Security Toolkit ソフトウェアによって出力が電子メールで自動的に送信されます。他のオプションを使用すると、ログに加えて電子メールレポートも生成されます。

sunfire_15k_sc-config.driver を呼び出し、電子メール通知オプションを使用して Solaris Security Toolkit ソフトウェアを実行する場合、次のように指定します。

```
# ./jass-execute -m root -a sunfire_15k_sc-config.driver
[...]
```

出力ファイルオプション

-o *output-file* オプションは、jass-execute のコンソール出力を別ファイル *output-file* に転送します。

このオプションは、JASS_REPOSITORY ディレクトリで維持されるログには影響しません。Solaris Security Toolkit は大量の出力を生成するので、このオプションは、低速の端末接続環境で実行する場合に特に便利です。

このオプションは、-d、-u、または -a オプションとともに使用できます。

-o オプションを使用すると、以下のような出力が生成されます。

コード例 6-2　　-o オプションの出力例

```
# ./jass-execute -o jass-output.txt -a secure.driver
[NOTE] Executing driver, secure.driver
[NOTE] Recording output to jass-output.txt
#
```

非出力オプション

-q オプションを使用すると、セキュリティー強化の実行時に Solaris Security Toolkit から標準入出力 (stdio) ストリームへの出力が行われなくなります。

このオプションは、JASS_REPOSITORY ディレクトリで維持されるログには影響しません。-o オプションと同様、このオプションも、Solaris Security Toolkit ソフトウェアを cron ジョブを利用して実行する場合や、低速のネットワーク接続環境で実行する場合に特に便利です。

このオプションは、-d、-u、または -a オプションとともに使用できます。

-q オプションを使用すると、以下のような出力が生成されます。

コード例 6-3　　-q オプションの出力例

```
# ./jass-execute -q -a secure.driver
[NOTE] Executing driver, secure.driver
```

詳細オプション

-V オプションは、監査の詳細レベルを指定します。このオプションは監査でのみ使用できます。詳細レベルを使用すると、監査結果の表示方法を柔軟に制御できます。たとえば、監査対象のマシンが 100 台ある場合、出力をマシンごとに 1 行ずつ表示す

るように制限すれば、合格したマシンと不合格のマシンを簡単に判別できます。その後、不合格のマシンに対して、拡張出力を生成する監査を実行すれば、問題箇所を絞り込めます。

-V オプションによって 5 つの詳細レベル (0 ～ 4) を制御します。レベルを上げるごとに情報が詳細になり、これらの情報を基に合格したチェックと不合格のチェックをより詳しく理解できます。表 6-2 に詳細レベルを示します。

表 6-2 監査の詳細レベル

レベル	出力
0	合格または不合格を示す単一行が表示されます。
1	スクリプトごとに合格または不合格を示す行が 1 行表示されるほか、スクリプト行がすべて表示されたそのあとに総合計スコアを示す行が 1 行表示されます。
2	スクリプトごとに、すべてのチェック結果が表示されます。
3	バナーとヘッダーメッセージを示す全出力が複数行で表示されます。これはデフォルト設定です。
4	複数行 (レベル 3 で表示される全データ) に加え、logDebug ログ関数によって生成された項目がすべて表示されます。これはデバッグ用のレベルです。

注 – jass-execute -V コマンドのデフォルトの詳細レベルは 3 です。

監査の詳細レベルについての詳しい説明は、jass-execute のマニュアルページ、または『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 7 章の「JASS_VERBOSITY」を参照してください。

バナーおよびメッセージ出力

Solaris Security Toolkit の監査オプションは、バナーとメッセージをレポートまたは無視するように構成できます。JASS_LOG_BANNER 変数は、詳細レベル 0 ～ 2 では使用できません。これらの出力オプションは、詳細レベル 3 および 4 に適用されます。たとえば、出力から合格メッセージ (JASS_LOG_SUCCESS 変数) を削除すれば、不合格メッセージ (JASS_LOG_FAILURE 変数) のみをレポートして、それだけを処理できます。

表 6-3 に、ログ変数によって制御できるバナーとメッセージを示します。ログ変数についての詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 7 章の「JASS_LOG_BANNER」を参照してください。詳細レベルについての詳しい説明は、jass-execute のマニュアルページ、または『Solaris Security Toolkit 4.2 リファレンスマニュアル』、第 7 章の「JASS_VERBOSITY」を参照してください。ロ

グ変数に 0 が設定されていると、そのメッセージの種類では出力が生成されません。逆に、ログ変数が 1 に設定されている場合は、メッセージが表示されます。これらの各変数は、デフォルトでは出力を表示します。表 6-3 にログ変数を示します。

表 6-3 監査出力へのバナーとメッセージの表示

ログ変数	ログ接頭辞	説明
JASS_LOG_BANNER	すべてのバナー出力	このパラメタは、バナーメッセージの表示を制御します。これらのメッセージは、通常は等号 (=) またはハイフン (-) 文字のいずれかを含む区切り文字で囲まれています。
JASS_LOG_ERROR	[ERR]	このパラメタは、エラーメッセージの表示を制御します。0 に設定されている場合、エラーメッセージは生成されません。
JASS_LOG_FAILURE	[FAIL]	このパラメタは、失敗メッセージの表示を制御します。0 に設定されている場合、失敗メッセージは生成されません。
JASS_LOG_NOTICE	[NOTE]	このパラメタは、通知メッセージの表示を制御します。0 に設定されている場合、通知メッセージは生成されません。
JASS_LOG_SUCCESS	[PASS]	このパラメタは、成功または合格状態メッセージの表示を制御します。0 に設定されている場合、成功メッセージは生成されません。
JASS_LOG_SUMMARY	[SUMMARY]	このパラメタは、概要メッセージの表示を制御します。0 に設定されている場合、概要メッセージは表示されません。
JASS_LOG_WARNING	[WARN]	このパラメタは、警告メッセージの表示を制御します。0 に設定されている場合、警告メッセージは生成されません。

これらのオプションは、特定のメッセージのみ表示する必要がある場合に便利です。これらのオプションを設定すれば、出力を最小限に抑えながら、重要な箇所に焦点を絞ることができます。たとえば、JASS_LOG_FAILURE (デフォルト設定の 1 のまま) を除くすべてのログ変数に 0 を設定すると、監査結果として logFailure 関数によって生成された失敗のみレポートされます。

コード例 6-4 監査の失敗のみをレポートする場合の出力例

```
# JASS_LOG_FAILURE=1
# export JASS_LOG_FAILURE
# JASS_LOG_BANNER=0
# JASS_LOG_ERROR=0
# JASS_LOG_NOTICE=0
# JASS_LOG_SUCCESS=0
# JASS_LOG_SUMMARY=0
```

コード例 6-4　　監査の失敗のみをレポートする場合の出力例 (続き)

```
# JASS_LOG_WARNING=0
# export JASS_LOG_BANNER JASS_LOG_ERROR
# export JASS_LOG_NOTICE JASS_LOG_SUCCESS
# export JASS_LOG_WARNING
# bin/jass-execute -a abc.driver -V2
update-at-deny      [FAIL] User adm is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User zz9999999 is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User gdm is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User lp is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User nobody4 is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User root is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User smmsp is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User sys is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User uucp is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] User webservd is not listed in /etc/cron.d/at.deny.
update-at-deny      [FAIL] Script Total: 10 Errors
update-inetd-conf   [FAIL] Service svc:/network/telnet:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/ftp:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/finger:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/login:rlogin was enabled.
update-inetd-conf   [FAIL] Service svc:/network/shell:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/login:eklogin was enabled.
update-inetd-conf   [FAIL] Service svc:/network/login:klogin was enabled.
update-inetd-conf   [FAIL] Service svc:/network/shell:kshell was enabled.
update-inetd-conf   [FAIL] Service svc:/application/font/stfsloader:default was
enabled.
update-inetd-conf   [FAIL] Service svc:/network/security/ktkt_warn:default was
enabled.
update-inetd-conf   [FAIL] Service svc:/network/rpc/smserver:default was
enabled.
update-inetd-conf   [FAIL] Service svc:/network/rpc/rstat:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/rpc/rusers:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/nfs/rquota:default was enabled.
update-inetd-conf   [FAIL] Service svc:/network/rpc/gss:default was enabled.
update-inetd-conf   [FAIL] Service 100235 is enabled in /etc/inet/inetd.conf.
update-inetd-conf   [FAIL] Service 100083 is enabled in /etc/inet/inetd.conf.
update-inetd-conf   [FAIL] Service 100068 is enabled in /etc/inet/inetd.conf.
update-inetd-conf   [FAIL] Script Total: 18 Errors
abc.driver          [FAIL] Driver Total: 28 Errors
abc.driver          [FAIL] Grand Total: 28 Errors
#
```

ホスト名、スクリプト名、タイムスタンプの出力

Solaris Security Toolkit の監査オプションは、詳細レベル 0 ～ 2 でホスト名、スクリプト名、およびタイムスタンプ情報も出力するように構成できます。たとえば、監査対象のマシンが多数ある場合、ホスト名、スクリプト名、またはタイムスタンプで出力をソートできます。表 6-4 に変数を示します。

表 6-4　ホスト名、スクリプト名、およびタイムスタンプ監査出力の表示

変数名	変数の説明
JASS_DISPLAY_HOSTNAME	このパラメタを 1 に設定すると、各ログ項目の先頭にシステムのホスト名が付加されます。ホスト名は JASS_HOSTNAME パラメタに従います。このパラメタにはデフォルトでは値が設定されていないため、ホスト名は表示されません。
JASS_DISPLAY_SCRIPTNAME	このパラメタはデフォルトでは 1 に設定されているため、各ログ項目の先頭に現在実行されている監査スクリプトの名前が付加されます。このパラメタに 1 以外の値を設定すると、監査スクリプトの名前が表示されなくなります。
JASS_DISPLAY_TIMESTAMP	このパラメタを 1 に設定すると、各ログ項目の先頭に監査実行時のタイムスタンプが付加されます。タイムスタンプは JASS_TIMESTAMP パラメタに従います。このパラメタにはデフォルトでは値が設定されていないため、タイムスタンプは表示されません。

ホスト、スクリプト、およびタイムスタンプ情報を付加するように Solaris Security Toolkit ソフトウェアを構成することで、単一のシステムまたは複数のシステムからの多くの実行結果を連結し、キーとなるデータを基にそれらをソートできます。これらの情報から、複数のシステムにかかわる問題や、配備プロセスの問題を探すことができます。たとえば、情報をこのように利用すれば、管理者は、特定のプロセスによって構築されたシステムで常に同じチェックが失敗するかどうかを調べることができます。

JASS_DISPLAY_TIMESTAMP パラメタを 1 に設定し、JASS_DISPLAY_SCRIPTNAME の値を 0 に設定すると、次のような出力が生成されます。

コード例 6-5　ログ項目の監査の出力例

```
# JASS_DISPLAY_TIMESTAMP=1
# JASS_DISPLAY_SCRIPTNAME=0
# export JASS_DISPLAY_TIMESTAMP JASS_DISPLAY_SCRIPTNAME
# bin/jass-execute -a abc.driver -V2
20050716132908 [FAIL] User adm is not listed in /etc/cron.d/at.deny.
20050716132908 [PASS] User bin is listed in /etc/cron.d/at.deny.
20050716132908 [FAIL] User zz999999 is not listed in /etc/cron.d/at.deny.
...
...
```

コード例 6-5　　ログ項目の監査の出力例 (続き)

```
...
20050716132908 [FAIL] Script Total: 18 Errors
20050716132908 [FAIL] Driver Total: 28 Errors
20050716132908 [FAIL] Grand Total: 28 Errors
20050716132908 [SUMMARY] Results Summary for AUDIT run of dan.driver
20050716132908 [SUMMARY] The run completed with a total of 2 scripts run.
20050716132908 [SUMMARY] There were  Failures in   2 Scripts
20050716132908 [SUMMARY] There were  Errors   in   0 Scripts
20050716132908 [SUMMARY] There were  Warnings in   0 Scripts
20050716132908 [SUMMARY] There was a Note     in   1 Script
20050716132908 [SUMMARY] Failure Scripts listed in:
/var/opt/SUNWjass/run/20050716132908/jass-script-failures.txt
20050716132908 [SUMMARY] Notes Scripts listed in:
/var/opt/SUNWjass/run/20050716132908/jass-script-notes.txt
#
```

セキュリティー監査の実行

システムの定期的なセキュリティー評価は、実装されたセキュリティープロファイルにセキュリティーがどの程度準拠しているかを示すベンチマークになります。セキュリティー評価を実行する最も一般的な例として、新規インストールのセキュリティー強化後に行うセキュリティー保守があります。セキュリティー評価オプションは、システムのセキュリティー強化に使用したものと同じ強化ドライバを実行すれば済むように設計されていましたが、現在では、セキュリティー強化中に実装されたセキュリティープロファイルと比較することで現在の状態をチェックする -a オプションを使用します。これによって複雑さが軽減するとともに、柔軟性が得られます。たとえば、セキュリティープロファイルをアップデートすると、以降のセキュリティー評価は、アップデート後のセキュリティープロファイルを使用して行われます。

次に、すでに配備みのシステムのセキュリティー管理を担当する場合を考えてみます。システムのセキュリティーを強化する前に、セキュリティー評価を実行するとします。このような場合、独自のセキュリティープロファイルを定義した上で、Solaris Security Toolkit のセキュリティープロファイルテンプレートをカスタマイズするか、セキュリティープロファイルテンプレートをそのまま使用します。

▼ セキュリティー監査を実行する

監査を実行する前に、セキュリティープロファイルを定義または選択する必要があります。詳細については、96 ページの「セキュリティー監査の準備」を参照してください。

注意 – まだセキュリティーが強化されていない配備済みシステムにセキュリティー評価を実行する場合は、最初にマシンをバックアップして再起動し、マシンの構成が既知の作業状態にあり、整合性が保たれていることを確認してください。この再起動時に検出されたエラーや警告を修正またはメモしてから、セキュリティー評価を実行します。

1. **使用するセキュリティープロファイル (強化ドライバ) を選択します。**
   - すでにシステムのセキュリティーを強化済みの場合は、同じセキュリティープロファイルを使用します。

     たとえば、`secure.driver` を使用します。
   - まだシステムのセキュリティーを強化していない場合は、標準または独自のセキュリティープロファイルを使用します。

     たとえば、`secure.driver` または `abccorp-secure.driver` を使用します。

   標準ドライバおよび製品固有ドライバの最新の全一覧と情報については、`security_drivers` のマニュアルページ、または『Solaris Security Toolkit 4.2 リファレンスマニュアル』の第 4 章を参照してください。
2. **使用するコマンド行オプションと出力の制御方法を決定します。**

   96 ページの「オプションの使用と監査出力の制御」を参照してください。
3. **`jass-execute -a` コマンド、セキュリティープロファイル名、および使用するオプションを入力します。**

   次に、`abc-secure.driver` を使用して監査を実行した例を示します。

コード例 6-6 監査の出力例

```
# ./jass-execute -a abc-secure.driver
[NOTE] Executing driver, abc-secure.driver

[...]

================================================================
abc-secure.driver: Audit script: enable-rfc1948.aud
================================================================

#----------------------------------------------------------------
# RFC 1948 Sequence Number Generation
#
# Rationale for Verification Check:
```

コード例 6-6　　監査の出力例 (続き)

```
#
# The purpose of this script is to verify that the system is
# configured and is in fact using RFC 1948 for its TCP sequence
# number generation algorithm (unique-per-connection ID). This is
# configured by setting the 'TCP_STRONG_ISS' parameter to '2' in
# the /etc/default/inetinit file.
#
# Determination of Compliance:
#
[...]
#----------------------------------------------------------------

[PASS] TCP_STRONG_ISS is set to '2' in /etc/default/inetinit.
[PASS] System is running with tcp_strong_iss=2.

# The following is the vulnerability total for this audit script.

[PASS] Audit Check Total : 0 Error(s)

================================================================

# The following is the vulnerability total for this driver profile.

[PASS] Driver Total : 0 Error(s)

================================================================
abc-secure.driver: Driver finished.
================================================================

# The following is the vulnerability grand total for this run.

[PASS] Grand Total : 0 Error(s)
```

監査が開始されると、Solaris Security Toolkit ソフトウェアは JASS_HOME_DIR/Audit ディレクトリからファイルにアクセスします。JASS_HOME_DIR/Audit と JASS_HOME_DIR/Finish の両ディレクトリ内のファイルは、同じベースファイル名を共有していますが、ファイル名の接尾辞が異なります。driver.run スクリプトは、JASS_SCRIPTS 変数で定義されている終了スクリプトの接尾辞を .fin から .aud に変更することによってそれらのスクリプトを自動的に監査スクリプトに変換するとともに、実行中に各警告レベルで生成されるメッセージをすべて含むファイルを示します。

監査が開始され、Solaris Security Toolkit ソフトウェアの状態が初期化されます。実行中にアクセスされる各ドライバによって、そのファイルテンプレートと監査スクリプトの状態がすべて評価されます。各チェックの結果、脆弱性を示すゼロまたはゼロ以外の値によって成功または失敗が示されます。通常、失敗は 1 で表されます。実行された各スクリプトは、スクリプトに含まれる各チェックの脆弱性値の合計に基づいて、セキュリティースコアの合計を算出します。各ドライバの総合的な脆弱性値は、ドライバによる評価が完了したところで表示されます。全スコアの総合計が実行終了時に表示されます。

セキュリティー評価オプションは、評価開始時にシステムの状態を全体的に確認する手段です。Solaris Security Toolkit ソフトウェアは、構成ファイルを調べることにより、保存されているシステム状態をチェックするとともに、プロセステーブル情報やデバイスドライバ情報などを調べることにより、システムの実行状態をチェックします。Solaris Security Toolkit ソフトウェアは、各ファイルまたはサービスの有無をチェックし、サービスに関連付けられたソフトウェアがインストールされているか、構成されているか、有効になっているか、および実行されているかどうかをチェックします。このようにして、現在のシステム状態に関する正確なスナップショットが得られます。

第7章

システムのセキュリティーの確保

この章では、これまでに説明してきた内容と専門知識を、新しい Solaris 8 または 9 OS のインストールとセキュリティー確保に実際に適用する方法について説明します。この章では、Solaris Security Toolkit ソフトウェアを Solaris 8 OS 対応の Check PointFirewall-1 NG とともに配備する方法を具体的に示します。

この章の内容は、新しいシステムとアプリケーションのセキュリティーを確保する際の指針 (ケースシナリオ) として利用してください。

Sun BluePrint マニュアルとオンライン掲載記事には、Sun の多くのシステムの最小化および強化プロセスが詳しく記載されています。製品別の最新のマニュアルと記事については、次の Web サイトを参照してください。

```
http://www.sun.com/blueprints
```

この章では、以下の項目を説明します。

計画と準備

このケーススタディでの説明に従って、最小化され、セキュリティーで保護されたシステムを効果的かつ効率的に配備するには、計画と準備が重要です。基本となるネットワークインフラ、ポリシー、および手順を適切に構築する必要があります。また、システムのサポートと保守が定義され、伝達済みである必要もあります。計画と準備

についての詳細は、第 2 章を参照してください。この章で説明するシナリオでは、ファイアウォールシステム用に最小化および強化された Solaris OS イメージを実現するためにシステム管理者 (SA) が行う処理と作業が記載されています。

このシナリオでは、システム管理者は、顧客にファイアウォールサービスを提供するサービスプロバイダ (xSP) 向けの Check PointFirewall-1 NG システムを構築および配備する、自動化されたスケーラブルなソリューションを作成します。xSP の要件と検討事項は、次のとおりです。

- xSP では、このような多くのシステムを配備する予定であるため、各システムの構築と配備に要する時間が重要で、作業の効率化が求められます。
- システムは、各システムの内部 Ethernet インタフェースに接続された専用の管理ネットワークを使用してインストールされます。このネットワークは xSP のスタッフ専用で、加入者は使用しません。
- その他すべてのインタフェースは個別の物理ネットワークインタフェース上にあり、フィルタが設定されます。
- 管理ネットワークのセキュリティーは、配備するファイアウォールシステムのセキュリティー全体にとって重要です。

以上の要件に基づいて、システム管理者は JumpStart テクノロジと Solaris Security Toolkit ソフトウェアを利用して、OS イメージのインストール、最小化、および強化を自動化します。

前提条件と制限事項

この章では、Solaris Security Toolkit ソフトウェアと JumpStart テクノロジによるインストールを使用していることを前提とします。ソフトウェアのインストールの方法とガイドラインは、第 3 章に示されています。

この章では、特定のアプリケーションを最小化および強化するカスタム構成を開発することも前提とします。Solaris Security Toolkit ソフトウェアには、そのアプリケーション専用のドライバや JumpStart プロファイルがありません。したがって、アプリケーション用のカスタムドライバとプロファイルを作成する必要があります。そのためには、既存のドライバとプロファイルをコピーし、アプリケーションに適合するように変更します。

このケースシナリオでは、次のスキルレベルがシステム管理者に求められます。

- OS とアプリケーションの構成に関する豊富な知識と経験。
- 構成のテスト方法と調整方法に関する知識。
- クライアントシステムのインストール元となる JumpStart 環境の構築方法に関する知識。Sun BluePrint マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照。

- OS の最小化手法に精通していること。『Enterprise Security: Solaris Operating Environment Security Journal, Solaris Operating Environment Versions 2.5.1, 2.6, 7, and 8』を参照。
- Solaris Security Toolkit ソフトウェアの基本事項に精通しており、かつ最小化および強化テクニックとガイドラインを使用してカスタム構成を構築する準備が整っていること。第 1 章を参照。

システム環境

このシナリオは、以下のハードウェアおよびソフトウェア環境に基づきます。

- Check PointFirewall-1 NG
- Solaris 8 OS
- JumpStart テクノロジ
- Solaris OS クラスタ (`SUNWCreq`)
- Solaris Security Toolkit ソフトウェア
- SPARC テクノロジに基づくプラットフォーム
- 少なくとも 2 つの Ethernet インタフェース

セキュリティー要件

このシナリオでは、高度な要件とソフトウェアパッケージが示されていますが、すべてのパッケージについて固有のコンポーネントとサービスを特定する必要があります。また、システムの運用管理に必要な Solaris OS の機能も特定する必要があります。

以下に、ソフトウェアコンポーネントの使用方法を詳細に示します。

- 遠隔管理用の Secure Shell
- リモートファイルをダウンロードする FTP クライアント
- ファイルをコピーする `scp` と `sftp`

このリストを基に、セキュリティープロファイルを作成できます。セキュリティープロファイルの作成およびプロファイルテンプレートの使用についての詳細は、32 ページの「Solaris Security Toolkit プロファイルの開発および実装」を参照してください。

セキュリティープロファイルの作成

セキュリティープロファイルは、システムのセキュリティー構成を強化および最小化する際に Solaris Security Toolkit ソフトウェアが実行するセキュリティーの変更内容を定義します。Solaris Security Toolkit ソフトウェアに付属する標準のセキュリティープロファイルやドライバは、最小化された Check PointFirewall-1 NG システムの要件を満たしません。したがって、必要なシステム変更を実装するカスタムセキュリティープロファイルを作成する必要があります。

このシナリオの場合、セキュリティープロファイルの作成プロセスについては、この章のいくつかの節で説明されています。最初に、既存のドライバを基に新しいドライバファイルを作成します。次に、新しいドライバを変更して、前述したセキュリティー要件に準拠させます。最小化については、112 ページの「ソフトウェアのインストール」を参照し、変更の強化については、120 ページの「強化構成のカスタマイズ」を参照してください。

ソフトウェアのインストール

この節では、ソフトウェアのインストール処理を具体的に示します。サンプルシナリオには、例外事項やシナリオ特有の指示があります。ソフトウェアの一般的なインストール手順については、第 3 章を参照してください。

注 – 次の手順は、関連する状況においても手本として使用できます。

この節では、以下の作業を説明します。

- 112 ページの「セキュリティーソフトウェアのダウンロードとインストール」
- 113 ページの「パッチのインストール」
- 114 ページの「OS クラスタの指定とインストール」

セキュリティーソフトウェアのダウンロードとインストール

以下の操作を行なって、Solaris Security Toolkit とパッチなどのセキュリティーソフトウェアをダウンロードして、JumpStart サーバーにインストールします。

▼ セキュリティーソフトウェアをダウンロードしてインストールする

1. **Solaris Security Toolkit ソフトウェアと追加のセキュリティーソフトウェアをダウンロードします。**

   44 ページの「セキュリティーソフトウェアのダウンロード」を参照してください。

2. **ダウンロードした Solaris Security Toolkit ソフトウェアと追加のセキュリティーソフトウェアをインストールします。**

   53 ページの「ソフトウェアのインストールと実行」を参照してください。

注意 – Solaris Security Toolkit ソフトウェアはまだ実行しないでください。最初に、以下の節で説明する追加構成とカスタマイズを実行します。

パッチのインストール

OS パッチは、セキュリティーの脆弱性、可用性に関する問題、パフォーマンス上の問題、およびシステムのその他の問題を処理できます。新しい OS のインストール時、およびインストール後も継続的に、必要なパッチがインストールされていることを確認します。

Solaris Security Toolkit ソフトウェアは、SunSolve Online から入手可能な推奨およびセキュリティパッチクラスタをインストールするメカニズムを提供します。この OS 固有のパッチクラスタには、最も一般的に必要となるパッチが含まれています。

▼ パッチをインストールする

1. **最低限、推奨およびセキュリティパッチクラスタを `Patches` ディレクトリにダウンロードして圧縮解除します。**

   強化ドライバに `install-recommended-patches.fin` スクリプトが含まれている場合は、パッチクラスタが自動的にインストールされます。

   Check PointFirewall-1 NG にはほかにも問題があります。推奨およびセキュリティパッチクラスタに含まれていない専用のパッチが必要です。Check PointFirewall-1 NG には以下のパッチが必要です。

   - 108434
   - 108435

2. **パッチ 108434 と 108435 のインストールを自動化するには、SunSolve OnLine から最新バージョンをダウンロードし、それを `Patches` ディレクトリに置きます。**

3. 各パッチの名前を指定して add_patch ヘルパー関数を呼び出す終了スクリプト (fw1-patch-install.fin など) を作成します。

この終了スクリプトは、Check PointFirewall-1 NG に必要な 2 つのパッチ ID を指定して正しいヘルパー関数を呼び出します。次に例を示します。

```
# !/bin/sh

# add_patch 108434-10

# add_patch 108435-10
```

OS クラスタの指定とインストール

OS をインストールするためのディスクレイアウトを定義したら、次は、インストールする Solaris OS クラスタを指定します。Solaris OS で使用できるインストールクラスタには、SUNWCreq、SUNWCuser、SUNWCprog、SUNWCall、SUNWCXall の 5 つがあり、そのいずれかを選択します。

▼ OS クラスタを指定してインストールする

1. インストールする OS クラスタを指定します。

このケースシナリオの目的は最小化した専用ファイアウォールの構築であるため、使用可能な Solaris OS クラスタの中で最小の SUNWCreq を指定します。このパッケージは Core とも呼ばれます。

このクラスタには比較的少数のパッケージしか含まれていないため、他のパッケージも必要です。これら必要なほかのパッケージを Solaris OS クラスタ定義によってプロファイルに含める必要があります。

ベースラインプロファイル定義により、定義済みプロファイルに以下の項目を追加します。

```
cluster          SUNWCreq
```

SUNWCreq インストールクラスタには、ファイアウォール Sun サーバーの適切な動作には不要なパッケージが含まれています。作業ベースラインを用意できたら余分なパッケージを削除します。Sun BluePrints OnLine 掲載記事『Minimizing the Solaris Operating Environment for Security: Updated for the Solaris 9 Operating Environment』を参照してください。

2. **ユーザー自身で定義したセキュリティープロファイルでインストールを実行し、パッケージの依存関係がないかどうかを確認します。**

   パッケージの依存関係によってはインストール中に検出される場合もあります。Check PointFirewall-1 NG には、次の Solaris OS パッケージが必要です。

   - `SUNWter` – 端末情報
   - `SUNWadmc` – システム管理コアライブラリ
   - `SUNWadmfw` – システムおよびネットワーク管理フレームワーク
   - `SUNWlibC` および `SUNWlibCx` – Check PointNG アプリケーションに必要

   プロファイル内のパッケージの完全なリストは、以下のとおりです。

```
cluster          SUNWCreq
package          SUNWter          add
package          SUNWlibC         add
package          SUNWlibCx        add
package          SUNWadmc         add
package          SUNWadmfw        add
```

注 – このリストは、このケーススタディには適切ですが、この構成を配備する実際の環境によってはほかのパッケージの追加または削除を行うことができます。

126 ページの「品質保証のテスト」で説明するように、機能とセキュリティーの両面からシステムが検証されるまでは、パッケージの最終的なリストの変更が必要な場合もあります。そのような場合は、プロファイルを修正してシステムを再インストールし、テストを繰り返します。

3. **前の 2 つの手順でのパッケージの依存関係に基づいて、`minimize-firewall.fin script` を作成します。**

JumpStart サーバーおよびクライアントの構成

この節では、JumpStart サーバーとクライアントを構成して、カスタムセキュリティープロファイルを使用して最小化を行う方法について具体的に説明します。JumpStart 環境での Solaris Security Toolkit ソフトウェアの使用についての詳細は、第 5 章を参照してください。

この節では、以下の作業を説明します。

- 116 ページの「インフラストラクチャーの準備」
- 118 ページの「rules ファイルの検証とチェック」

インフラストラクチャーの準備

以下の作業を行なって、インフラストラクチャーを準備します。以下の作業では、既存のドライバ、プロファイル、および終了スクリプトを使用してクライアントのベースライン構成を作成するプロセスを具体的に示します。このベースラインを適切に準備できたら、適切に機能することを確認し、対象アプリケーションに合わせてカスタマイズします。

▼ インフラストラクチャーを準備する

1. JumpStart サーバーと環境を構成します。

   詳細な手順については、第 5 章を参照してください。

2. `add-client` コマンドを使用して、クライアントを JumpStart サーバーに追加します。

   コード例 7-1　　JumpStart サーバーへのクライアントの追加

```
# pwd
/jumpstart
# bin/add-client -c jordan -o Solaris_8_2002-02 -m sun4u
-s nomex-jumpstart
cleaning up preexisting install client "jordan"
removing jordan from bootparams
updating /etc/bootparams
```

3. クライアントの `rules` ファイル項目を作成し、正しい JumpStart プロファイルと終了スクリプトを指定します。次に例を示します。

```
hostname jordan - Profiles/xsp-minimal-firewall.profile \
  Drivers/xsp-firewall-secure.driver
```

4. Solaris Security Toolkit ソフトウェアに付属するファイルをコピーして、`xsp-minimal-firewall.profile` という名前のプロファイルファイルと `xsp-firewall-secure.driver` という名前のドライバファイルを作成します。

   これらのファイルを作成しないと、次の手順を正しく完了できません。最初は、Solaris Security Toolkit ソフトウェアとともに配布されたファイルのコピーを使用できます。

注 – Solaris Security Toolkit ソフトウェアのオリジナルファイルは変更しないでください。

以下にファイルの作成方法を示します。

コード例 7-2　　プロファイルの作成

```
# pwd
/jumpstart/Drivers
# cp install-Sun_ONE-WS.driver xsp-firewall-secure.driver
# cp hardening.driver xsp-firewall-hardening.driver
[...]
# pwd
/jumpstart/Profiles
# cp minimal-Sun_ONE-WS-Solaris8-64bit.profile \
     xsp-minimal-firewall.profile
```

この例では、専用ファイアウォールの開発に適したベースラインという理由から、専用の Web サーバー構成を利用しています。

5. **プロファイルとドライバファイルを作成したら、これらのファイルを以下のように変更します。**

   a. `hardening.driver` **への** `xsp-firewall-secure.driver` **参照を** `xsp-firewall-hardening.driver` **に置き換えます。**

   b. `JASS_SCRIPTS` **に定義されている 2 つの終了スクリプトを** `minimize-firewall.fin` **と終了スクリプト (たとえば、**`fw1-patch-install.fin`**) への参照に置き換えます。**

   変更後のスクリプトは、以下のようになります。

コード例 7-3　　変更後のスクリプトの出力例

```
DIR="`/bin/dirname $0`"
export DIR
. ${DIR}/driver.init
. ${DIR}/config.driver
JASS_SCRIPTS="
                minimize-firewall.fin
                fw1-patch-install.fin"
. ${DIR}/driver.run
. ${DIR}/xsp-firewall-hardening.driver
```

6. 以下のコマンドを使用して、rules ファイル項目が正しいことを確認します。

コード例 7-4　　rules ファイルが適正であることの確認

```
# pwd
/jumpstart
# ./check
Validating rules...
Validating profile Profiles/end-user.profile...
Validating profile Profiles/xsp-minimal-firewall.profile...
Validating profile Profiles/test.profile...
Validating profile Profiles/entire-distribution.profile...
Validating profile Profiles/oem.profile...
The custom JumpStart configuration is ok.
```

この時点で、クライアント (この例では jordan) への JumpStart のインストールを開始できる必要があります。作成した JumpStart 構成、JumpStart ドライバ、終了スクリプト、およびプロファイルを使用します。

7. rules ファイルの確認中に問題が検出された場合は、118 ページの「rules ファイルの検証とチェック」を参照してください。

8. クライアントの ok プロンプトから次のコマンドを入力して、JumpStart インフラストラクチャーによってクライアントをインストールします。

```
ok> boot net - install
```

クライアントが構築されない場合は、構成を確認し、その構成が適切に機能するまで変更します。この節では、JumpStart 構成の一部の側面についてのみ説明します。詳細については、Sun BluePrint マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

rules ファイルを正しく実行し、パッチが正しくインストールされていることを確認したら、クライアントシステムの基本的なインストールと、その最小化と強化を開始できます。

rules ファイルの検証とチェック

rules ファイルの妥当性の検証時には、さまざまな問題が検出されることがあります。この節では、最も一般的な問題のいくつかについて説明します。

rules ファイルのチェックを初めて実行したときは、以下の出力が生成されます。

コード例 7-5　　rules ファイルの出力例

```
# pwd
/jumpstart
# ./check
Validating rules...
Validating profile Profiles/xsp-minimal-firewall.profile...
Error in file "rules", line 20
hostname jordan - Profiles/xsp-minimal-firewall.profile
Drivers/xsp-firewall-secure.driver
ERROR: Profile missing:
   Profiles/xsp-minimal-firewall.profile
```

この例では、jordan の rules 項目に指定されたプロファイルが存在しません。プロファイル xsp-minimal-firewall.profile が Profiles ディレクトリに存在しませんでした。一般に、このエラーは、ファイル名のスペルミス、プロファイルの正しいディレクトリの指定し忘れ、およびプロファイルが未作成であることが原因で生成されます。問題を修正して、チェックを再度実行します。

2 回目の実行では、別の問題が 2 つ検出されています。1 つ目の問題は、xsp-firewall-secure.driver で呼び出されるドライバに関するものです。xsp-firewall-secure.driver で、xsp-firewall-hardening.driver ではなく hardening.driver を呼び出しています。

2 つ目の問題では、JASS_SCRIPTS 変数に minimize-firewall.fin ではなく minimize-Sun_ONE-WS.fin が誤って設定されています。

以下に誤ったスクリプトを示します。

コード例 7-6　　誤ったスクリプトの例

```
#!/bin/sh
DIR="`/bin/dirname $0`"
export DIR
. ${DIR}/driver.init
. ${DIR}/config.driver
JASS_SCRIPTS="minimize-Sun_ONE-WS.fin"
. ${DIR}/driver.run
. ${DIR}/hardening.driver
```

以下に正しいスクリプトの例を示します。

コード例 7-7　　正しいスクリプトの例

```
#!/bin/sh
DIR="`/bin/dirname $0`"
export DIR
. ${DIR}/driver.init
. ${DIR}/config.driver
JASS_SCRIPTS="
minimize-firewall.fin"
. ${DIR}/driver.run
. ${DIR}/xsp-firewall-hardening.driver
```

強化構成のカスタマイズ

このファイアウォールの強化構成をカスタマイズして調整する準備が整いました。最初のスクリプトは hardening.driver に基づきます。具体的には、システムの「穴をすべてふさぐ」こと、つまりシステムのサービスがすべて無効になります。

Solaris 8 OS には Secure Shell クライアントが含まれていないため、ファイアウォールをネットワークベースで遠隔管理するには変更が必要です。このケースシナリオの場合、ファイアウォールの要件は、FTP サービスを有効のままにすること、および遠隔管理のために Secure Shell クライアントをインストールすることです。これら両方のサービスをプライベート管理ネットワークのみに限定することで、他のネットワークインタフェースでのリスニングを無効にします。これらのサービスの制限については、Sun BluePrints OnLine 掲載記事『Solaris Operating Environment Security: Updated for Solaris 9 Operating Environment』を参照してください。

これら 2 つのサービスを有効にしておくことに加え、RPC サービスも有効にして、Solstice DiskSuite GUI から Solstice DiskSuit を構成してディスクミラーリングを実行できるようにします。Solstice DiskSuite GUI を使用しないなら、RPC サービスは必要ありません。この例では、この GUI が必要なので、RPC サービスを有効のままにします。Solstice DiskSuite のインストールと構成方法については、このマニュアルでは説明しません。

このクライアントに必要な最後の変更は、xSP の中央 SYSLOG サーバーを使用するカスタム syslog.conf を作成することです。このカスタム syslog.conf ファイルは、各ファイアウォールシステムにインストールする必要があります。

上記変更を行った場合、さまざまな Solaris Security Toolkit 構成オプションを変更する必要があります。必要な各変更については、以下の節で詳しく説明します。

■ 121 ページの「FTP サービスの有効化」

- 122 ページの「Secure Shell ソフトウェアのインストール」
- 123 ページの「RPC サービスの有効化」
- 124 ページの「syslog.conf ファイルのカスタマイズ」

FTP サービスの有効化

このケースシナリオのファイアウォールの場合、FTP サービスを有効のままにしておきます。

▼ FTP サービスを有効にする

1. **FTP を有効のままにしておくには、`JASS_SVCS_DISABLE` および `JASS_SVCS_ENABLE` 変数を設定することにより、`update-inetd-conf.fin` ファイルのデフォルトの動作を変更します。**

   FTP を除くすべての標準 Solaris OS サービスを無効にする場合に、このケースシナリオで最適な方法は、`JASS_SVCS_ENABLE` が `ftp` になるよう定義し、`JASS_SVCS_DISABLE` には `finish.init` スクリプトから取得したデフォルト値をそのまま利用することです。『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

2. **環境変数を利用して変更を実装するには、以下のような項目を `xsp-firewall-secure.driver` に追加してから `xsp-firewall-hardening.driver` を呼び出します。**

   ```
   JASS_SVCS_ENABLE="ftp"
   ```

3. **ファイアウォールソフトウェアを利用して FTP を実装することにより、FTP が xSP の管理ネットワーク上でのみ使用できるようにします。**

   ほかの要件の 1 つは、FTP を xSP の管理ネットワーク上でのみ使用可能にすることでした。Solaris 8 OS の場合、この要件の実装には、システムに TCP ラッパーを組み込むか、ファイアウォールソフトウェアを利用します。このケースシナリオでは、ファイアウォールソフトウェアを利用します。

Secure Shell ソフトウェアのインストール

注 – ここでの説明は、Solaris OS バージョン 8 を実行しているシステムにのみ適用されます。システムで Solaris 9 または Solaris10 OS を稼働させている場合は、次に示す OpenSSH のインストール手順は省略し、Solaris OS に付属の Secure Shell ソフトウェアを使用できます。

Solaris 8 OS には Secure Shell クライアントは含まれません。。このため、システムが Solaris 8 OS を稼働させている場合は、Secure Shell クライアントをインストールしてリモート管理を行う必要があります。

Solaris Security Toolkit ソフトウェアを構成して、OpenSSH ツールをインストールできます。ここでは、`install-openssh.fin` スクリプトを使用します。このスクリプトは、`xsp-firewall-secure.driver` が使用する `config.driver` ファイルにリストされます。

▼ Secure Shell をインストールする

1. **デフォルトの `config.driver` を `xsp-firewall-config.driver` にコピーします。**
2. **ファイルのコピー内で `install-openssh.fin` の項目をコメント解除します。**
3. **`xsp-firewall-secure.driver` 内の `config.driver` を呼び出す項目を `xsp-firewall-config.driver` を呼び出すように変更します。**
4. **OpenSSH の最新バージョンを入手します。**

   パッチや OS リリースと同様、OpenSSH の最新バージョンを使用します。最新リリースについては、OpenSSH の Web ページを参照してください。

   `http://www.openssh.org`

5. **最新の OpenSSH パッケージをコンパイルし、名前を付けて、`Packages` ディレクトリにインストールします。**

   このパッケージについての詳細は、Sun BluePrints OnLine 掲載記事『Configuring OpenSSH for the Solaris Operating Environment』を参照してください。

6. **install-openssh.fin スクリプトを正しい OpenSSH パッケージ名で更新します。**

   install-openssh.fin スクリプトの更新も必要な場合があります。このスクリプトは、フォーマットする OpenSSH パッケージのパッケージ名を次のように定義します。

```
OBSDssh-3.5p1-sparc-sun4u-5.8.pkg
```

   ここでは、パッケージ名のあとにバージョン番号 (3.5p1) と、アーキテクチャー (sparc)、アーキテクチャーのバージョン (sun4u)、パッケージのコンパイル対象の OS (5.8)、pkg 接尾辞が続きます。

7. **ファイアウォールソフトウェアを利用して SSH を実装することにより、SSH を xSP の管理ネットワーク上でのみ使用できるようにします。**

   ほかの要件の 1 つは、Secure Shell を xSP の管理ネットワーク上でのみ使用可能にすることでした。Solaris 8 OS の場合、この要件の実装には、システムに TCP ラッパーを組み込むか、ファイアウォールソフトウェアを利用します。このケースシナリオでは、ファイアウォールソフトウェアを利用します。この要件の実装は、Secure Shell サーバーの構成を変更して行うこともできます。

RPC サービスの有効化

RPC サービスを有効のままにして、RPC を必要とするディスクミラーリングで Solstice DiskSuite を使用できるようにします。

固有の終了スクリプト disable-rpc.fin を使用すると、Solaris Security Toolkit の実行中に RPC サービスを無効にできるので、この変更は比較的簡単に行えます。

注 – システム上の RPC サービスへのリモートアクセスは、システムのファイアウォール構成によって明示的に拒否する必要があります。

▼ RPC を有効にする

- **xsp-firewall-hardening.driver 内の disable-rpc.fin の項目をコメントアウトします。**

  削除するのではなく、コメントにすることにより、ドライバからのスクリプトを無効にします。コメント値には特定の組み合わせしか使用できないため、JASS_SCRIPTS 定義内の項目をコメントアウトするときは注意が必要です。

  次に示すのは、driver.funcs script 内のコメントです。これは、Solaris Security Toolkit ソフトウェアで、JASS_SCRIPTS 定義内のコメント指示子として認識される要素についてのコメントです。

```
#Very rudimentary comment handler. This code will only recognize
#comments where a single '#' is placed before the file name
#(separated by white space or not). It then will only skip the
#very next argument.
```

syslog.conf ファイルのカスタマイズ

このクライアントに必要な最後の変更は、xSP の中央 SYSLOG サーバーを使用するカスタム syslog.conf を作成することです。このカスタム syslog.conf ファイルは、各ファイアウォールシステムにインストールする必要があります。

▼ syslog.conf ファイルをカスタマイズする

1. **xSP の標準 syslog.conf ファイルをコピーし、syslog.conf.jordan という名前を付けて、Files/etc ディレクトリに置きます。**

   Solaris Security Toolkit ソフトウェアでは、いくつか異なるモードでファイルをコピーできます。この構成に最適なオプションとしては、syslog.conf ファイルにファイアウォール固有の変更が含まれるので、システムのホスト名を接尾辞としてファイルに追加し、そのファイルを jordan にのみコピーすることです。ここの例では、クライアントは jordan と呼ばれているので、Files/etc で使用される実際のファイル名は syslog.conf.jordan です。JASS_FILES 定義にこの接尾辞が追加されないように注意が必要です。接尾辞についての詳細は、『Solaris Security Toolkit 4.2 リファレンスマニュアル』を参照してください。

2. **xS の標準 syslog.conf ファイルを使用できない場合は、以下のようにカスタム syslog.conf ファイルを作成します。**

   a. **Solaris Security Toolkit に付属する syslog.conf ファイルをコピーし、syslog.conf.jordan という名前を付けて、Files/etc ディレクトリに置きます。**

b. syslog.conf.jordan を変更して、SYSLOG の xSP 標準に準拠させます。

3. /etc/syslog.conf ファイルが xsp-firewall-hardening.driver の JASS_FILES 定義に表示されることを確認します。

デフォルトでは、xsp-firewall-hardening.driver 内の変更された JASS_FILE 定義は、以下のように表示されます。

コード例 7-8 変更された xsp-firewall-hardening.driver の出力例

```
JASS_FILES="
                    /etc/dt/config/Xaccess
                    /etc/init.d/inetsvc
                    /etc/init.d/nddconfig
                    /etc/init.d/set-tmp-permissions
                    /etc/issue
                    /etc/motd
                    /etc/notrouter
                    /etc/rc2.d/S00set-tmp-permissions
                    /etc/rc2.d/S07set-tmp-permissions
                    /etc/rc2.d/S70nddconfig
                    /etc/syslog.conf
"
```

これで、必要な変更がすべて完了しました。OS のインストール、最小化、および強化が特定のアプリケーション用にカスタマイズされ、完全に自動化されます。唯一完全に自動化されないプロセスは、ファイアウォールソフトウェアと Solstice DiskSuite の構成とインストールです。これらの構成は、JumpStart テクノロジを使用して実行できますが、このマニュアルでは説明しません。Sun BluePrints マニュアル『JumpStart Technology: Effective Use in the Solaris Operating Environment』を参照してください。

クライアントのインストール

ドライバにすべての変更を加えたら、この節での説明に従って、クライアントをインストールします。

▼ クライアントをインストールする

1. ドライバに必要な変更をすべて加えたら、JumpStart インフラストラクチャーを使用してクライアントをインストールします。

   クライアントの ok プロンプトから、次のコマンドを使用します。

```
ok> boot net - install
```

2. エラーが検出された場合は、それを修正してクライアントの OS を再インストールします。

品質保証のテスト

一連の作業の最後に、システムが提供するアプリケーションとサービスが正しく機能することを確認します。また、セキュリティープロファイルに必要な変更が正しく実装されていることも確認します。

この作業は、強化および最小化したプラットフォームの再起動の直後に入念に行うことが重要であり、検出された障害や問題はすぐに修正してください。この作業は、プロファイルのインストールの検証と、アプリケーションおよびサービスの機能性の検証という 2 つの作業に分かれています。

▼ プロファイルのインストールを確認する

Solaris Security Toolkit ソフトウェアによってセキュリティープロファイルが正しくインストールされ、エラーがないことを確認するには、次の点を確認および評価します。

1. インストールログファイルを確認します。

   このファイルは、JASS_REPOSITORY/jass-install-log.txt にインストールされています。

注 – このログファイルから、Solaris Security Toolkit ソフトウェアがシステムに実行した処理内容を正確に把握できます。実行のたびに、その開始時刻に基づいて、ディレクトリ内に新しいログファイルが保存されます。これらのファイルや、JASS_REPOSITORY ディレクトリ内のほかのファイルを直接変更してはなりません。

---

2. **監査オプションを使用して、システムのセキュリティー構成を評価します。**

   監査オプションについての詳細は、第 6 章を参照してください。このシナリオでは、クライアント上の Solaris Security Toolkit ソフトウェアのインストール先ディレクトリから以下のコマンドを使用します。

   コード例 7-9　　セキュリティー構成の評価

```
# ./jass-execute -a xsp-firewall-secure.driver
[NOTE] Executing driver, xsp-firewall-secure.driver
================================================================
xsp-firewall-secure.driver: Driver started.
================================================================

================================================================
Solaris Security Toolkit Version:   4.2.0
[...]
```

   Solaris Security Toolkit の確認で何らかの矛盾が検出された場合は、その内容が記録されます。実行レポートの最後に、矛盾箇所の検出数の合計が記録されます。実行の出力全体は、JASS_REPOSITORY ディレクトリ内にあります。

▼ アプリケーションとサービスの機能を確認する

アプリケーションとサービスの確認作業では、適切に定義されたテストおよび受け入れ計画を実行します。この計画を使用して、システムやアプリケーションの各種のコンポーネントを実行し、それらが使用可能で正常に機能する状態であることを確認します。このような計画を用意**できない**場合は、システムの使用方法に応じて妥当な方法でテストします。目標は、強化プロセスがアプリケーションやサービスの機能の実行に一切影響していないことを確認することです。

1. **システムの強化後にアプリケーションやサービスが正常に機能しないことが検出された場合は、第 2 章で説明する方法で問題を特定します。**

   たとえば、truss コマンドを使用します。このコマンドを使用すると、アプリケーションの問題がどの段階で発生するかを特定できます。問題が発生する段階がわかれば、問題箇所を絞り込んで、Solaris Security Toolkit ソフトウェアが行った変更箇所まで遡ることができます。

注 – このマニュアルに記載のアプローチは、Solaris Security Toolkit ソフトウェアを配備した多くのユーザーの経験に基づいているため、ほとんどの問題を回避できます。

---

2. **同様の方法で、Check PointFirewall-1 NG ソフトウェアをテストし、Solaris Security Toolkit ソフトウェアによる変更箇所までさかのぼって問題を修正します。**

3. パッケージの最終的なリストを変更する必要がある場合は、プロファイルを変更してシステムを再インストールし、テストを繰り返します。

用語集

ここでは、Solaris Security Toolkit で使用されている略語と頭字語を一覧にまとめています。

A

ab2	AnswerBook2
ABI	Application Binary Interface (アプリケーションバイナリインタフェース)
ARP	Address Resolution Protocol (アドレス解決プロトコル)
ASPPP	Asynchronous Point-to-Point Protocol (非同期ポイントツーポイントプロトコル)

B

BART	Basic Auditing and Reporting Tool (基本監査報告機能)
BIND	Berkeley Internet Name Domain
BSD	Berkeley Software Distribution
BSM	Basic Security Model (Solaris)

C

CD compact disc (コンパクトディスク)

CD-ROM compact disc read-only memory (コンパクトディスク読み取り専用メモリー)

CDE Common Desktop Environment (共通デスクトップ環境)

cp(1) ファイルコピーコマンド

cron(1M) クロックデーモンコマンド

D

DHCP Dynamic Host Configuration Protocol (動的ホスト構成プロトコル)

DMI Desktop Management Interface (デスクトップ管理インタフェース)

DMTF Distributed Management Task Force

DNS Domain Name System (ドメインネームシステム)

E

EEPROM electronically erasable programmable read-only memory (電気的に消去できるプログラム可能な読み取り専用メモリー)

F

FACE Framed Access Command Environment

FMRI Fault Management Resource Identifier (障害管理リソース識別子)

FTP File Transfer Protocol (ファイル転送プロトコル)

G

GID group identifier (グループ識別子)

GUI graphical user interface (グラフィカルユーザーインタフェース)

H

HSFS High Sierra File System

HTTP HyperText Transfer Protocol (ハイパーテキストトランスファープロトコル)

I

ID identifier (識別子)

IETF Internet Engineering Task Force (インターネット特別技術調査委員会)

INETD Internet service daemon (インターネットサービスデーモン)

IP Internet Protocol (インターネットプロトコル)

IPF Internet Protocol Filter (インターネットプロトコルフィルタ)

ISA instruction set architecture (命令セットアーキテクチャー)

J

JASS JumpStart Architecture and Security Scripts (現在は Solaris Security Toolkit)

K

KDC Kerberos Key Distribution (Kerberos 鍵配布)

L

LDAP　Lightweight Directory Access Protocol

lp(1)　ラインプリンタコマンド (印刷要求の発行)

M

MAN　management network (管理ネットワーク) (Sun Fire ハイエンドシステムの内部 I1 ネットワーク)

MD5　message-digest 5 algorithm (メッセージダイジェスト 5 アルゴリズム)

MIP　Mobile Internet Protocol (モバイルインターネットプロトコル)

MSP　midframe service processor (ミッドフレームサービスプロセッサ)

mv(1)　ファイルの移動コマンド

N

NFS　Network File System (ネットワークファイルシステム)

NG　Next Generation (次世代)

NGZ　non-global zone (非グローバルゾーン)

NIS、NIS+　Network Information Services (ネットワーク情報サービス)

NP　no password (パスワードなし)

NSCD　name service cache daemon (ネームサービスキャッシュデーモン)

O

OEM　Original Equipment Manufacturer

OS　Operating System (オペレーティングシステム)

P

PAM Pluggable Authentication Module

PDF Portable Document Format

Perl Practical Extraction and Report Language

PICL Platform Information and Control Library

PPP Point-to-Point Protocol (ポイントツーポイントプロトコル)

PROM programmable read-only memory (プログラム可能な読み取り専用メモリー)

Q

QA quality assurance (品質保証)

R

RBAC role-based access control (ロールベースのアクセス制御)

rc 実行コントロール (ファイルまたはスクリプト)

`rlogin`(1) リモートログインコマンド

RFC Remote Function Call

RPC Remote Procedure Call (遠隔手続き呼び出し)

`rsh`(1) リモートシェルコマンド

S

SA system administrator (システム管理者)

SC system controller (システムコントローラ) (Sun Fire ハイエンドシステムおよびミッドレンジシステム)

scp(1)	セキュアコピーコマンド (遠隔ファイルコピープログラム)
SCCS	Source Code Control System (ソースコード制御システム)
SLP	Service Location Protocol (サービスロケーションプロトコル)
SMA	System Management Agent (システム管理エージェント)
SMC	Solaris Management Console
SMF	Service Management Facility (サービス管理機能)
SMS	System Management Services (システム管理サービス)
SNMP	Simple Network Management Protocol (簡易ネットワーク管理プロトコル)
SP	service provider (サービスプロバイダ)
SPARC	Scalable Processor Architecture
SPC	SunSoft Print Client
SSH	Secure Shell (セキュアシェル) (Solaris 9 および 10 OS)
SVM	Solaris Volume Manager (Solaris ボリュームマネージャー)

T

TCP	Transmission Control Protocol
tftp(1)	trivial file transfer program
ttl	time-to-live (生存時間)

U

UDP	User Diagram Protocol
UFS	Unix File System (UNIX ファイルシステム)
UID	user identifier (ユーザー識別子)
UUCP	UNIX-to-UNIX Copy (UNIX-to-UNIX コピー)

V

VOLD Volume Management daemon (ボリューム管理デーモン)

W

WBEM Web-based Enterprise Management

索引

T

U

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

そ

た

ひ

ふ

へ

ほ

ま

む

め

も

ゆ

よ