Oracle® Solaris 11.4 でのシステムおよび接続されたデバイスのセキュリティー保護

Part No: E75230-01
2018 年 8 月

Part No: E75230-01

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility ProgramのWeb サイト(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc)を参照してください。

Oracle Supportへのアクセス

サポートをご契約のお客様には、My Oracle Supportを通して電子支援サービスを提供しています。詳細情報は(http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info) か、聴覚に障害のあるお客様は (http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs)を参照してください。

目次

このドキュメントの使用法

- **概要** – Oracle Solaris システムへのアクセスを制御およびモニターする方法について説明します。ウィルス対策ソフトウェアの使用方法も示します。
- **対象者** – 企業ネットワーク上にセキュリティーを実装する責任のあるシステム管理者。
- **必要な知識** – Oracle Solaris でサポートされているセキュリティーの概念および機能について熟知していること。

製品ドキュメントライブラリ

この製品および関連製品のドキュメントとリソースは `http://www.oracle.com/pls/topic/lookup?ctx=E75431-01` で入手可能です。

フィードバック

このドキュメントに関するフィードバックを `http://www.oracle.com/goto/docfeedback` からお聞かせください。

♦ ♦ ♦ 第 1 章

1 コンピュータシステムセキュリティーの管理

コンピュータシステムを改ざんやマルウェアに対して安全な状態に保持することは、システム管理の重要な責任です。この章では、コンピュータシステムのセキュリティー管理に関する概要を説明します。

システムおよびデバイスをセキュリティー保護する Oracle Solaris 11.4 の新機能

このセクションでは、既存のお客様のために、未承認アクセスからシステムおよびデバイスをセキュリティー保護する、このリリースの重要な新機能に関する情報に焦点を当てます。

パスワードおよびログインの新機能については、次を参照してください。

- 16 ページの「パスワードパラメータ」
- 20 ページの「OTP とスマートカードによる 2 要素認証」
- `/etc` ディレクトリ内のファイルではなく `account-policy` SMF サービスを使用してシステムセキュリティーの値を追跡する方法については、`account-policy(8S)` のマニュアルページおよび『Securing Users and Processes in Oracle Solaris 11.4』の「Modifying Rights System-Wide As SMF Properties」を参照してください。
- ベリファイドブートの変更内容については、38 ページの「ベリファイドブートのためのファームウェアアップグレード」を参照してください。

- セキュリティー拡張の変更内容については、次を参照してください。
  - 58 ページの「adiheap を使用したプロセスヒープの破損の防止」
  - 59 ページの「adistack を使用した ADI ベースのスタック保護」
  - 60 ページの「セキュリティー拡張のステータス継承の有効化」
  - 62 ページの「オブジェクトごとのセキュリティー拡張の指定」

コンピュータシステムへのアクセスを制御する

ワークスペースでは、サーバーに接続されたすべてのコンピュータを 1 つの大規模多重システムと見なすことができます。システム管理者は、この大規模なシステムのセキュリティー管理に責任があります。システム管理者は、ネットワークの外部からの侵入を防ぐ必要があります。また、ネットワーク内部のコンピュータ上のデータの完全性を確保する必要もあります。

ファイルレベルにおいて、Oracle Solaris には標準セキュリティー機能が組み込まれており、ファイル、ディレクトリ、および接続されたデバイスを保護するために使用できます。システムレベルとネットワークレベルでは、セキュリティーの内容はほぼ同じです。以降のセクションで説明されているように、セキュリティー防御の第一線はシステムへのアクセスを制御することです。

物理的なセキュリティーの管理

システムへのアクセスを制御するには、コンピュータ環境の物理的なセキュリティーを管理する必要があります。たとえば、システムにログインしたままこれを放置することは未承認アクセスを招く原因になります。侵入者がオペレーティングシステムやネットワークにアクセスしないとも限らないからです。コンピュータの周辺環境やコンピュータハードウェアは、不当なアクセスから物理的に保護される必要があります。

ハードウェア設定に対する未承認アクセスから SPARC システムを保護できます。eeprom コマンドを使って、パスワードがないと PROM にアクセスできないようにしてください。詳細は、74 ページの「SPARC ハードウェアへのアクセスにパスワードを必要にする方法」を参照してください。x86 ハードウェアを保護するには、ベンダーのドキュメントを参照してください。

ブートプロセスへのアクセスの制御

Oracle Solaris では、ブートプロセスへのアクセスを制御する 2 つのテクノロジが提供されています。

- ベリファイドブート – 署名付きのブートおよびカーネルソフトウェアのみにシステムでの実行を許可します。

  `boot_policy` プロパティーの値がベリファイドブートを制御します。このポリシーには、`bootblk` のチェックと、すべてのカーネルモジュール (`unix` および `genunix` を含む) のロードが含まれています。

  ポリシー設定は、Oracle ILOM、Fujitsu SPARC M12、Fujitsu M10 XSC などのサービスプロセッサ (SP) に格納されます。SP がハードウェアプラットフォームを管理します。セキュリティー上の理由から、ポリシー設定は意図的に、ブートされる Oracle Solaris 環境の外部に格納されます。

  詳細は、40 ページの「ベリファイドブートのポリシー」を参照してください。
- Trusted Platform Module (TPM) – システムをセキュリティー保護するための暗号化機能を提供する専用マイクロコントローラ。TPM は、暗号化キーストアを提供し、システムのブートに使用されるファームウェアおよびソフトウェアのハッシュを記録します。

USB ポートへのアクセスの制御

SPARC T7 プラットフォームなど、一部の Oracle プラットフォームには外部 USB ポートがあります。システムコントローラ、配電盤 (PDU)、ネットワークスイッチなどのデバイスが、このような USB 接続を使用することがあります。システム管理者は、USB 接続を悪用した攻撃からシステムを保護する必要があります。Oracle Integrated Lights Out Manager (ILOM) では、外部 USB ポートを介したシステムアクセスを拒否または制限できます。

詳細は、53 ページの「ILOM を使用した、USB ポートへのアクセスの防止」を参照してください。

ログインの制御

パスワード割り当てとログイン制御によって、システムやネットワークへの未承認のログインを防止できます。パスワードはシンプルな認証メカニズムです。システム上のすべてのアカウントには、パスワードが必要です。アカウントにパスワードを設定しないと、ユーザー名を推測できる侵入者であれば誰でもネットワーク全体にアクセスできることになります。力ずくの野蛮な攻撃を許さないためには、強力なパスワードアルゴリズムが必要です。

ユーザーがシステムにログインすると、`login` コマンドはネームスイッチサービス `svc:/system/name-service/switch` 内の情報に従って、該当するネームサービスまたはディレクトリサービスデータベースを確認します。ネームサービスデータベースの値を変更するには、SMF コマンドを使用します。ネームサービスは、ログインに影響を与えるデータベースの場所を示します。

- `files` – ローカルシステムの `/etc` ファイルを指定します
- `ldap` – LDAP サーバーの LDAP ディレクトリサービスを指定します
- `nis` – NIS マスターサーバーの NIS データベースを指定します
- `dns` – ネットワーク上のドメインネームサービスを指定します。

ネームサービスの説明は、`nscd(8)` のマニュアルページを参照してください。ネームサービスおよびディレクトリサービスについては、『Oracle Solaris 12 ディレクトリサービスとネームサービスでの作業: DNS と NIS』および『Oracle Solaris 12 ディレクトリサービスとネームサービスでの作業: LDAP』を参照してください。

`login` コマンドは、ユーザーによって指定されたユーザー名とパスワードを検証します。ユーザー名がパスワードデータベース内に存在しない場合、`login` コマンドはシステムへのアクセスを拒否します。あるいは、指定されたユーザー名に対するパスワードが正しくないと、`login` コマンドはシステムへのアクセスを拒否します。有効なユーザー名とそれに対応するパスワードが入力されれば、システムはシステムへのアクセスをユーザーに認可します。

PAM モジュールには、システムへのログインが正常に完了したあとのアプリケーションへのログインを効率化できます。詳細は、『Managing Authentication in Oracle Solaris 11.4』 の 第 1 章, 「Using Pluggable Authentication Modules」を参照してください。

Oracle Solaris システムには、精巧な認証メカニズムと承認メカニズムが備わっています。ネットワークレベルでの認証メカニズムや承認メカニズムについては、33 ページの「リモートアクセスの認証と承認」を参照してください。

パスワード情報の管理

ユーザーはシステムにログインするときに、ユーザー名とパスワードの両方を入力する必要があります。ログイン名は公開されていますが、パスワードは秘密にしなければなりません。ユーザーは、自分のパスワードを他人に知られてはいけません。

組織は業界標準に従ったパスワードポリシーを持つべきです。ユーザーは、自分のパスワードを慎重に選択し、サイトのパスワードポリシーに従う必要があります。

ユーザーの初期パスワードは、ユーザーのアカウントを設定するときに作成します。パスワードをロックすることでユーザーアカウントを無効にできます。詳細については、次をレビューしてください。

- 『Managing User Accounts and User Environments in Oracle Solaris 11.4』 の 第 1 章, 「About User Accounts and User Environments」

- 『Oracle Solaris 11.4 Security and Hardening Guidelines』 の 「Passwords and Password Policy」
- passwd(1)

ローカルパスワード

ネットワークでローカルファイルを使用してユーザーを認証している場合、パスワード情報はシステムの /etc/passwd ファイルと /etc/shadow ファイルに保持されます。ユーザー名などの情報は、/etc/passwd ファイルに保持されます。暗号化されたパスワード自体は、個別のシャドウファイル (/etc/shadow) に保持されます。このセキュリティー方式によって、暗号化されたパスワードにアクセスされることを防ぎます。/etc/passwd ファイルは、システムにログインできるすべてのユーザーが使用できますが、/etc/shadow ファイルを読み取ることができるのは root アカウントだけです。passwd コマンドを使用すると、ローカルシステム上のユーザーのパスワードを変更できます。

NIS パスワード

ネットワークで NIS を使用してユーザーを認証している場合、パスワード情報は NIS パスワードマップに保持されます。NIS では、パスワードの有効期間を指定できません。NIS パスワードマップに保持されているユーザーのパスワードを変更するには、コマンド passwd -r nis を使用します。

LDAP パスワード

Oracle Solaris の LDAP ネームサービスは、パスワード情報とシャドウ情報を LDAP ディレクトリツリーの ou=people コンテナに格納します。Oracle Solaris LDAP ネームサービスクライアントでユーザーのパスワードを変更するには、passwd -r ldap コマンドを使用します。LDAP ネームサービスは、パスワードを LDAP リポジトリに格納します。

パスワードポリシーは Oracle Directory Server Enterprise Edition で適用されます。具体的には、クライアントの pam_ldap モジュールは Oracle Directory Server Enterprise Edition で適用されているパスワードポリシー制御に従います。詳細は、『Working With Oracle Solaris 11.4 Directory and Naming Services: LDAP』 の 「LDAP Naming Service Security Model」 を参照してください。

パスワードパラメータ

Oracle Solaris 11.4 リリースでは、デフォルトのパスワード長およびサポートされるパスワードハッシュが変更され、期間のパラメータが追加されています。ファイル内のパスワードパラメータ情報を表示して変更する従来の方法は、SMF サービス `account-policy` で置き換えられています。

注記 - `account-policy` SMF ステンシルを使用している場合で、`config/etc_default_passwd` プロパティーが有効になっているときは、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」を確認してください。`account-policy(8S)` のマニュアルページも参照してください。

次に示す `account-policy` SMF ステンシル内のパスワード属性のリストは、構成可能なパスワードパラメータを示しています。

```
password/history count
password/value_authorization astring solaris.account.setpolicy
password/aging_defaults/max_days count
password/aging_defaults/min_days count
password/aging_defaults/warn_days count
password/complexity/max_repeats count
password/complexity/min_alpha count
password/complexity/min_diff count
password/complexity/min_digit count
password/complexity/min_lower count
password/complexity/min_nonalpha count
password/complexity/min_special count
password/complexity/min_upper count
password/complexity/namecheck boolean
password/complexity/passlength count
password/complexity/whitespace boolean
password/crypt/algorithms_allow astring 2a 5 6
password/crypt/algorithms_deprecate astring
password/crypt/default astring 5
password/dictionary/db_dir astring
password/dictionary/min_word_length count
password/dictionary/word_list astring
```

パスワードの長さを 8 文字未満にすることはできなくなりました。ユーザーのパスワード長は変更できます。

パスワードハッシュ

パスワードの強力な暗号化は攻撃に対する最初の防壁になります。Oracle Solaris ソフトウェアには 6 つのパスワード暗号化アルゴリズムが用意されています。SHA アルゴリズムは、強力なパスワード暗号化を提供します。

注記 - FIPS 140-2 承認にするには、SHA アルゴリズムを使用します。詳細は、『Oracle Solaris 12 での FIPS 140 対応システムの使用』 の 「passwd Command as a FIPS 140-2 Consumer」を参照してください。

パスワードアルゴリズムの識別子

`account-policy` SMF ステンシルの `config/etc_default_passwd` プロパティーを有効にすることにより、サイトのアルゴリズム構成を指定できます。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」を確認してください。`account-policy(8S)` のマニュアルページも参照してください。

次の表に示すように、アルゴリズムを識別子で指定します。識別子とアルゴリズムのマッピングについては、`/etc/security/crypt.conf` ファイルを参照してください。

注記 - 可能な場合は、FIPS 140-2 承認アルゴリズムを使用してください。FIPS 140-2 承認アルゴリズムのリストについては、『Oracle Solaris 12 での FIPS 140 対応システムの使用』 の 「FIPS 140-2 Algorithm Lists and Certificate References for Oracle Solaris Systems」を参照してください。

表 1 パスワードハッシュアルゴリズム

識別子	説明	アルゴリズムのマニュアルページ
`1`	BSD システムや Linux システムの MD5 アルゴリズムと互換性のある MD5 アルゴリズム。	Unresolved link to " crypt_bsdmd57"
`2a`	BSD システムの Blowfish アルゴリズムと互換性のある Blowfish アルゴリズム。 **注記 -** FIPS 140-2 セキュリティーを向上させるには、`password/crypt/algorithms_allow` から Blowfish アルゴリズム (2a) を削除します。	Unresolved link to " crypt_bsdbf7"
`md5`	BSD バージョンや Linux バージョンの MD5 よりも強力とされている Sun MD5 アルゴリズム。	Unresolved link to " crypt_sunmd57"
`5`	SHA256 アルゴリズム。SHA は、Secure Hash Algorithm (セキュアハッシュアルゴリズム) を表します。このアルゴリズムは、SHA-2 ファミリのメンバーです。SHA256 では 255 文字のパスワードがサポートされます。このアルゴリズムがデフォルトです (`CRYPT_DEFAULT`)。	Unresolved link to " crypt_sha2567"
`6`	SHA512 アルゴリズム。	Unresolved link to " crypt_sha5127"
`__unix__`	非推奨。従来の UNIX 暗号化アルゴリズム。このアルゴリズムは、古いシステムに接続するときに使用できます。	Unresolved link to " crypt_unix7"

注記 - そのユーザーの新しいパスワードを生成する際は、ユーザーの初期パスワードに使用されたアルゴリズムが引き続き使用されます (ユーザーの新しいパスワードを生成する前に、別のデフォルトアルゴリズムが選択された場合でも)。このメカニズムは、次の条件で適用されます。

- アルゴリズムがパスワード暗号化で使用することが許可されているアルゴリズムのリストに含まれている。
- 識別子が `_unix_` 以外である。

パスワード暗号化のアルゴリズムを切り替える方法については、67 ページの「パスワード暗号化のデフォルトアルゴリズムを変更する」を参照してください。

パスワードハッシュ構成

注記 - `account-policy` SMF ステンシルを使用している場合で、`config/etc_default_passwd` プロパティーが有効になっているときは、SMF でパスワード構成を変更できます。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』の「Modifying Rights System-Wide As SMF Properties」を参照してください。`account-policy(8S)` のマニュアルページも参照してください。

`account-policy` サービスには、パスワードハッシュに影響を与えるパラメータが 3 つあります。

```
password/crypt/algorithms_allow astring 2a 5 6
password/crypt/algorithms_deprecate astring
password/crypt/default astring 5
```

`password/crypt/default` の値を変更すると、新規ユーザーのパスワードは、新しい値に関連付けられたアルゴリズムで暗号化されます。

既存のユーザーがパスワードを変更したときに新しいパスワードがどのアルゴリズムで暗号化されるかは、古いパスワードがどのように暗号化されているかによって異なります。たとえば、管理者がパスワードパラメータを `CRYPT_ALGORITHMS_ALLOW=1,2a,md5,5,6` および `password/crypt/default=6` に変更したとします。次の表は、パスワードの暗号化にどのアルゴリズムが使用されるかを示します。パスワードは「識別子=アルゴリズム」で構成されます。

元のパスワード	変更後のパスワード	説明
`1` = `crypt_bsdmd5`	同じアルゴリズムを使用します。	`1` 識別子は `CRYPT_ALGORITHMS_ALLOW` リストにあります。ユーザーのパスワードは引き続き `crypt_bsdmd5` アルゴリズムで暗号化されます。
`2a` = `crypt_bsdbf`	同じアルゴリズムを使用します。	`2a` 識別子は `CRYPT_ALGORITHMS_ALLOW` リストにあります。このため、新しいパスワードは `crypt_bsdbf` アルゴリズムで暗号化されます。

元のパスワード	変更後のパスワード	説明
md5 = crypt_md5	同じアルゴリズムを使用します。	md5 識別子は CRYPT_ALGORITHMS_ALLOW リストにあります。このため、新しいパスワードは crypt_md5 アルゴリズムで暗号化されます。
5 = crypt_sha256	同じアルゴリズムを使用します。	5 識別子は CRYPT_ALGORITHMS_ALLOW リストにあります。このため、新しいパスワードは引き続き crypt_sha256 アルゴリズムで暗号化されます。
6 = crypt_sha512	同じアルゴリズムを使用します。	6 識別子は CRYPT_DEFAULT の値です。このため、新しいパスワードは引き続き crypt_sha512 アルゴリズムで暗号化されます。
__unix__ = crypt_unix	crypt_sha512 アルゴリズムを使用します。	__unix__ 識別子は CRYPT_ALGORITHMS_ALLOW リストにありません。このため、crypt_unix アルゴリズムを使用することはできません。新しいパスワードは CRYPT_DEFAULT アルゴリズムで暗号化されます。

選択したアルゴリズムの構成の詳細については、account-policy(8S) のマニュアルページを参照してください。パスワード暗号化アルゴリズムを指定する場合は、67 ページの「パスワード暗号化のデフォルトアルゴリズムを変更する」を参照してください。

特殊なシステムアカウント

root アカウントは特殊なシステムアカウントの 1 つです。これらのアカウントのうち、root アカウントにのみパスワードが割り当てられ、ログインできます。nuucp アカウントはファイル転送用にログインできます。他のシステムアカウントは、ファイルを保護したり、または root の完全な権限を使用せずに管理プロセスを実行したりします。

注意 - システムアカウントのパスワード設定は決して変更しないでください。Oracle Solaris からのシステムアカウントは、安全かつ確実な状態で配布されます。UID が 101 以下のシステムファイルは修正したり、作成したりしないでください。

次の表に、一部のシステムアカウントとその使用方法の一覧を示します。システムアカウントは特殊な機能を実行します。この一覧の各アカウントは、100 より小さい UID を持ちます。システムファイルの完全なリストを表示するには、logins -s コマンドを使用します。

表 2 選択されたシステムアカウントとその使用

システムアカウント	UID	用途
root	0	ほぼ無制限です。他の保護および許可をオーバーライドできます。root アカウントはシステム全体へのアクセス権を持ちます。root アカウントのパスワードは、非常に注意深く保護するようにしてください。root アカウントは、ほとんどの Oracle Solaris コマンドを所有しています。

システムアカウント	UID	用途
daemon	1	バックグラウンド処理を制御します。
bin	2	一部の Oracle Solaris コマンドを所有します。
sys	3	多数のシステムファイルを所有します。
adm	4	一部のシステム管理ファイルを所有します。
lp	71	プリンタ用のオブジェクトデータファイルとスプールデータファイルを所有します。
uucp	5	UNIX 間のコピープログラム、UUCP 用のオブジェクトデータファイルとスプールデータファイルを所有します。
nuucp	9	システムにログインしてファイル転送を開始するためにリモートシステムで使用されます。

OTP とスマートカードによる 2 要素認証

Oracle Solaris はスマートカードとワンタイムパスワード (OTP) をサポートしています。これらのテクノロジでは、ユーザーは識別情報を 2 つの形式で提供する必要があります。最初の形式は、UNIX ユーザー名とパスワードです。2 つ目は、スマートカードと PIN か、またはモバイルオーセンティケータと OTP です。『Managing Authentication in Oracle Solaris 11.4』 の 第 3 章,「Using Smart Cards for Multifactor Authentication in Oracle Solaris」および『Managing Authentication in Oracle Solaris 11.4』 の 第 4 章,「Using One-Time Passwords for Multifactor Authentication in Oracle Solaris」を参照してください。

リモートログイン

侵入者にとって、リモートログインは魅力的な手段です。Oracle Solaris は、リモートログインをモニター、制限、および無効にする、いくつかのコマンドを提供します。手順については、表4を参照してください。

デフォルトでは、システムのマウスやキーボード、フレームバッファー、オーディオデバイスなど、ある種のシステムデバイスについては、リモートログインを通して制御したり読み取ったりすることはできません。詳細は、`logindevperm(5)` のマニュアルページを参照してください。

システムリソースへのアクセス制御

一部のシステムリソースは、デフォルトで保護されています。さらに、システム管理者はシステムの動作状態を制御したり、モニターしたりすることができます。システム管理者は、だれがどのリソースを使用できるかを制限したり、リソースの使用状況を記録したり、だれがリソースを使用しているかをモニターしたりできます。システ

ム管理者は、リソースの不適切な使用を最小限に抑えるようにシステムを設定することもできます。

デフォルトでのセキュリティー強化 (Secure By Default) 構成の使用

デフォルトでは、Oracle Solaris がインストールされると、一連の多数のネットワークサービスが無効になります。この構成は「デフォルトでのセキュリティー強化 (Secure By Default)」(SBD) と呼ばれます。SBD により、ネットワークリクエストを受け入れるネットワークサービスは `sshd` デーモンだけになります。ほかのネットワークサービスはすべて無効になるか、ローカル要求だけを処理するようになります。`ftp` などの個々のネットワークサービスを有効にするには、Oracle Solaris のサービス管理機能 (SMF) を使用します。詳細は、`smf(7)` のマニュアルページを参照してください。

システムリソースの意図的な誤用の回避

マルウェアがメモリー、プロセスヒープ、バッファーなどカーネル内の脆弱な部分を対象とする場合があります。Oracle Solaris では、アプリケーションをマルウェアから保護するためのセキュリティー拡張フレームワークを提供しています。セキュリティー拡張はシステムレベルでのセキュリティー防御を提供しますが、アプリケーションでは特定の防御を利用するかどうかをそれぞれ明示的に選択できます。

セキュリティー拡張フレームワークは、その名が示すとおり、特定のカーネルオブジェクトおよびハードウェアオブジェクトにセキュリティーを拡張することによって、ハードウェアまたはカーネルソフトウェアに対する悪意のある攻撃に対抗するように設計されています。どの場合でも、このフレームワークでは、管理者がシステム環境のリスクの程度を判断して拡張を有効にし、リスクにさらされているオブジェクトへの攻撃を軽減できます。フレームワークによる一部の軽減策によってパフォーマンスが低下する場合があります。

詳細は、54 ページの「セキュリティー拡張を使用した、マルウェアに対する保護」および `sxadm(8)` のマニュアルページを参照してください。

スーパーユーザーアクセスの制限とモニタリング

システムでスーパーユーザーアクセスを行うには、`root` パスワードが必要です。デフォルトの構成では、ユーザーはリモートからシステムに `root` としてログインできません。リモートログインするときに、ユーザーは自分のユーザー名でログインしてから、`su` コマンドを使用して `root` になる必要があります。管理者は、必

要に応じて su コマンドを使用中のユーザー (特にスーパーユーザーのアクセス権を取得しようとしているユーザー) をモニターできます。スーパーユーザーをモニタリングしたり、スーパーユーザーのアクセス権を制限したりする手順については、71 ページの「root アクセスのモニタリングと制限」を参照してください。

役割に基づくアクセス制御を構成してスーパーユーザーを置き換える

Oracle Solaris の機能である役割に基づくアクセス制御 (RBAC) は、スーパーユーザーの権限を管理役割に分散します。役割のこれらの権限は、権利プロファイルと呼ばれるバンドルを介して取得されます。標準ユーザーにも権利プロファイルを割り当てることができます。

スーパーユーザーすなわち root ユーザーは、システムのすべてのリソースにアクセスできますが、RBAC を使用すると、root の責任の多くを、個別の権限を持つ一連の役割に置き換えることができます。たとえば、ユーザーアカウントの作成を処理する 1 つの役割と、システムファイルの変更を処理する別の役割を設定できます。root アカウントを変更しない場合でも、このアカウントを役割として残し、その役割を割り当てないようにできます。この方法によって、システムへの root アクセスが事実上削除されます。

各役割を使用するには、既知のユーザーが自分のユーザー名とパスワードを使用してログインする必要があります。ログインしたユーザーは、特別な役割パスワードを入力してその役割を引き受けます。ユーザーに権利プロファイルを直接割り当てた場合、ユーザーは管理権限を取得するためにプロファイルシェルを開く必要があります。RBAC の詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「User Rights Management」を参照してください。

システムリソースの意図しない誤用の回避

システム管理者は、自分自身やユーザーによって意図しないエラーが引き起こされないように防止できます。

- PATH 変数を正しく設定することによって、トロイの木馬の実行を防止できます。
- 制限されたシェルをユーザーに割り当てることもできます。システムのうち各人の作業に必要な部分だけをユーザーに提供するという方法でシェル機能を制限すると、ユーザーエラーを避けることができます。実際、慎重に設定すれば、作業を能率的に行う上で必要な部分以外にユーザーがアクセスできないように制限できます。
- そのユーザーがアクセスする必要がないファイルには、限定的なアクセス権を設定できます。

PATH 変数の設定

よく注意して、PATH 変数を正しく設定してください。そうしなければ、だれかが持ち込んだプログラムを誤って実行してしまい、セキュリティーが危険にさらされる可能性があります。データを壊したりシステムを損傷したりするおそれがあります。このようなプログラムは、「トロイの木馬」と呼ばれます。たとえば、公開ディレクトリの中に別の su プログラムが置かれていると、システム管理者が気づかずに実行してしまう可能性があります。このようなスクリプトは正規の su コマンドとまったく同じに見えます。このようなスクリプトは実行後に自らを削除してしまうため、トロイの木馬が実際に実行されたという証拠はほとんど残りません。

PATH 変数はログイン時に自動的に設定されます。このパスは、.bashrc や /etc/profile などの初期設定ファイルを通して設定されます。現在のディレクトリ (.) への検索パスを最後に指定すれば、トロイの木馬のようなタイプのプログラムを実行するのを防ぐことができます。root アカウントの PATH 変数には現在のディレクトリを一切含めないようにしてください。

ユーザーに制限付きシェルを割り当てる

標準シェルを使用すると、ユーザーはファイルを開く、コマンドを実行するなどの操作を行うことができます。制限付きシェルを使用すると、ディレクトリの変更やコマンドの実行などのユーザー能力を制限できます。制限付きシェルは、/usr/lib/rsh コマンドで呼び出されます。制限付きシェルは、リモートシェル /usr/sbin/rsh ではありません。

標準のシェルと異なる点は次のとおりです。

- ユーザーのアクセスはホームディレクトリ内に限定されるため、ユーザーは cd コマンドを使用してディレクトリを変更できません。したがって、システムファイルを閲覧することはできません。
- ユーザーは PATH 変数を変更できないため、システム管理者によって設定されたパスのコマンドしか使用できません。さらに、完全なパス名を使ってコマンドやスクリプトを実行することもできません。
- ユーザーは、> または >> を使用して出力をリダイレクトできません。

制限付きシェルでは、ユーザーが使用できるシステムファイルを制限できます。このシェルは、特定のタスクを実行するユーザーのために限られた環境を作成します。ただし、制限付きシェルは完全にセキュアなわけではなく、あくまでも経験の少ないユーザーが誤ってシステムファイルを損傷するのを防止することが目的です。

制限付きシェルについては、man -s8 rsh コマンドを使用して rsh(8) のマニュアルページを参照してください。

ファイル内のデータへのアクセス制限

Oracle Solaris はマルチユーザー環境なので、ファイルシステムのセキュリティーは、システムのもっとも基本的なセキュリティーリスクです。ファイルの保護には、従来の UNIX のファイル保護と、より確実なアクセス制御リスト (ACL) との両方が使用できます。

あるユーザーには一部のファイルの読み取りを許可したり、別のユーザーには一部のファイルを変更または削除するアクセス権を付与したりできます。一方、あるデータを、どのユーザーからも読み取られないよう設定することもできます。『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 第 1 章, 「ファイルアクセスの制御」では、ファイルアクセス権の設定方法について説明されています。

`setuid` 実行可能ファイルの制限

実行可能ファイルがセキュリティーリスクとなる場合があります。いくつかの実行可能プログラムは引き続き、正しく機能するには `root` として実行する必要があります。これらの `setuid` プログラムは、ユーザー ID が `0` に設定された状態で実行されます。このようなプログラムはだれが実行したとしても `root` ID で実行されます。`root` ID で動作するプログラムは、プログラムがセキュリティーを念頭に置いて作成されていない限り、セキュリティーの問題をはらんでいます。

Oracle Solaris が `setuid` ビットを `root` に設定して提供する実行可能プログラムを除き、`setuid` プログラムの使用を禁止することをお勧めします。`setuid` プログラムの使用を禁止できない場合は、その使用を制限する必要があります。しっかりした管理を行うためには `setuid` プログラムの数を少なくする必要があります。

詳細は、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 「実行可能ファイルを原因とするセキュリティーへの悪影響を防止する」を参照してください。手順については、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 「セキュリティーリスクのあるプログラムからの保護」を参照してください。

リソース管理機能の使用

Oracle Solaris ソフトウェアには、精巧なリソース管理機能があります。これらの機能を使用することで、サーバー統合環境内のアプリケーションによるリソース利用の割り当て、スケジュール、モニター、上限設定などを行うことができます。リソース制

御フレームワークにより、プロセスが使用するシステムリソースを制限できます。このような制約を行うことで、システムリソースを混乱させようとするスクリプトによるサービス拒否攻撃を防ぎやすくなります。

これらのリソース管理機能により、特定のプロジェクトに対してリソースを指定できます。また、使用できるリソースを動的に調整することもできます。詳細は、『Oracle Solaris 12 でのリソースの管理』を参照してください。

Oracle Solaris ゾーンの使用

Oracle Solaris ゾーンは、単一の Oracle Solaris OS インスタンス内に存在するほかのシステムからプロセスが分離されるアプリケーション実行環境です。この分離を行うことで、1 つのゾーン内で稼働しているプロセスがほかのゾーンで稼働しているプロセスをモニタリングしたりそれらのプロセスに影響を及ぼしたりすることが防止されます。これは、スーパーユーザー権限によって稼働しているプロセスでも同様です。

Oracle Solaris ゾーンは、単一のサーバー上にアプリケーションを複数配置する環境に適しています。詳細は、『Oracle Solaris ゾーンの紹介』を参照してください。

システムの使用状況の監査

システム管理者は、システムの動作をモニターする必要があります。次のことを含めて、コンピュータシステムのすべての側面に気を配る必要があります。

- 通常の負荷はどの程度か
- 誰がシステムへのアクセス権を持っているか
- 各ユーザーはいつシステムにアクセスするか
- システムでは通常どのようなプログラムを実行するか

このような情報を把握していれば、ツールを使用してシステムの使用状況を監査し、各ユーザーのアクティビティーをモニターできます。セキュリティー侵害と思われる場合は、モニタリング作業が特に役立ちます。監査サービスの詳細は、『Managing Auditing in Oracle Solaris 11.4』 の 第 1 章, 「About Auditing in Oracle Solaris」を参照してください。

コンプライアンスのモニタリング

システム管理者は、システムがサイトのセキュリティー要件に準拠していることをモニターする必要があります。`compliance` コマンドでは、1 つ以上のシステムがそのシ

ステムに設定されているセキュリティープロファイルに準拠していることを確認できます。詳細は、『Oracle Solaris 12 セキュリティーコンプライアンスガイド』を参照してください。28 ページの「ファイルの整合性のモニタリング」で説明されているように、コンプライアンスでは BART よりも完全なチェックが提供されます。

ファイルアクセスの制御

Oracle Solaris は、システムにログインしているすべてのユーザーが、ほかのユーザーに属しているファイルを読み取ることができるマルチユーザー環境です。さらに、適切なアクセス権をもっているユーザーは、ほかのユーザーに属しているファイルを使用できます。詳細は、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 第 1 章,「ファイルアクセスの制御」を参照してください。ファイルに適切なアクセス権を設定する手順については、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 「ファイルの保護」を参照してください。

ディスク上のファイルの暗号化

ほかのユーザーがアクセスできないようにすることによって、ファイルを安全に保つことができます。たとえば、`600` のアクセス権を持つファイルは、その所有者と `root` アカウントを除き、読み取ることができません。アクセス権 `700` の付いたディレクトリも同様です。ただし、ほかのだれかがユーザーパスワードや `root` パスワードを推測して発見すると、そのファイルにアクセスできます。さらに、アクセス不能なはずのファイルも、システムファイルのバックアップをオフラインメディアにとるたびに、バックアップテープ上に保存されます。保護を強化するために、ディスク上の暗号化または暗号化フレームワークのコマンドを使用できます。

ZFS ファイルシステムの詳細は、『Managing ZFS File Systems in Oracle Solaris 11.4』の 「Encrypting ZFS File Systems」を参照してください。

暗号化フレームワークは、`digest`、`mac`、および `encrypt` コマンドを提供します。通常のユーザーは、これらのコマンドを使用してファイルやディレクトリを保護することができます。詳細は、『Managing Encryption and Certificates in Oracle Solaris 11.4』 の 第 1 章,「About Cryptographic Providers in Oracle Solaris」を参照してください。

アクセス制御リストの使用

ACL (「アクル」と読む) では、ファイルアクセス権の制御をより強化できます。ACL は、従来の UNIX ファイル保護機能では不十分な場合に追加で使用します。従来の

UNIX ファイル保護機能は、 所有者、グループ、その他のユーザーという 3 つのユーザークラスに読み取り権、書き込み権、実行権を提供します。ACL では、ファイルセキュリティーを管理するレベルがさらに詳細になります。

ACL を使用すると、次に示すような、きめ細かいファイルアクセス権を定義できます。

- 所有者のファイルアクセス権
- 所有者のグループのファイルアクセス権
- 所有者のグループに属していないユーザーのファイルアクセス権
- 特定ユーザーのファイルアクセス権
- 特定グループのファイルアクセス権
- 以上のカテゴリそれぞれのデフォルトアクセス権

アクセス制御リスト (ACL) で ZFS ファイルを保護するには、『Securing Files and Verifying File Integrity in Oracle Solaris 11.4』 の 「Setting ACLs on ZFS Files」を参照してください。レガシーファイルシステムでの ACL の使用については、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 「アクセス制御リストによる UFS ファイルの保護」を参照してください。

システム間でのファイルの共有

ネットワークファイルサーバーは、どのファイルを共有できるかを制御できます。また、共有ファイルにアクセスできるクライアント、およびそれらのクライアントに許可するアクセス権の種類も制御します。ファイルサーバーは、すべてのクライアントまたは特定のクライアントに、読み取り権と書き込み権、または読み取り専用アクセス権を与えることができます。アクセス制御は、`share` コマンドでリソースを利用可能にするときに指定します。

ZFS ファイルシステムの NFS 共有を作成すると、共有を削除するまでファイルシステムは永続的に共有されます。システムをリブートすると、SMF は共有を自動的に管理します。詳細は、『Managing ZFS File Systems in Oracle Solaris 11.4』 の 「Oracle Solaris ZFS Features」を参照してください。

共有ファイルへの `root` アクセスの制限

通常、スーパーユーザーは、ネットワーク上で共有されるファイルシステムには `root` としてアクセスできません。NFS システムは、要求者のユーザーをユーザー ID `60001` を持つユーザー `nobody` に変更することによって、マウントされているファイルシステムへの `root` アクセスを防止します。ユーザー `nobody` のアクセス権は、公共ユー

ザーに与えられているアクセス権と同じです。つまり、ユーザー nobody のアクセス権は資格をもたないユーザーのものと同じです。たとえば、ファイルの実行権しか公共に許可していなければ、ユーザー nobody はそのファイルを実行することしかできません。

NFS サーバーは、共有ファイルシステムへの root アクセスをホスト単位で与えることができます。これらの特権を付与するには、share コマンドの root=*hostname* オプションを使用します。このオプションは慎重に使用してください。NFS でのセキュリティーオプションについては、『Managing Network File Systems in Oracle Solaris 11.4』の 第 5 章,「Commands for Managing Network File Systems」を参照してください。

ファイルへのラベルの割り当て

Oracle Solaris では、ラベルを使用してソフトウェアで企業のセキュリティーポリシーを適用するシステムを構成できます。提供されているラベルを使用するか、または組織のセキュリティーフレーズ (「Confidential - Internal Only」など) を表示するようにラベルをカスタマイズできます。Oracle Solaris ラベルポリシーを使用すると、これらのラベルを機密データを含む既存のファイルシステムまたは新しいファイルシステムに割り当て、一連の信頼できるユーザーに、それらのユーザーの認可上限に基づいてファイルにアクセスする機能を割り当てることができます。

詳細は、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 第 2 章,「データ損失保護のためのファイルのラベル付け」を参照してください。

ほかのファイルセキュリティーオプションについては、26 ページの「ファイルアクセスの制御」を参照してください。

ファイルの整合性のモニタリング

システム管理者は、管理対象のシステムにインストールされたファイルが予想外の方法で変更されないことを保証する必要があります。大規模インストールでは、各システム上のソフトウェアスタックの比較や報告を行うツールを使用すればシステムの追跡、記録が行えます。基本監査報告機能 (BART) を使用すると、一定期間にわたって1 つ以上のシステムをファイルレベルでチェックし、システムを包括的に検証できます。一定期間にわたってすべてのシステムまたは 1 つのシステムにおける BART目録の変化を調べることで、システムの整合性を検証できます。BART には、目録作成機能、目録比較機能、レポート生成規則などが用意されています。詳細は、『Oracle Solaris 12 でのファイルのセキュリティー保護とファイル整合性の検証』 の 第 3 章,「BART を使用したファイル整合性の検証」を参照してください。

デバイスアクセスの制御

コンピュータシステムに接続された周辺機器は、セキュリティーリスクをもたらします。たとえば、マイクは会話をキャッチし、その会話をリモートシステムに送信します。CD-ROM の場合、その情報を CD-ROM に残して、CD-ROM デバイスを次に使うユーザーが読み取れるようにすることができます。プリンタは、リモートサイトからもアクセスできます。システムの必須デバイス (たとえば、`bge0` などのネットワークインタフェース) もまた、セキュリティー問題を引き起こす可能性があります。

Oracle Solaris ソフトウェアには、デバイスへのアクセスを制御するための方法がいくつか用意されています。

- **デバイスポリシーを設定する** – 特定のデバイスにアクセスしているプロセスが特定の特権セットで実行されるように要求できます。それらの権限を持たないプロセスは、そのデバイスを使用できません。ブート時に、Oracle Solaris ソフトウェアはデバイスポリシーを構成します。サードパーティーのドライバは、そのインストール時にデバイスポリシーを構成できます。インストール後、管理者はデバイスポリシーをデバイスに追加できます。
- **デバイスを割り当て可能にする** – ユーザーがデバイスを使用する前に割り当てる必要があるように要求できます。割り当てによって、デバイスの使用が一度に 1 人のユーザーに制限されます。さらに、ユーザーがそのデバイスの使用を承認されていることを要求できます。
- **デバイスの使用を防ぐ** – コンピュータシステム上のどのユーザーも特定のデバイス (マイクなど) を使用できないように設定できます。たとえば、ある種のデバイスを使用できないようにする例としては、コンピュータキオスクが挙げられます。
- **デバイスを特定のゾーンに限定する** – デバイスの使用を非大域ゾーンに割り当てることができます。詳細は、『Creating and Using Oracle Solaris Zones』 の 「Device Use in Non-Global Zones」を参照してください。

デバイスポリシー

デバイスポリシーメカニズムを使用することで、デバイスを開こうとするプロセスに特定の権限を要求するように指定できます。デバイスポリシーによって保護されたデバイスをアクセスできるのは、デバイスポリシーで指定されている権限で稼働しているプロセスだけです。Oracle Solaris はデフォルトのデバイスポリシーを提供します。たとえば、`bge0` などのネットワークインタフェースでは、そのインタフェースにアクセスするプロセスが `net_rawaccess` 特権で実行されていることが必要です。この要件はカーネルで適用されます。特権の詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Process Rights Management」を参照してください。

Oracle Solaris では、デバイスはファイルアクセス権とデバイスポリシーで保護されます。たとえば、`/dev/ip` ファイルのアクセス権は `666` です。しかし、このデバイスは適切な権限を持つプロセスによってしかオープンできません。

デバイスポリシーの構成は監査の対象とすることができます。デバイスポリシーの変更は、`AUE_MODDEVPLCY` 監査イベントによって記録されます。

デバイスポリシーの詳細は、次のページを参照してください。

- 表5
- 89 ページの「デバイスポリシーコマンド」
- 『Securing Users and Processes in Oracle Solaris 11.4』 の 「Privileges and Devices」

デバイスの割り当て

デバイス割り当てメカニズムを使用すれば、CD-ROM などの周辺機器に対するアクセスを制限できます。デバイス割り当てが有効になっていない場合、周辺機器の保護はファイルアクセス権によってのみ行われます。たとえば、デフォルトでは周辺機器は次のように使用できます。

- CD-ROM ドライブまたはディスクの読み取りと書き込みは、任意のユーザーが行うことができます。
- すべてのユーザーがマイクを接続できます。
- すべてのユーザーが接続されたプリンタにアクセスできます。

デバイス割り当てを行うことで、承認されたユーザーにだけデバイスの使用を限定できます。デバイス割り当てによって、デバイスアクセスを完全に防ぐこともできます。デバイスを割り当てるユーザーは、そのユーザー自身が割り当てを解除するまでそのデバイスを独占的に使用できます。デバイスの割り当てが解除される際には、残っているすべてのデータがデバイスクリーンスクリプトによって消去されます。デバイスにスクリプトがない場合には、デバイスクリーンスクリプトを作成してそのデバイスから情報を一掃できます。この例は、96 ページの「新しいデバイスクリーンスクリプトの作成」を参照してください。

デバイス割り当てに関連した試み (デバイスの割り当て、デバイスの割り当て解除、割り当て可能なデバイスの一覧表示) は、監査の対象とすることができます。監査イベントは、`other` 監査クラスの一部です。

デバイス割り当てについての詳細は、次を参照してください。

- 表6
- 89 ページの「デバイスの割り当て」
- 91 ページの「デバイス割り当てコマンド」

ネットワークアクセスの制御

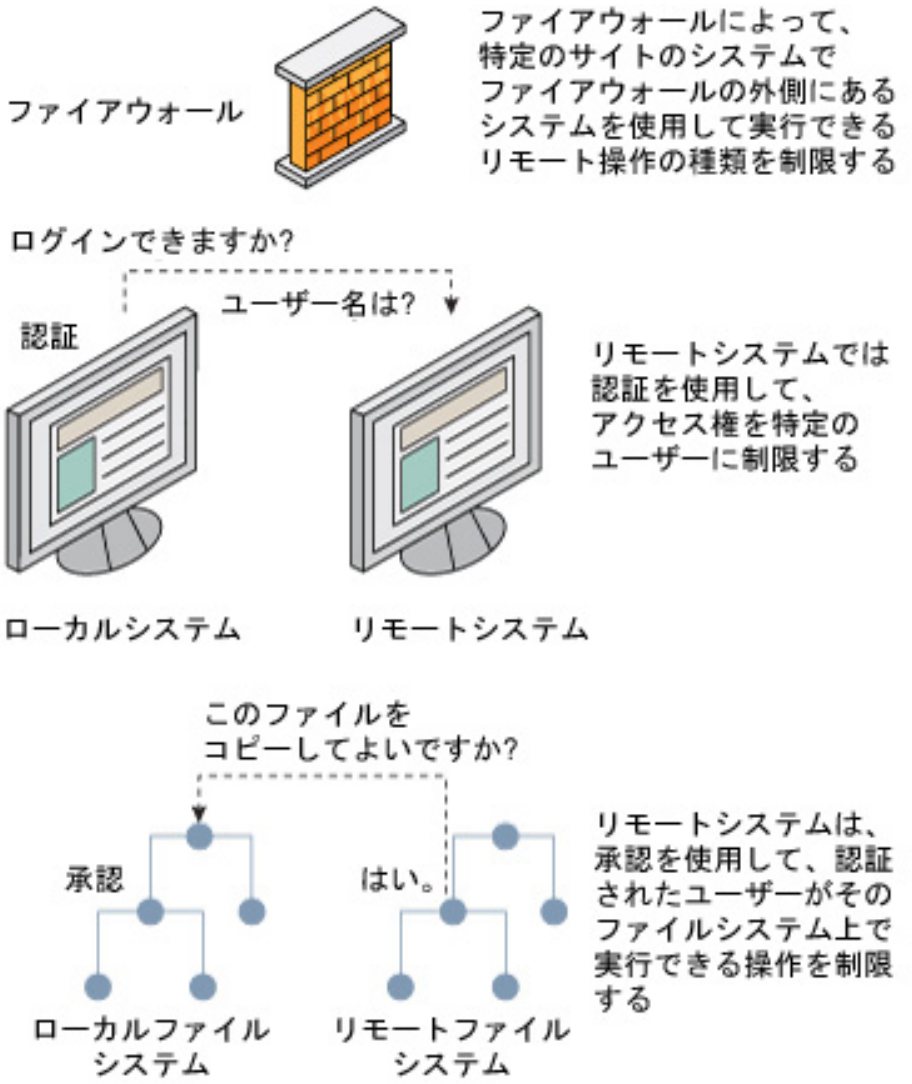
多くの場合、コンピュータは、接続されたコンピュータ間で情報を交換できるコンピュータネットワークの一部になっています。さらに、ネットワークに接続されたコンピュータは、ネットワーク上のほかのコンピュータにあるデータなどのリソースにアクセスできます。コンピュータをネットワーク化するとコンピューティング環境の処理能力と性能が向上しますが、ネットワークのコンピュータセキュリティーが複雑になります。

たとえば、コンピュータのネットワーク内では、個々のシステムは情報を共有できます。未承認アクセスがセキュリティーリスクとなります。多くの人々がネットワークにアクセスするので、(特にユーザーエラーを通して) 未承認アクセスが発生する可能性も大きくなります。また、パスワードの不適切な扱いも未承認アクセスの原因となります。

ネットワークセキュリティーメカニズム

一般にネットワークセキュリティーは、リモートシステムからの操作の制限またはブロックに基づきます。次の図は、リモート操作に適用できるセキュリティー制限を示します。

図 1 リモート操作のセキュリティー制限



リモートアクセスの認証と承認

認証は、ユーザーがリモートシステムへのアクセスを試みる際にアクセスを制御する方法です。認証は、システムレベルでもネットワークレベルでも設定できます。ユーザーがリモートシステムにアクセスすると、「承認」という方法でそのユーザーが実行できる操作が制限されます。次の表は、認証と承認を提供するサービスを示したものです。

表 3 リモートアクセスのための認証サービス

サービス	説明	詳細情報
IPsec	IPsec は、ホストに基づく認証および認可に基づく認証と、ネットワークトラフィックの暗号化を行います。	『Securing the Network in Oracle Solaris 11.4』 の 第 6 章, 「About IP Security Architecture」
Kerberos	Kerberos は、システムにログインしているユーザーの認証と承認を暗号化を通して行います。	例については、 『Managing Kerberos in Oracle Solaris 11.4』 の 「How the Kerberos Service Works」 を参照してください。
LDAP	LDAP ディレクトリサービスは、認証と承認の両方をネットワークレベルで提供できます。	『Oracle Solaris 12 ディレクトリサービスとネームサービスでの作業: DNS と NIS』
リモートログインコマンド	リモートログインコマンドを使用すると、ユーザーはネットワーク経由でリモートシステムにログインし、そのリソースを使用できます。`ssh` コマンドはデフォルトで有効になっています。信頼されるホストの場合、認証は自動です。それ以外の場合は、自分自身を認証するように求められます。	『Oracle Solaris 12 でのリモートシステムの管理』 の 第 3 章, 「リモートシステムへのアクセス」
SASL	簡易認証セキュリティー層 (SASL) は、ネットワークプロトコルに認証サービスとセキュリティーサービス (オプション) を提供するフレームワークです。プラグインによって、適切な認証プロトコルを選択できます。	『Managing Authentication in Oracle Solaris 11.4』 の 「About SASL」
Secure NFS	MIT Kerberos V は、通信を保護することによって整合性、プライバシ、および認証に関してセキュアな NFS 環境をサポートします。	『Oracle Solaris 12 での Kerberos の管理』
Secure Shell	Secure Shell は、セキュアでないネットワークを経由したネットワークトラフィックを暗号化します。Secure Shell は、パスワード、公開鍵、またはこの両方の使用による認証を提供します。	『Managing Secure Shell Access in Oracle Solaris 11.4』 の 「About Secure Shell」

Oracle Solaris の特権ポートメカニズムは、Secure Shell 通信を保護できます。特権ポートには、`1024` 未満のポート番号が割り当てられます。クライアントシステムは、クライアントの資格を認証したあと、特権ポートを使用してサーバーへの接続を設定し

ます。次に、サーバーは接続のポート番号を検査してクライアントの資格を検証します。

Oracle Solaris ソフトウェアを使用していないクライアントは、特権ポート上で通信できないことがあります。クライアントが特権ポートを使って通信できない場合は、次のようなエラーメッセージが表示されます。

```
"Weak Authentication
NFS request from unprivileged port"
```

ファイアウォールシステム

ファイアウォールシステムを設定すると、ネットワーク内のリソースを外部のアクセスから保護できます。「ファイアウォールシステム」は、内部ネットワークと外部ネットワークの間のバリアとして機能するセキュリティー保護ホストです。内部ネットワークは、ほかのネットワークを「信頼できる状態でない」ものとして扱います。内部ネットワークと、インターネットなどの外部ネットワークとの間に、このような設定を必ず行うようにしてください。

ファイアウォールはゲートウェイとしても機能しますし、バリアとしても機能します。ゲートウェイとしては、ネットワーク間でデータを通過させます。バリアーとしては、データのネットワークとの間の自由な通過をブロックします。内部ネットワーク上のユーザーがリモートネットワーク上のホストシステムにアクセスするには、ファイアウォールシステムにログインする必要があります。また、外部ネットワーク上のユーザーは、内部ネットワーク上のホストシステムにアクセスする前に、まずファイアウォールシステムにログインしなければなりません。

ファイアウォールは、一部の内部ネットワーク間でも有効です。たとえば、パケットの転送をアドレスまたはプロトコルで制限するには、ファイアウォールまたはセキュアなゲートウェイコンピュータを設定できます。これにより、メールを転送するためのパケットを許可するが、`ftp` コマンドのパケットは許可しないようにできます。

さらに、内部ネットワークから送信されるすべての電子メールは、まずファイアウォールシステムに送信されます。ファイアウォールは、このメールを外部ネットワーク上のシステムに転送します。ファイアウォールシステムは、すべての着信電子メールを受信して内部ネットワーク上のシステムに配信するという役割も果たします。

注意 - ファイアウォール上で厳格かつ強固に適用されたセキュリティーを維持している場合でも、ネットワーク上のその他のシステムでセキュリティーを緩くすれば、ファイアウォールシステムを突破できる侵入者は、内部ネットワーク上のその他のすべてのシステムへのアクセスを取得できる可能性があります。

ファイアウォールシステムには、信頼されるホストを配置しないでください。信頼されるホストとは、ユーザーがログインするときに、パスワードを入力する必要がない

ホストシステムのことです。ファイアウォールシステムでは、ファイルシステムを共有しないでください。また、ほかのサーバーのファイルシステムをマウントしないでください。

Oracle Solaris の IPsec およびパケットフィルタ機能は、ファイアウォール保護を提供できます。ネットワークトラフィックの保護の詳細は、『Oracle Solaris 12 でのネットワークのセキュリティー保護』を参照してください。

暗号化システムとファイアウォールシステム

ネットワーク外部からの未承認ユーザーは、宛先に到達する前にパケットを捕捉し、元の経路にパケットを戻す前に任意のデータを内容に挿入することで、パケット内のデータを破損させたり、破棄したりできます。この方法は、「パケットスマッシング」と呼ばれます。

ローカルエリアネットワーク上では、パケットはサーバーを含むすべてのシステムに同時に到達するので、パケットスマッシングは不可能です。ただし、ゲートウェイ上ではパケットスマッシングが可能なため、ネットワーク上のすべてのゲートウェイを保護する必要があります。

もっとも危険なのは、データの完全性に影響するような攻撃です。このような攻撃を受けると、パケットの内容が変更されたり、ユーザーが偽装されたりします。

その他の攻撃でも盗聴が伴う可能性がありますが、データの整合性が損なわれたり、ユーザーが偽装されたりすることはありません。盗聴者は、会話を記録して、あとで再生します。盗聴攻撃によってデータの完全性が損なわれることはありませんが、プライバシが侵害されます。ネットワーク上でやりとりされるデータを暗号化すると、重要な情報のプライバシを保護できます。

- セキュリティー保護されていないネットワーク経由のリモート操作を暗号化する方法については、『Managing Secure Shell Access in Oracle Solaris 11.4』 の 第 1 章, 「Using Secure Shell」を参照してください。
- ネットワーク内のデータを暗号化および認証する方法については、『Managing Kerberos in Oracle Solaris 11.4』 の 第 1 章, 「Kerberos on Oracle Solaris」を参照してください。
- IP データグラムを暗号化する方法については、『Securing the Network in Oracle Solaris 11.4』 の 第 6 章, 「About IP Security Architecture」を参照してください。

セキュリティー問題の報告

会社で重大なセキュリティー違反が発生した疑いがある場合は、Computer Emergency Response Team/Coordination Center (CERT/CC) に連絡してください。CERT/CC は、

Defense Advanced Research Projects Agency (DARPA) の資金提供を受けたプロジェクトで、カーネギメロン大学の Software Engineering Institute にあります。CERT/CC はセキュリティー問題の解決を支援できます。また、特定のニーズに合った他の Computer Emergency Response Team を紹介することもできます。最新の連絡先情報については、CERT/CC (`https://www.sei.cmu.edu/about/divisions/cert/index.cfm`) Web サイトを参照してください。

♦ ♦ ♦ 第 2 章

2 Oracle Solaris システムの整合性の保護

未承認のカーネルモジュール、トロイの木馬アプリケーション、およびシステムにロードされるその他の脅威から Oracle Solaris システムを保護できます。この章では、このような脅威からの保護を提供し、システム全体の整合性を保持する Oracle Solaris のセキュリティー機能について説明します。

この章の内容は次のとおりです。

ベリファイドブートの使用

悪質のあるプログラムは、サードパーティーに情報を渡したり、Oracle Solaris の動作を変更したりする可能性があります。一般に、サードパーティーモジュールに悪意がなくても、サイトの変更を制御するポリシーに違反している可能性があります。したがって、このようなモジュールが承認なしでインストールされることからシステムを保護する必要もあります。

Oracle Solaris のベリファイドブートによって、システムのブートプロセスがセキュリティー保護されます。この機能は有効にする必要があり、それによってシステムは次のような脅威から保護されます。

- カーネルモジュールの破損
- 正当なカーネルモジュールになりすました悪意のあるプログラム (トロイの木馬ウイルス、スパイウェア、ルートキットなど) の挿入または置換
- 未承認のサードパーティーカーネルモジュールのインストール

ベリファイドブートを使用するには、ファームウェアのアップグレードが必要になる場合があります。詳細は、38 ページの「ベリファイドブートのためのファームウェアアップグレード」を参照してください。

ベリファイドブートを次の構成で、次のツールを使用して有効にできます。

- Oracle Solaris SPARC システム – 40 ページの「ベリファイドブートのポリシー」を参照してください。
- x86 の UEFI セキュアブート (BIOS メニュー) – セキュアブートの構成については、使用しているプラットフォームの説明を参照してください。
- Oracle Solaris カーネルゾーン – 『Oracle Solaris カーネルゾーンの作成と使用』 の「Using Verified Boot to Secure an Oracle Solaris Kernel Zone」を参照してください。
- 論理ドメイン (LDOM) – 『Oracle VM Server for SPARC 3.4 管理ガイド』の「ベリファイドブートの使用」を参照してください。
- Oracle Integrated Lights Out Manager (ILOM) – 『*Oracle ILOM* 構成および保守用管理者ガイドファームウェア *Release 3.2.x*』の「SPARC 検証済みブートプロパティーの構成」を参照してください。

SPARC: ベリファイドブートのためのファームウェアアップグレード

SPARC ファームウェアは工場でインストールされます。一部の SPARC プラットフォームで、拡張ブートブロックにはベリファイドブート機能との互換性がありません。ベリファイドブートのレベルが `enforce` に設定されている場合、システムはブートせず、次のようなメッセージが表示されます。

```
WARNING: Total size of bootblk fcode greater than expected
FATAL: Bootblk fcode extraction failed
Missing cmn-xxx[ caused cmn-append with 'verified boot policy =
enforce, halting boot' argument to fail
```

ベリファイドブートのレベルが `warning` に設定されている場合は、次のようなメッセージが表示されます。

```
WARNING: Bootblk fcode extraction failed
WARNING: Signature verification of boot-archive bootblk failed
```

ベリファイドブートが有効になっている SPARC システムでは、次のようにファームウェアを更新してください。

- T5 シリーズ、M5 シリーズ、および M6 シリーズ – ファームウェア 9.6.5 以降にアップグレードします。
- T7 シリーズおよび M7 シリーズ – ファームウェア 9.7.1 以降にアップグレードします。
- Fujitsu M10 – XCP 2320 以降にアップグレードします。

ファームウェアを更新するには、`fwupdate`(1M) のマニュアルページを参照してください。

ベリファイドブートと ELF 署名

Oracle Solaris では、ブート検証は `elfsign` の署名または鍵を使用して実行されます。Oracle Solaris カーネルモジュールは、工場でこれらの鍵を使用して署名されます。ファイル形式から、これらのモジュールは ELF オブジェクトとも呼ばれます。署名は、オブジェクトファイルで選択した ELF レコードの SHA-256 チェックサムを使用して作成されます。SHA-256 チェックサムは、RSA-2048 の非公開鍵と公開鍵のペアを使用して署名されます。公開鍵は `/etc/certs/elfsign` ディレクトリで配布されていますが、非公開鍵は配布されていません。

すべての鍵は、システムのブート前環境に格納されています。これは、Oracle Solaris をブートする前に実行されるソフトウェアまたはファームウェアです。このファームウェアは、`platform/.../unix` をロードおよびブートします。

ブート前環境は、次のように SPARC システムのカテゴリごとに異なります。

- Oracle Integrated Lights Out Manager (ILOM) のベリファイドブートがサポートされている SPARC システム - 鍵および構成設定は ILOM に格納されます。

  Oracle ILOM はオペレーティングシステムのファイルシステム外部にあるため、ベリファイドブートの構成は、オペレーティングシステムのユーザー (管理者 (root) 特権を持つユーザーを含む) による改ざんから保護されます。したがって、このシステムカテゴリでは、ベリファイドブートがよりセキュアです。

  ベリファイドブートの構成が承認なしで変更されることを回避するには、ILOM へのアクセスがセキュアであることを確認する必要があります。ILOM のセキュリティー保護の詳細は、「システム管理および診断 (`http://www.oracle.com/goto/ilom/docs`)」にあるドキュメントを参照してください。
- SPARC M5 シリーズ、SPARC M6 シリーズ、および SPARC T5 シリーズ - 構成設定はシステムの ILOM に格納されます。SPARC ファームウェアが構成情報を Oracle Solaris に送信します。
- Fujitsu SPARC M12 および Fujitsu M10 システム - 構成設定は、システムの XSCF に格納されます。Fujitsu SPARC M12 および Fujitsu M10 XSCF ファームウェアは、ベリファイドブートや証明書の有効化に関するポリシーなどの構成情報を Oracle Solaris に送信します。OpenBoot (OBP) は、Oracle Solaris システムをブートする前にこの構成情報を読み取ります。

  Fujitsu SPARC M12 システムのすべての XCP ファームウェアはベリファイドブートをサポートしています。ベリファイドブートの構成の詳細は、次のガイドを参照してください。

  - *Fujitsu SPARC M12* および *M10/SPARC M10* システム運用・管理ガイド
  - *Fujitsu M10/SPARC M10* システム プロダクトノート – Fujitsu M10 システムのベリファイドブートをサポートする XCP ファームウェアバージョン向け

システムブート時の検証シーケンス

ベリファイドブートによって、Oracle Solaris カーネルモジュールの `elfsign` 署名の検証が自動化されます。管理者はベリファイドブートを使用することで、システムのリセットからブートプロセスの完了までのブートプロセスに、検証可能な信頼チェーンを作成できます。

システムのブート中に、ブートプロセスで開始されたコードの各ブロックで、次にロードする必要があるブロックが検証されます。検証およびロードのシーケンスは、最後のカーネルモジュールがロードされるまで続行されます。

あとでシステムで電源の再投入が実行されるときに、新しい検証シーケンスが開始されます。管理者は、検証に失敗したときに適切なアクションが行われるように、ベリファイドブートを構成することもできます。

SPARC システムでの Oracle Solaris のブートフローを次に示します。

`Firmware -> Bootblock -> /platform/.../unix -> genunix ->` *other kernel modules*

このファームウェアは、初期の Oracle Solaris モジュールである Oracle Solaris の `/platform/.../unix` モジュールを検証してから、ロードします。同様に、`unix` モジュールの一部である Oracle Solaris カーネルの実行時ローダー `krtld` は、汎用の UNIX (`genunix`) モジュールおよび後続のモジュールを検証し、ロードします。

ベリファイドブートのポリシー

このリリースでは、ベリファイドブートには `boot_policy` という 1 つのポリシープロパティーしかありません。`boot_policy` プロパティーは、ブートプロセス中にカーネルモジュールをロードするときにベリファイドブート動作を管理します。

レガシー SPARC システムおよび x86 システム上では、`boot_policy` プロパティーは `/etc/system` ファイル内に定義されています。Oracle ILOM のベリファイドブートがサポートされている SPARC システムでは、`boot_policy` は `/HOST`*n*`/verified_boot` 内にある ILOM のプロパティーで、ここで *n* は物理ドメイン (PDomain) 番号です。

`boot_policy` プロパティーは、次の値のいずれかを使用して構成できます。

- `none` – ブート検証が実行されません。これはデフォルトです。
- `warning` – モジュールがロードされる前に、各カーネルモジュールの `elfsign` 署名が検証されます。モジュールの検証に失敗した場合でも、モジュールはロードされます。不一致は、システムコンソールまたはシステムログ (使用可能な場合) に記録されます。デフォルトのログは `/var/adm/messages` です。

- `enforce` – モジュールがロードされる前に、各カーネルモジュールの `elfsign` 署名が検証されます。モジュールの検証に失敗した場合は、モジュールがロードされません。不一致は、システムコンソールまたはシステムログ (使用可能な場合) に記録されます。デフォルトのログは `/var/adm/messages` です。

注記 - デフォルトでは、3.4 より前の Oracle VM Server for SPARC のバージョンで作成された論理ドメインは `boot-policy=warning` を設定します。カーネルモジュールが署名されていないか破損している場合、この設定ではサーバーの更新後のドメインのブート時に、警告メッセージが発行されます。

ベリファイドブートの公開鍵証明書

ベリファイドブートは、次のソースにある公開鍵証明書を使用します。

- `/etc/certs/elfsign` ディレクトリ

  イメージにサードパーティーベンダーによって署名された ELF オブジェクトが含まれている場合は、このディレクトリにベンダーの証明書を追加する必要があります。
- `zoneadm` コマンドによって追加されたカーネルゾーン
- x86 の UEFI セキュアブート (BIOS メニュー)
- SPARC の Oracle ILOM

  ベリファイドブートをサポートしている SPARC の Oracle ILOM では、事前にインストールされたベリファイドブートの証明書ファイル `/etc/certs/elfsign/ORCLS11SE` が提供されています。証明書には、Oracle Solaris で署名された ELF オブジェクトの `elfsign` 署名を検証する際に使用される RSA 公開鍵が含まれています。すべての証明書は個別の PDomain にロードされ、管理されます。
- ILOM の構文は、ハードウェアプラットフォームおよびファームウェアバージョンによって異なります。Oracle ILOM を使用して証明書を構成するには、『Oracle® ILOM 構成および保守用管理者ガイドファームウェア Release 3.2.x』の「SPARC 検証済みブートプロパティーの構成」を確認してください。

カーネルモジュールの署名を手動で検証することもできます。手動での検証は、診断時に正しい署名が存在していることを確認するために役立ちます。

例 1 カーネルモジュールの署名の手動検証

次のように、`elfsign verify -v` *kernel_module* コマンド構文を使用します。

```
$ elfsign verify -v /kernel/misc/sparcv9/bignum
elfsign: verification of /kernel/misc/sparcv9/bignum passed.
Elfsign signature format: rsa_sha256
Signer: O=Oracle Corporation, OU=Corporate Object Signing, OU=Solaris Signed Execution,
 CN=Solaris 11
```

Trusted Platform Module の使用

Trusted Platform Module (TPM) は、システムに固有の暗号化済み構成情報が格納されるデバイスおよび実装を表します。情報は、システムのブート中にプロセスの測定に使用するメトリックとしての役割を果たします。TPM は、PKCS #11 ライブラリおよび `pktool` コマンドを使用してアクセスできるセキュアなハードウェアキーストアとしての役割を果たします。

Oracle Solaris では、次のコンポーネントに TPM が実装されています。

- TPM デバイスドライバは TPM デバイスと通信します。
- Trusted Computing Group (TCG) Software Stack (TSS) は、`tcsd` デーモンを使用した TPM デバイスとの通信チャネルとして機能します。
- PKCS #11 ライブラリには、TPM を使用して鍵を生成し、機密操作を実行するハードウェアトークンまたはプロバイダが実装されています。プロバイダでは、TPM デバイス内部でのみ使用可能な鍵で暗号化することで、すべてのプライベートデータオブジェクトが保護されます。PKCS #11 ライブラリは、RSA Security Inc. PKCS #11 Cryptographic Token Interface (Cryptoki) 標準に準拠しています。
- ブートプロセスの検証で TPM 関連の側面を管理するために、`tpmadm` コマンドが使用されます。

  詳細は、`tpmadm(8)` のマニュアルページを参照してください。

プラットフォーム所有者は、特権操作を承認する際に使用される所有者のパスワードを設定することで、TPM を初期化する必要があります。プラットフォーム所有者は TPM 所有者とも呼ばれます。従来のスーパーユーザーと異なる点は、次の 2 つです。

- TPM 機能にアクセスするために、プロセス特権は必要ありません。呼び出し元プロセスの特権レベルに関係なく、特権操作では所有者のパスワードを把握しておくことが必要です。
- TPM 所有者は、TPM 鍵で保護されたデータのアクセス制御をオーバーライドできません。所有者は TPM を再初期化することで、データを効率的に破棄できます。ただし、所有者は、その他のユーザーが所有する TPM 鍵で暗号化されたデータにはアクセスできません。

このガイドで説明したその他の方法とともに Trusted Platform Module を使用すると、ユーザーまたはアプリケーションによる未承認アクセスからシステムがセキュリティー保護されます。

Oracle Solaris システムでの TPM の初期化とバックアップ

このセクションには、Oracle Solaris システムで TPM を初期化したり、TPM データおよび鍵をバックアップしたりするための手順が含まれています。SPARC システムと

x86 システムとでは、手順が異なります。ただし、TPM を初期化するための特定の前提条件は、両方のプラットフォームで共通です。

- システムに TPM デバイス `/dev/tpm` をインストールする必要があります。
- TPM では TCG Trusted Platform Module 仕様バージョン 1.2 (別名 ISO/IEC 11889-1:2009) が使用されている必要があります。`https://trustedcomputinggroup.org//tpm-main-specification/` で公開されている仕様を参照してください。
- 次の Oracle Solaris TPM パッケージがインストールされている必要があります。
  - Trusted Platform Module ドライバ (`driver/crypto/TPM`)
  - TrouSerS TCG ソフトウェア (`library/security/trousers`)

  これらのパッケージをインストールするには、次のコマンドを使用します。

```
# pkg install driver/crypto/tpm
# pkg install library/security/trousers
```

▼ TPM デバイスがオペレーティングシステムで認識されているかどうかを確認する方法

この手順を使用して、インストールされている TPM デバイスが Oracle Solaris で認識されているかどうかを確認します。この手順は、SPARC システムと x86 システムの両方に適用されます。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

● **端末ウィンドウで、次のコマンドを発行します。**

```
# prtconf -v |grep tpm
```

TPM デバイスが認識されている場合は、コマンドで次のような出力が生成されます。

```
# prtconf -v |grep tpm
tpm, instance #0
dev_path=/pci@0,0/isa@lf/tpm@0,fed40000:tpm
dev_link=/dev/tpm
```

出力が生成されない場合は、TPM が無効になっている可能性があります。デバイスを有効にする方法については、システムのプラットフォームに応じて、44 ページの「Oracle ILOM インタフェースを使用して TPM を初期化する方法」または47 ページの「BIOS を使用して TPM を初期化する方法」を参照してください。

注記 - 代わりに `ls` コマンドを使用しても、同じ情報を取得できます。ただし、この出力に含まれる情報は、`prtconf` コマンドで提供される情報よりも少ないです。

```
# ls -l /dev/tpm
lrwxrwxrwx  1 root root  44 May 22 2012 /dev/tpm ->
../devices/pci@0,0/isa@lf/tpm@0,fed40000:tpm
```

▼ SPARC: Oracle ILOM インタフェースを使用して TPM を初期化する方法

SPARC システムで TPM を初期化するには、システムの ILOM と Oracle Solaris の両方のインタフェースを使用します。

この手順には、TPM データおよび鍵をバックアップするための手順が含まれています。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **ILOM プロンプトで、ホストシステムを停止します。**

   - **単一ホストサーバーの場合:**

     ```
     -> stop /System
     ```

   - **マルチドメインサーバーの場合:**

     ```
     -> stop /Servers/PDomains/PDomain_n/HOST
     ```

   サーバーの停止にはしばらく時間がかかることがあります。次の手順に進む前に、ホストコンソールに次のメッセージが表示されるまで待つ必要があります。

   ```
   -> SP NOTICE: Host is off
   ```

   注記 - 前のステップでホストが停止しない場合にのみ、`-f|force` オプションを追加してホストシステムを停止してください。

2. **TPM をアクティブにします。**

   SPARC システムに応じて、次のコマンドセットのいずれかを使用して TPM をアクティブにします。

   - SPARC M5 シリーズサーバーおよび SPARC T5 シリーズサーバーでは、次のコマンドを使用します。

     ```
     -> set /HOST/tpm mode=activated
     ```

- SPARC M5-32 シリーズサーバーでは、次のコマンドを使用します。

```
-> set /HOST0/tpm mode=activated
```

- SPARC T4 サーバーでは、次のコマンドを使用します。

```
-> set /HOST/tpm enable=true activate=true
-> show /HOST/tpm
```

3. **Oracle Solaris プロンプトで、TPM を初期化します。**

   TPM を初期化すると TPM 所有者となり、所有者パスワード (所有者 PIN とも呼ばれる) を割り当てる必要があります。

```
# tpmadm init
TPM Owner PIN:
Confirm TPM Owner PIN
```

4. **TPM のステータスを確認します。**

```
# tpmadm status
TPM Version: 1.2 (ATML Rev: 13.9, SpecLevel: 2, ErrataRev: 1)
TPM resources
Contexts: 16/16 available
Sessions: 2/3 available
Auth Sessions: 2/3 available
Loaded Keys: 18/21 available
Platform Configuration Registers (24)
PCR 0: E1 EE 40 D8 66 28 A9 08 B6 22 8E AF DC 3C BC 23 71 15 49 31
PCR 1: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 2: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 3: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 4: AF 98 77 B8 72 82 94 7D BE 09 25 10 2E 60 F9 60 80 1E E6 7C
PCR 5: E1 AA 8C DF 53 A4 23 BF DB 2F 4F 0F F2 90 A5 45 21 D8 BF 27
PCR 6: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 7: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 9: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 14: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 15: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 16: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 17: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 18: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 19: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 21: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 22: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 23: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

5. **システムの移行またはハードウェアの交換中に、TPM データおよび鍵を将来の使用のためにバックアップします。**

   - **Oracle Solaris がインストールされているマルチドメインシステムの場合は、TPM を含む SP ボードのフェイルオーバーを有効にします。**

```
# tpmadm failover
Enter TPM Owner PIN:
Enter PIN for the migration key:
Confirm PIN for the migration key:
```

注記 - TPM 所有者の PIN は、TPM を初期化するときに使用された PIN です。

将来のシステムの移行またはハードウェアの交換のために、移行鍵に指定した PIN を使用して TPM キーストアをバックアップおよび復元できるように、その PIN を記録しておいてください。詳細は、52 ページの「TPM フェイルオーバーオプション」および tpmadm(8) のマニュアルページを参照してください。

- **その他のすべてのプラットフォームの場合は、TPM データおよび鍵の手動バックアップを実行します。手順については、46 ページの「TPM データおよび鍵をバックアップする方法」を参照してください。**

6. **(オプション) TPM 暗号化プロバイダを有効にします。**

注記 - TPM 暗号化プロバイダは、Oracle Solaris よりも低速です。この手順は、TPM で暗号化操作を実行する場合にのみ実行してください。

```
# cryptoadm install provider='/usr/lib/security/$ISA/pkcs11_tpm.so'
# cryptoadm list -mv provider='/usr/lib/security/$ISA/pkcs11_tpm.so'
```

▼ SPARC: TPM データおよび鍵をバックアップする方法

システムをはじめてブートしたあと、将来のシステムの移行またはハードウェアの交換中に使用できるように TPM データおよび鍵をバックアップするようにしてください。

Oracle Solaris がインストールされているマルチドメインシステムの場合は、`tpmadm failover` コマンドを使用して、TPM データおよび鍵がサーバー上のスタンバイ SP に自動的にバックアップされるように指定します。そのバックアップされた TPM データおよび鍵を新しい SP 上でシステムの移行またはハードウェアの交換に使用できます。手順については、47 ページの「BIOS を使用して TPM を初期化する方法」のバックアップステップを参照してください。

その他のすべてのプラットフォームの場合は、システムの移行またはハードウェアの交換中に使用できるように、次の手順を使用して TPM データおよび鍵を手動でバックアップします。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **端末ウィンドウで、TPM が有効になっていることを確認します。**

```
# tpmadm status
```

TPM 所有者がインストールされていないと表示される場合は、TPM が初期化されていません。続行しないでください。

2. **ストレージルート鍵 (SRK) の ID を使用して、移行データをバックアップします。**

```
#  tpmadm migrate export 00000000-0000-0000-0000-00000000000b
```

その鍵に承認が必要な場合は、システムから鍵のパスワードを入力するよう求められます。また、移行鍵のパスワードの入力も求められます。

3. **`/var/tpm/system` 内の移行ファイルを探すことによって、データがバックアップされたことを確認します。**

```
# ls -l /var/tpm/system/tpm-migration.*
-rw-------   1 root  root  563 July 21 10:45 /var/tpm/system/tpm-migration.dat
-r--------   1 root  root  766 July 21 10:36 /var/tpm/system/tpm-migration.key
```

▼ x86: BIOS を使用して TPM を初期化する方法

x86 システムでは、Oracle Solaris を使用してサービスを初期化する前に、システムの BIOS で次の手順を実行します。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **端末ウィンドウで、システムをリブートします。**

```
# reboot -p
```

2. **システムのブート中に F2 キーを押して、BIOS メニューにアクセスします。**

3. **BIOS メニューオプションを使用して、TPM を構成します。**

   a. **「Advanced」 -> 「Trusted Computing」に移動します。**

   b. **次のメニュー項目の値を指定することで、TPM を設定します。**

```
TCG/TPM Support [Yes]
Execute TPM Command [Enabled]
```

   c. **Esc キーを押して、BIOS メニューを終了します。**

   d. **「Save Changes and Exit」を選択します。**

   e. **ブートプロセスを続行するには、「Ok」をクリックします。**

4. **ブートプロセスが完了したら、`tcsd` デーモンを有効にします。**

```
# svcadm enable -s svc:/application/security/tcsd
```

5. **TPM を初期化します。**

   TPM を初期化すると TPM 所有者となり、所有者パスワードを割り当てる必要があります。

```
# tpmadm init
TPM Owner PIN:
Confirm TPM Owner PIN
```

6. **TPM のステータスを確認します。**

```
# tpmadm status
TPM Version: 1.2 (ATML Rev: 13.9, SpecLevel: 2, ErrataRev: 1)
TPM resources
Contexts: 16/16 available
Sessions: 2/3 available
Auth Sessions: 2/3 available
Loaded Keys: 18/21 available
Platform Configuration Registers (24)
PCR 0: E1 EE 40 D8 66 28 A9 08 B6 22 8E AF DC 3C BC 23 71 15 49 31
PCR 1: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 2: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 3: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 4: AF 98 77 B8 72 82 94 7D BE 09 25 10 2E 60 F9 60 80 1E E6 7C
PCR 5: E1 AA 8C DF 53 A4 23 BF DB 2F 4F 0F F2 90 A5 45 21 D8 BF 27
PCR 6: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 7: 5B 93 BB A0 A6 64 A7 10 52 59 4A 70 95 B2 07 75 77 03 45 0B
PCR 8: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 9: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 14: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 15: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 16: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 17: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 18: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 19: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 21: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 22: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 23: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

7. **(オプション) TPM 暗号化プロバイダを有効にします。**

注記 - TPM 暗号化プロバイダは、Oracle Solaris よりも低速です。この手順は、TPM で暗号化操作を実行する場合にのみ実行してください。

```
# cryptoadm install provider='/usr/lib/security/$ISA/pkcs11_tpm.so'
# cryptoadm list -mv provider='/usr/lib/security/$ISA/pkcs11_tpm.so'
```

▼ セキュアなキーストアとして TPM を使用するために PKCS #11 コンシューマを有効にする方法

始める前に この手順を実行するには、システムに TPM をインストールし、有効にする必要があります。`tcsd` デーモンも動作していることを確認します。

`root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **(オプション) TPM PKCS #11 トークンプロバイダがインストールされていない場合、そのプロバイダをインストールします。**

   注記 - この手順が必要かどうかを確認するには、`cryptoadm list` コマンドの実行時に `pkcs11_tpm.so` プロバイダが含まれていることを確認します。

   ```
   # pkg install pkcs11_tpm
   # cryptoadm install provider='/usr/lib/security/$ISA/pkcs11_tpm.so'
   ```

2. **TPM デバイスがインストールされていることを確認します。**

   ```
   # ls -alF /dev/tpm
   lrwxrwxrwx 1 root 39 Dec 27 2011 /dev/tpm -> ../devices/pci@0,0/isa@1/tpm@1,1670:tpm
   ```

3. **`tcsd` デーモンを有効にします。**

   ```
   # svcadm enable tcsd
   ```

4. **(オプション) TPM 所有者がインストールされていない場合、TPM を初期化します。**

   注記 - この手順が必要かどうかを確認するには、`tpmadm status` コマンドを実行します。

   ```
   # tpmadm init
   ```

5. **個人用の TPM で保護されたトークンの格納領域を初期化します。**

   ```
   $ pktool inittoken currlabel=TPM
   ```

   注記 - この手順は、個々のユーザーが実行する必要があります。

6. **セキュリティー責任者のトークン PIN を設定します。**

   ```
   $ pktool setpin token=tpm/TPM usertype=so
   ```

7. **ユーザーの PIN を設定します。**

   ```
   $ pktool setpin token=tpm/TPM
   ```

8. **トークンを初期化するときに使用したトークン名を指定することで、TPM デバイスを使用する鍵と証明書を生成します。**

```
$ pktool gencert token=tpm/TPM -i
$ pktool list token=tpm/TPM
```

既存のアプリケーションで `libpkcs11` の暗号化フレームワークがすでに使用されている場合は、アプリケーションでセッション用の TPM トークンデバイスを選択することで、それらの操作で TPM トークンを使用できます。

例 2 TPM を使用するための PKCS #11 コンシューマの有効化

この例では、最初に TPM トークンに新しい名前が割り当てられます。トークン上のすべての後続アクションで、この新しい名前が参照されます。

```
$ pktool inittoken currlabel=TPM newlabel=JanDoeTPM
$ pktool setpin token=tpm/JanDoeTPM so
$ pktool gencert token=tpm/JanDoeTPM -i
$ pktool list token=tpm/JanDoeTPM
```

TPM のトラブルシューティング

このセクションの内容は次のとおりです。

TPM ステータスのモニタリング

このセクションで説明するコマンドを使用して、正常な TPM の使用を可能にするさまざまな動作コンポーネントをモニターし、TPM の問題のトラブルシューティングを行います。

- `tcsd` デーモンが動作していることを確認するには:

```
# svcs tcsd
STATE      STIME      FMRI
online     Nov_07     svc:/application/security/tcsd:default
```

- TPM デバイスがインストールされていることを確認するには:

```
# ls -alF /dev/tpm
lrwxrwxrwx 1 root 39 Dec 27 2011 /dev/tpm -> ../devices/pci@0,0/isa@1/tpm@1,1670:tpm
```

- TSS ソフトウェアパッケージがインストールされていることを確認するには:

```
# pkg info trousers
Name: library/security/trousers
Summary: TrouSerS TCG software to access a TPM device
Description: The TrouSerS library provides a software stack from the
Trusted Computer Group (TCG) that accesses a Trusted Platform Module
(TPM) hardware device.
Category: System/Security
State: Installed
Publisher: solaris
Version: 0.3.6
Build Release: 5.11
Branch: 0.175.1.0.0.24.0
Packaging Date: September 4, 2012 05:28:21 PM
Size: 3.65 MB
FMRI: pkg://solaris/library/security/
trousers@0.3.6,5.11-0.175.1.0.0.24.0:20120904T1728212
```

- TPM の現在のステータスを確認するには:
  - 次の出力は、TPM が初期化されていないことを意味します。

```
# tpmadm status
TPM Version: 1.2 (STM  Rev: 13.12, SpecLevel: 2, ErrataRev: 3)
No TPM owner installed.
```

  - 次の出力は、`svcadm enable tcsd` コマンドを使用して `tcsd` サービスを起動する必要があることを意味します。

```
# tpmadm status
Connect context: Communication failure (TSS.TSS_E_COMM_FAILURE 0x3011).
Make sure the tcsd service "svc:/application/security/tcsd" is running.
```

  - 次の出力は、TPM が初期化されていることを意味します。

```
# tpmadm status
TPM Version: 1.2 (IFX Rev: 3.16, SpecLevel: 2, ErrataRev: 2)
TPM resources
        Contexts: 32/32 available
        Sessions: 20/20 available
        Authentication Sessions: 20/20 available
        Loaded Keys: 8/10 available
Platform Configuration Registers (24)
        PCR 0:  D1 8A 59 A6 64 6C 38 D7 01 14 F6 F5 05 77 2B 2C AA 4A AC 7F
        PCR 1:  AE 00 DE C4 9F 35 C6 A4 1B 5D E7 7D 57 73 87 2C B2 B9 F2 79
        PCR 2:  3C 80 7F A0 CE 0D 71 47 3D BB 27 62 B8 26 81 23 F6 37 C1 4C
        PCR 3:  3A 3F 78 0F 11 A4 B4 99 69 FC AA 80 CD 6E 39 57 C3 3B 22 75
        PCR 4:  67 36 B9 7C 15 A0 1E 59 5A E5 83 F7 D5 B4 60 16 FB F3 9F 07
        PCR 5:  A0 AD 25 17 E3 1A 35 7D 70 2B 46 3C 2D 82 6A 64 8A DE 82 5A
```

```
PCR 6:  3A 3F 78 0F 11 A4 B4 99 69 FC AA 80 CD 6E 39 57 C3 3B 22 75
PCR 7:  3A 3F 78 0F 11 A4 B4 99 69 FC AA 80 CD 6E 39 57 C3 3B 22 75
PCR 8:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 9:  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 11: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 12: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 13: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 14: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 15: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 16: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
PCR 17: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 18: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 19: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 20: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 21: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 22: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
PCR 23: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

- 以前に TPM が再インストールされたあとの要件として、TPM をクリアするには:
  - Oracle Solaris プロンプトで:

    ```
    # tpmadm clear owner
    ```

  - ILOM プロンプトで:

    ```
    -> stop /SYS
    -> set /HOST/tpm forceclear=true
    -> start /SYS
    ```

SPARC: TPM フェイルオーバーオプション

Oracle Solaris 11.4 がインストールされている SPARC マルチドメインサーバーには、TPM を含む SP/SPP ボードにフェイルオーバーする機能があります。`tpmadm` コマンドの `-failover` オプションを使用することで、TPM フェイルオーバーを有効にできます。

`-failover` オプションは、TPM 所有者の PIN と、移行キーの新しい PIN を求めるプロンプトを表示します。これらの設定は、TPM チップが別の SPARC SP/SPP ボード上の新しい TPM チップにフェイルオーバーした場合に TPM キーストアをバックアップおよび復元するために使用されます。

手順については、44 ページの「Oracle ILOM インタフェースを使用して TPM を初期化する方法」のバックアップ手順を参照してください。`tpmadm(8)` のマニュアルページも参照してください。

SPARC: TPM データおよび鍵の移行または復元

Oracle Solaris 11.4 がインストールされている SPARC マルチドメインサーバーは、`-failover` オプションが以前に有効になっている場合、TPM を含む SP/SPP ボードにフェイルオーバーできます。52 ページの「TPM フェイルオーバーオプション」を参照してください。

その他のすべてのプラットフォームでは、手動バックアップが作成されている必要があります。46 ページの「TPM データおよび鍵をバックアップする方法」を参照してください。手動バックアップが作成されている場合は、次の手順を使用して TPM データおよび鍵のバックアップを新しい SP にインストールできます。

▼ SPARC: TPM データおよび鍵を移行または復元する方法

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **TPM データおよび鍵を移行します。**

```
#  tpmadm migrate import
```

2. **データが移行されたことを確認します。**

```
# tpmadm keyinfo
    [SYSTEM] 00000000-0000-0000-0000-000000000001 (loaded)
        [SYSTEM] 00000000-0000-0000-0000-00000000000b
         [USER] bc25ec53-239e-6ae8-f888-9e46d8f8f40f
           [USER] f5cc255c-2bd5-cb2d-e961-874f82dad286
```

ILOM を使用した、USB ポートへのアクセスの防止

Oracle ILOM は、T7 プラットフォームなど一部の SPARC プラットフォームにプリインストールされているシステム管理ファームウェアです。Oracle ILOM を使用すると、サーバーにインストールされているコンポーネントをアクティブに管理およびモニターできます。

注記 - 次の手順は、すべてのプラットフォームで完全にサポートされているわけではありません。このオプションが使用可能かどうかを判断するには、使用しているプラットフォームのマニュアルを確認してください。

▼ ILOM を使用して USB ポートを無効にする方法

始める前に このタスクには Oracle ILOM のツールを使用します。この手順を使用するには、ILOM にアクセスできる必要があります。詳細は、Oracle® Integrated Lights Out Manager のドキュメントライブラリを参照してください。

1. **ILOM で、USB 制御を無効状態に設定します。**

```
->set /SYS/MB/USB_CTRL requested_config_state=Disabled
```

2. **設定を有効にするために、DC 電源の再投入を開始します。**

```
->stop /SYS
->start /SYS
```

セキュリティー拡張を使用した、マルウェアに対する保護

Oracle Solaris では、アドレス空間、プロセスヒープ、プロセススタック、ADI ヒープ、および ADI スタックをセキュリティー拡張フレームワークによって保護します。Kerberos デーモンなどのカーネルプロセスでは、セキュリティー拡張はデフォルトで有効になっています。

セキュリティー拡張は、Oracle Solaris の選択されたアプリケーションバイナリを保護します。たとえば、Apache HTTP サーバー、DHCP、Secure Shell、および `sendmail` はセキュリティー拡張によって保護されます。バイナリがセキュリティー拡張で保護されるかどうかを確認するには、例5「バイナリがセキュリティー拡張で保護されているかどうかの確認」を参照してください。

フレームワークの `sxadm` コマンドを使用すると、選択したバイナリのセキュリティー拡張を有効または無効にしたり、それらのプロパティーを管理したりできます。

バイナリのセキュリティー拡張の構成には、次のものがあります。

- **無効** – セキュリティー拡張は、すべてのバイナリに対して無効です。
- **タグ付きバイナリ** – セキュリティー拡張は、バイナリ内にコーディングされているタグによって制御されます。
- **有効** – セキュリティー拡張は、無効にするためのタグが明示的に付いているバイナリを除くすべてのバイナリに対して有効です。

`sxadm` にはデバッグインタフェース `sxadm exec` があります。これは、特定のプログラムを、その 1 回の実行でセキュリティー拡張を有効または無効に指定して実行します。

アドレス空間レイアウトのランダム化

Oracle Solaris では、そのユーザーランドバイナリの多くが、アドレス空間レイアウトのランダム化 (ASLR) セキュリティー拡張でタグ付けされます。ASLR では、アドレス空間の主要な部分の開始アドレスがランダム化されます。このセキュリティー防御メカニズムにより、ソフトウェアの脆弱性を悪用しようとする ROP (Return Oriented Programming) 攻撃を失敗させることができます。

ゾーンは、そのプロセス用にこのランダム化されたレイアウトを継承します。ASLR はすべてのバイナリに最適であるとは限らないため、その使用は、ゾーンのレベルとバイナリのレベルで構成できます。

Oracle Solaris での ASLR のデフォルト値は、`tagged-binaries` です。ASLR を使用するために Oracle Solaris の多くのバイナリにタグが付いています。

`sxadm` コマンドを実行するには、`root` 役割になる必要があります。例および詳細については、`sxadm(8)` のマニュアルページを参照してください。開発者向けの情報については、『Oracle Solaris 12 セキュリティーサービス開発ガイド』を参照してください。

悪影響からのプロセスヒープと実行可能スタックの保護

コンピュータ攻撃の一般的な方法は、悪意のあるコードをメモリー内に入れて、そのコードにジャンプすることです。そのような攻撃は、書き込み可能で実行可能なセグメントに依存しています。`nxheap` および `nxstack` セキュリティー拡張を使用すると、すべての Oracle Solaris プロセスのスタックとヒープを意図的に非実行可能にできます。`nxstack` セキュリティー拡張によって、`noexec_user_stack` システム変数が置き換えられます。

プログラムは、スタック上のデータの読み取りと書き込みを行います。通常、それらはコード用に特別に指定されたメモリーの読み取り専用部分から実行されます。スタック上のバッファーをオーバーフローさせる一部の攻撃では、新しいコードをそのスタックに挿入し、プログラムにそれを実行させようとします。スタックメモリーから実行権を削除すると、これらの攻撃が成功するのを防ぐことができます。ほとんどのプログラムは、実行可能スタックを使用せずに正しく機能できます。

64 ビットプロセスには、常に非実行可能スタックがあります。デフォルトでは、32 ビット SPARC プロセスには実行可能スタックがあります。デフォルトで有効になっている `nxstack` セキュリティー拡張は、32 ビットプロセスのスタックが実行可能にならないようにします。プログラムがスタック上でコードを実行しようとすると

SIGSEGV シグナルが送信されます。このシグナルが送信されると、通常、プログラムはコアダンプして終了します。

ログがデフォルトで書き込まれます。このログは、nxstack セキュリティー拡張を設定したために正しく動作しなくなった、実行可能スタックに依存する有効なプログラムを確認するために役立ちます。メッセージが記録されない場合でも、SIGSEGV シグナルは引き続き送られるので、実行中のプログラムはコアダンプで終了します。57 ページの「プロセススタックおよびプロセスヒープからの悪意のあるコードの実行を防止する方法」および sxadm(8) のマニュアルページを参照してください。

プログラムは、スタック実行を明示的にマークまたは防止することができます。プログラム内の mprotect() 関数は、スタックを実行可能として明示的にマークします。詳細は、mprotect(2) のマニュアルページを参照してください。-z nxstack=enable でコンパイルされたプログラムは、システム全体の設定には関係なく、スタックを非実行可能にします。

ヒープは、動的割り当て用に確保されたメモリーです。これは、アプリケーション (プロセス) が終了すると再利用されます。プロセスヒープから実行権を削除すると、悪意のあるコードがヒープに格納されるのを防ぐことができます。ほとんどのプログラムは、ヒープでコードを実行せずに正しく機能します。

nxheap セキュリティー拡張は、ログと同様にデフォルトで有効になっています。例および詳細については、57 ページの「プロセススタックおよびプロセスヒープからの悪意のあるコードの実行を防止する方法」および sxadm(8) のマニュアルページを参照してください。

nxstack および noexec_user_stack の互換性

noexec_user_stack および noexec_user_stack_log システム変数は非推奨です。ただし、変数が /etc/system ファイルに残っている場合、実行可能スタックの保護は次のことを強制することで実行されます。

- noexec_user_stack が 1 に設定されている場合、nxstack の値はすべてのプロセスで有効なままになります。
- noexec_user_stack が 0 に設定されている場合、nxstack の値は tagged-files になります。
- noexec_user_stack_log が 1 に設定されている場合、エラーメッセージのログファイルは保持されます。
- noexec_user_stack_log が 0 に設定されている場合、エラーメッセージのログファイルは保持されません。

▼ プロセススタックおよびプロセスヒープからの悪意のあるコードの実行を防止する方法

32 ビットの実行可能スタックのセキュリティーリスクに関する説明については、55 ページの「悪影響からのプロセスヒープと実行可能スタックの保護」を参照してください。

始める前に root 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **nxstack および nxheap セキュリティー拡張のステータスを表示します。たとえば、ADI をサポートしている SPARC プラットフォームでは、次のような出力になります。**

```
$ sxadm info
EXTENSION           STATUS                      CONFIGURATION
aslr                enabled (tagged-files)      enabled (default)
nxstack             enabled (all)               enabled (default)
nxheap              enabled (tagged-files)      enabled (default)
adiheap             enabled (tagged-files)      enabled (default)
adistack            enabled (tagged-files)      enabled (default)
```

解析可能な出力を得るには、-po オプションを付けてパラメータを指定します。

```
$ sxadm info -po extension,status,configuration
aslr:enabled.tagged-files:enabled.default
nxstack:enabled.all:enabled.default
nxheap:enabled.tagged-files:enabled.default
adiheap:enabled.tagged-files:enabled.default
adistack:enabled.tagged-files:enabled.default
```

2. **nxheap および nxstack セキュリティー拡張をデフォルトに戻します。**

nxheap または nxstack セキュリティー拡張がデフォルト値以外の値を表示する場合、カスタマイズを削除します。出力例では、次のコマンドを実行します。

```
# sxadm delcust nxheap
# sxadm info
EXTENSION           STATUS                      CONFIGURATION
aslr                enabled (tagged-files)      default (default)
nxstack             enabled (all)               enabled (default)
nxheap              enabled (tagged-files)      enabled (default)
```

nxheap および nxstack のログは /var/adm/messages ファイルに格納されます。

3. **(オプション) エラーメッセージのロギングを無効にするには、log プロパティーを無効にします。**

```
# sxadm set log=disable nxheap
# sxadm set log=disable nxstack
# sxadm get log
EXTENSION           PROPERTY                    VALUE
...
nxstack             log                         disable
nxheap              log                         disable
```

注意事項 `nxstack` 設定が無視される場合、`noexec_user_stack` および `noexec_user_stack_log` システム変数を `/etc/system` ファイルから削除します。次に、`nxstack` セキュリティー拡張を再度有効にします。

`/etc/system` ファイルで `noexec_user_stack` を無効にしても、エントリを削除しない場合、タグが付けられたバイナリは引き続き保護されます。この `tagged-files` 構成では、スタックが実行可能なときにのみ成功できるバイナリを成功させながら、ほとんどの実行可能スタックを悪意のあるコードから保護できます。詳細は、56 ページの「`nxstack` および `noexec_user_stack` の互換性」を参照してください。

adiheap を使用したプロセスヒープの破損の防止

M7 および T7 SPARC プロセッサは、Silicon Secured Memory (SSM) という包括的機能の中のハードウェア機能である、Application Data Integrity (ADI) を備えています。ADI は、リニアバッファーオーバーフローやはぐれたポインタ間接参照など、ソフトウェアを悪用した脅威の防止に役立ちます。

`adiheap` セキュリティー拡張は、ADI チェックをサポートしているメモリーアロケータを制御できます。`adiheap` は、リニアバッファーオーバーフローに対する信頼性の高い防御を提供し、解放済みメモリー使用の問題に対する有効な軽減策を提供します。また、`adiheap` では、バイナリの動作にまだ影響を与えてはいないが、無害なコード変更によってトリガーされる可能性のある、小さな潜在的バグを発見することもできます。

また、標準 C ライブラリ (`libc`) の `malloc` 関数が、ADI 関数をサポートするように拡張されています。システムで ADI がサポートされていないか、または `adiheap` が無効になっている場合、`malloc` メモリー割り当ては影響を受けません。

バイナリの `adiheap` を有効にするには、バイナリにタグ付けするか、`sxadm exec` コマンドを使用してバイナリを実行します。`sxadm` コマンドを使用して、システムの `adiheap` を有効にすることもできます。

アプリケーションが ADI 対応のアロケータをサポートしているかどうかをテストするには、次のようなコマンドを発行します。

```
# sxadm exec -s adiheap=enable application
```

注記 - `sxadm` コマンドでは、現在 `adiheap` に `model=all` プロパティー値は許可されていません。

adistack を使用した ADI ベースのスタック保護

M7 および T7 SPARC プロセッサは、Silicon Secured Memory (SSM) という包括的機能の下のハードウェア機能である、Application Data Integrity (ADI) を備えています。ADI は、リニアバッファーオーバーフローやはぐれたポインタ間接参照など、ソフトウェアを悪用した脅威の防止に役立ちます。

SPARC T7 ベースおよび M7 ベースのシステム、または Application Data Integrity (ADI) をサポートしているほかの新しいプラットフォームでは、adistack セキュリティー拡張が ADI ベースのスタックバッファーオーバーフロー検出を管理し、サポートします。このようなオーバーフローによって、SPARC 64 ビットユーザープロセスのスタックフレームのレジスタ保存領域が上書きされる可能性があります。アクティブになっている場合、adistack は ADI と一緒に、SPARC のレジスタウィンドウのスピルおよびフィル処理を活用して、このタイプのバッファーオーバーフローを検出し、SEGV シグナルを生成します。

adistack はデフォルトでは有効になっていません。アプリケーションでは、次の新しいリンカーオプションでサポートされるバイナリのタグ付けによって adistack を有効にできます。

```
-z sx=adistack[=enable|disable]
```

このオプションの詳細は、例3「adistack を有効にしたアプリケーションのコンパイル」および 62 ページの「オブジェクトごとのセキュリティー拡張の指定」を参照してください。

アプリケーションの adistack を有効にするには、sxadm exec コマンドを使用して拡張を実行します。

注記 - 一部のアプリケーションやインタプリタ型言語は自分のスタックを直接読み取ったり変更したりする場合があるため、adistack では model=all プロパティーはサポートされていません。特に、C++ および Java アプリケーションは adistack をまだサポートしていません。

詳細は、54 ページの「セキュリティー拡張を使用した、マルウェアに対する保護」および sxadm(8) のマニュアルページを参照してください。

例 3 adistack を有効にしたアプリケーションのコンパイル

この例では、基本的なメイクファイルとリンカーオプションを使用して、adistack をオブジェクトごとのセキュリティー拡張として指定し、拡張モードを enable に設定しています。

依存関係およびその解決方法を記述するメイクファイルルールを次のように指定します。

```
CFLAGS=-m64 -O
LDFLAGS = -z sx=adistack=enable

prog: prog.o
 ${CC} ${CFLAGS} ${LDFLAGS} -o prog prog.o
```

この例では、次のようになります。

- CFLAGS 変数は、作成されるオブジェクトが 64 ビットオブジェクトであることを指定します。
- LDFLAGS 環境変数は、作成されるオブジェクトのセキュリティー拡張として `adistack` を有効にします。
- prog バイナリは `prog.o` オブジェクトファイルの存在に依存しています。`prog.o` は `prog.c` に依存しています。

上記のメイクファイルルールを使用して次のコマンドを実行すると、依存関係を満たし、`adistack` を有効にした `prog` バイナリを作成できます。

```
make
cc -m64 -O  -c  prog.c
cc -m64 -O -z sx=adiheap=enable -o prog prog.o
```

詳細は、`make(1S)` および cc(1)のマニュアルページを参照してください。

セキュリティー拡張のステータス継承の有効化

`sxadm exec` コマンドの `-i` オプションは、セキュリティー拡張の構成の継承を有効または無効にします。

例 4 セキュリティー拡張の継承の説明

この例では、ASLR セキュリティー拡張の構成を継承する場合と継承しない場合を示します。

1. このシェルのみに対して ASLR を無効にします。

   ```
   # sxadm exec -s aslr=disable /bin/bash
   ```

2. このシェルからランダム化されたヒープアドレスを確認します。

   このシェルから `pmap self` コマンドを繰り返すと、ランダム化されたヒープアドレスが表示されます。

3. ```
   # pmap self | grep heap
   00000054BF32E000 8K rw---i- [ heap ]
   00000054BF330000 64K rw---i- [ heap ]
   # pmap self | grep heap
   0000005B50708000 32K rw---i- [ heap ]
   ```
   ```

```
# pmap self | grep heap
000000A48D30E000 8K rw---i- [ heap ]
000000A48D310000 64K rw---i- [ heap ]
```

4. シェルおよびすべての子孫に対して ASLR を無効にし、-i オプションを使用して、シェルを実行します。

```
# sxadm exec -i -s aslr=disable /bin/bash
```

5. このシェルから pmap self コマンドを繰り返すと、ASLR が無効になっている一定のヒープアドレスが表示されます。

```
# pmap self | grep heap
0000000080000000         64K rw---i-    [ heap ]
# pmap self | grep heap
0000000080000000         64K rw---i-    [ heap ]
# pmap self | grep heap
0000000080000000         64K rw---i-    [ heap ]
```

例 5 バイナリがセキュリティー拡張で保護されているかどうかの確認

elfdump -d コマンドを使用すると、特定のバイナリがセキュリティー拡張付きでコンパイルされているかどうかを確認できます。バイナリを保護するには、システムでセキュリティー拡張が有効になっている必要があります。次の出力は、3 つの拡張が有効で、2 つが無効であることを示しています。

```
$ sxadm infoEXTENSION           STATUS                      CONFIGURATION
aslr                enabled (tagged-files)       enabled (default)
nxstack             enabled (all)                enabled (default)
nxheap              enabled (tagged-files)       enabled (default)
adiheap             not supported                not supported
adistack            not supported                not supported
```

次の出力は、cat コマンドおよび ipsecconf コマンドがセキュリティー拡張で保護されていることを示しています。

```
$ elfdump -d /bin/cat | grep SUNW_SX
     [33]  SUNW_SX_ASLR     0x2         ENABLE
     [34]  SUNW_SX_NXHEAP   0x2         ENABLE
     [35]  SUNW_SX_NXSTACK  0x2         ENABLE
$ elfdump -d /usr/sbin/ipsecconf | grep SUNW_SX
     [35]  SUNW_SX_ASLR     0x2         ENABLE
     [36]  SUNW_SX_NXHEAP   0x2         ENABLE
     [37]  SUNW_SX_NXSTACK  0x2         ENABLE
```

次の出力は、DHCP および Secure Shell のユーザーランドバイナリがセキュリティー拡張で保護されていることを示しています。

```
$ elfdump -d /usr/sbin/dhcpagent | grep SUNW_SX
     [52]  SUNW_SX_ASLR     0x2         ENABLE
     [53]  SUNW_SX_NXHEAP   0x2         ENABLE
     [54]  SUNW_SX_NXSTACK  0x2         ENABLE
$ elfdump -d /usr/bin/ssh | grep SUNW_SX
     [43]  SUNW_SX_ASLR     0x2         ENABLE
     [44]  SUNW_SX_NXHEAP   0x2         ENABLE
```

```
[45]  SUNW_SX_NXSTACK  0x2        ENABLE
```

オブジェクトごとのセキュリティー拡張の指定

管理者は `ld -z` コマンドを使用して、オブジェクトごとのセキュリティー拡張を指定できます。このリリースでは、さまざまなセキュリティー拡張を次のように指定できる一貫した方法が、改訂されたコマンドオプションによって提供されています。

`ld -z sx=`*extension*`[`*mode*`],...`

extension 変数はセキュリティー拡張の名前に置き換えてください。*mode* 変数は `enable` または `disable` に置き換えてください。モードを省略した場合、拡張は有効になります。次のセキュリティー拡張を指定できます。

`aslr`
: プロセスのアドレス空間レイアウトのランダム化動作を指定します。

`nxheap`
: プロセスの実行不能ヒープの要件を指定します。

`nxstack`
: プロセスの実行不能スタックの要件を指定します。

`adiheap`
: プロセス内のメモリーアロケータに対する Application Data Integrity (ADI) 要件を指定します。

`adistack`
: プロセスの ADI (アプリケーションデータ整合性) スタック保護の要件を指定します。

  例3「`adistack` を有効にしたアプリケーションのコンパイル」で使用されているこのオプションを参照してください。詳細は、`ld(1)` のマニュアルページおよび『Oracle Solaris 12 リンカーとライブラリガイド』を参照してください。

♦ ♦ ♦ 第 3 章

3 システムアクセスの制御

この章では、Oracle Solaris システムにアクセスできるユーザーを制御する方法について説明します。

この章の内容は次のとおりです。

システムセキュリティーの概要については、第1章「コンピュータシステムセキュリティーの管理」を参照してください。

ログインとパスワードのセキュリティー

システムへのアクセスを保護するために、リモートログインを制限したり、ユーザーにパスワードを持つように要求したり、root アカウントに複雑なパスワードを設定するように要求したりできます。ユーザーアクセスを管理するために、ユーザーにセキュリティーメッセージを表示したり、失敗したアクセス試行をモニターしたり、ログインを一時的に無効にしたりできます。

次のタスクマップは、ユーザーログインをモニターする手順と、ユーザーログインを無効にする手順を示しています。

表 4 ログインとパスワードの保護タスクマップ

タスク	説明	手順
ログイン時に、ユーザーにサイトセキュリティーを通知します。	ログイン画面に、サイトセキュリティー情報を含むテキストメッセージを表示します。	64 ページの「バナーファイルにセキュリティーメッセージを配置する方法」
ユーザーのログインステータスを表示します。	ユーザーのログインアカウントについての広範な情報 (フルネーム、パスワードの有効期限など) を一覧表示します。	65 ページの「ユーザーのログインステータスを表示する方法」

タスク	説明	手順
パスワードを所有していないユーザーを発見します。	パスワードを必要としないアカウントを持つユーザーだけを検出します。	66 ページの「パスワードを持たないユーザーを表示する方法」
ログインを一時的に無効にします。	システムシャットダウンや定常的な保守の中でコンピュータシステムへのユーザーログインを拒否します。	66 ページの「ユーザーのログインを一時的に無効にする方法」

▼ バナーファイルにセキュリティーメッセージを配置する方法

この手順を使用して、サイトのセキュリティーポリシーが反映されたセキュリティーメッセージを 2 つのバナーファイル内に作成します。`/etc/issue` ファイルは認証前に表示され、`/etc/motd` ファイルは認証後に表示されます。

注記 - この手順のサンプルメッセージは、アメリカ合衆国政府の要件を満たしておらず、ユーザーのセキュリティーポリシーも満たしていない可能性があります。セキュリティーメッセージの内容については、会社の弁護士に相談してください。

始める前に Administrator Message Edit 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **`/etc/issue` ファイルを作成し、セキュリティーメッセージを追加します。**

```
# pfedit /etc/issue
ALERT   ALERT   ALERT   ALERT   ALERT

This system is available to authorized users only.

If you are an authorized user, continue.

Your actions are monitored, and can be recorded.
```

   `ssh`、`graphical-login/gdm`、および FTP サービスの場合と同様に、認証前に、`login` コマンドによって `/etc/issue` の内容が表示されます。

   詳細は、`issue(5)` および `pfedit(8)` のマニュアルページを参照してください。

2. **セキュリティーメッセージを `/etc/motd` ファイルに追加します。**

```
# pfedit /etc/motd
This system serves authorized users only. Activity is monitored and reported.
```

   Oracle Solaris では、ユーザーの初期シェルによって `/etc/motd` ファイルの内容が表示されます。

▼ ユーザーのログインステータスを表示する方法

始める前に `logins` コマンドを使用するには、User Management または User Security 権利プロファイルが割り当てられている管理者になる必要があります。デフォルトでは、`root` 役割がこの承認を持っています。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

- **`logins` コマンドを使用してユーザーのログインステータスを表示します。**

```
# logins -x -l username
```

`-x`	ログインステータス情報の拡張セットを表示します。
`-l` *username*	指定するユーザーのログインステータスを表示します。変数 *username* はユーザーのログイン名です。複数のログイン名はコンマで区切ります。

`logins` コマンドは、適切なパスワードデータベースを使ってユーザーのログインステータスを表示します。このデータベースは、ローカルの `/etc/passwd` ファイルか、ネームサービスのパスワードデータベースです。詳細は、`logins(8)` のマニュアルページを参照してください。

例 6 ユーザーのログインステータスを表示する

次の例では、ユーザー `jdoe` のログインステータスが表示されます。

```
# logins -x -l jdoe
jdoe       500     staff          10   Jaylee Jaye Doe
/home/jdoe
/bin/bash
PS 010103 10 7 -1
```

`jdoe`	ユーザーのログイン名を示します。
`500`	ユーザー ID (UID) を示します。
`staff`	ユーザーのプライマリグループを示します。
`10`	グループ ID (GID) を示します。
Jaylee Jaye Doe	コメントを示します。
`/home/jdoe`	ユーザーのホームディレクトリを示します。
`/bin/bash`	ログインシェルを示します。
`PS 010170 10 7 -1`	次のパスワード有効期限情報を示します。 ■ パスワードの最終変更日

- 次に変更するまでに必要な日数
- 変更しないで使用できる日数
- 警告期間

▼ パスワードを持たないユーザーを表示する方法

始める前に `logins` コマンドを使用するには、User Management または User Security 権利プロファイルが割り当てられている管理者になる必要があります。デフォルトでは、`root` 役割がこの承認を持っています。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』の「Using Your Assigned Administrative Rights」を参照してください。

- **`logins` コマンドを使用して、パスワードを持っていないユーザーをすべて表示します。**

```
# logins -p
```

`-p` オプションを指定すると、パスワードを持たないユーザーが一覧表示されます。`logins` コマンドは、`system/name-service/switch` サービスの `password` プロパティーで分散ネームサービスが指定されていないかぎり、ローカルシステムの `passwd` データベースを使用します。

例 7 パスワードを持たないアカウントの表示

次の例では、ユーザー `pmorph` と役割 `roletop` はパスワードを持っていません。

```
# logins -p
pmorph          501     other           1       Polly Morph
roletop         211     admin           1       Role Top
#
```

▼ ユーザーのログインを一時的に無効にする方法

システムシャットダウンや定常的な保守の際にユーザーのログインを一時的に無効にします。

注記 - この手順によって、すべてのユーザーが影響を受けるわけではありません。この手順で作成された `/etc/nologin` ファイルが存在していても、次のユーザーは引き続きシステムにログインできます。

- スーパーユーザー
- `root` 役割が割り当てられているユーザー
- `solaris.system.maintenance` 承認が割り当てられているユーザー

詳細は、nologin(5) のマニュアルページを参照してください。

始める前に solaris.admin.edit/etc/nologin 承認が割り当てられている管理者になる必要があります。デフォルトでは、root 役割がこの承認を持っています。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **テキストエディタで、/etc/nologin ファイルを作成します。**

   ```
   # pfedit /etc/nologin
   ```

   solaris.admin.edit/etc/nologin 承認の使用の例については、例8「ユーザーログインを無効にする」 を参照してください。

2. **システムの利用に関するメッセージを入力します。**

3. **ファイルを閉じて、保存します。**

例 8 ユーザーログインを無効にする

この例では、ユーザーが、システム使用不可の通知の書き込みを承認されます。

```
$ pfedit /etc/nologin
***No logins permitted.***

***The system will be unavailable until 12 noon.***
```

パスワード暗号化のデフォルトアルゴリズムを変更する

デフォルトの crypt_sha256 アルゴリズムは、値 5 によって表されます。別のアルゴリズムに切り替えるには、別の識別子を割り当てます。パスワード暗号化アルゴリズムと対応する識別子のリストについては、表1を参照してください。

注記 - 可能な場合は、FIPS 140-2 承認アルゴリズムを使用してください。FIPS 140-2 承認アルゴリズムのリストについては、『Oracle Solaris 12 での FIPS 140 対応システムの使用』 の 「FIPS 140-2 Algorithm Lists and Certificate References for Oracle Solaris Systems」を参照してください。

新しいアルゴリズムは新しいユーザーのパスワード暗号化にのみ適用されます。既存のユーザーの場合、以前のアルゴリズムが CRYPT_ALGORITHMS_ALLOW パラメータに定義されたままで、unix 以外であれば、それが引き続き機能します。この場合に暗号化の実装状態を確認する方法については、18 ページの「パスワードハッシュ構成」を参照してください。新しいパスワード暗号化アルゴリズムに既存のユーザーを追加する

には、CRYPT_ALGORITHMS_ALLOW パラメータから以前のアルゴリズムを削除してください。

選択したアルゴリズムの構成の詳細については、policy.conf(5) のマニュアルページを参照してください。

注記 - このセクションの手順および例は、account-policy サービスを使用している場合には機能しません。このサービスを有効にしている場合は、以前は policy.conf ファイルの編集によって変更していたセキュリティー属性の変更方法について、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」を参照してください。

▼ パスワード暗号化のアルゴリズムを指定する方法

始める前に root 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **/etc/security/policy.conf ファイルで、選択した暗号化アルゴリズムを表す識別子を CRYPT_DEFAULT 変数の値として指定します。**

2. **(オプション) 選択についての説明をファイルにコメントします。**
   例:

```
# cat  /etc/security/policy.conf
…
# Sets the SHA256 (5) algorithm as default.
# SHA256 supports 255-character passwords.
# Passwords previously encrypted with MD5 (1) will be encrypted
# with SHA256 (5) when users change their passwords.
#CRYPT_DEFAULT=1
CRYPT_DEFAULT=5
```

   この例では、CRYPT_DEFAULT の新しい値が 5 (SHA256、SHA256 アルゴリズム) になっています。SHA は、Secure Hash Algorithm (セキュアハッシュアルゴリズム) を表します。このアルゴリズムは、SHA-2 ファミリのメンバーです。SHA256 では 255 文字のパスワードがサポートされます。

3. **(オプション) CRYPT_ALGORITHM_ALLOWED から以前のアルゴリズムを削除して、新しいアルゴリズムを既存のユーザーに適用させます。**
   たとえば、SHA256 アルゴリズムが既存のユーザーにも確実に適用されるようにするには、CRYPT_ALGORITHM_ALLOWED から MD5 を示す以前の識別子 1 を除外するようにしてください。

注記 - さらに、FIPS 140-2 セキュリティーを向上させるには、Blowfish アルゴリズム (2a) をエントリから除外します。

```
CRYPT_ALGORITHMS_ALLOW=5,6
```

例 9 異機種システム混在環境でパスワードの暗号化アルゴリズムを制約する

この例では、BSD および Linux システムが含まれるネットワーク上の管理者は、すべてのシステムで使用できるようにパスワードを構成します。SHA512 暗号化は一部のネットワークアプリケーションで処理できないため、管理者はその識別子を許容されるアルゴリズムのリストに含めません。管理者は、CRYPT_DEFAULT 変数の値として SHA256 アルゴリズム 5 を保持しています。CRYPT_ALGORITHMS_ALLOW 変数には、BSD および Linux システムと互換性のある MD5 識別子と、BSD システムと互換性のある Blowfish 識別子が含まれています。5 は CRYPT_DEFAULT アルゴリズムであるため、CRYPT_ALGORITHMS_ALLOW リストに載せる必要はありません。しかし、保守のために、管理者は 5 を CRYPT_ALGORITHMS_ALLOW リストに入れ、使われていない識別子を CRYPT_ALGORITHMS_DEPRECATE リストに入れます。

```
CRYPT_ALGORITHMS_ALLOW=1,2a,5
#CRYPT_ALGORITHMS_DEPRECATE=__unix__,md5,6
CRYPT_DEFAULT=5
```

▼ NIS ドメイン用の新しいパスワードアルゴリズムを指定する方法

NIS ドメインのユーザーがパスワードを変更すると、NIS クライアントは、/etc/security/policy.conf ファイルにある自身のローカルアルゴリズム構成を調べ、NIS クライアントシステムでパスワードを暗号化します。

注記 - account-policy SMF ステンシルを使用している場合で、config/etc_default_passwd プロパティーが有効になっているときは、この新しいアルゴリズムを使用するすべてのシステムで、対応する SMF プロパティーを変更する必要があります。例については、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」の手順を参照してください。account-policy(8S) のマニュアルページも参照してください。

始める前に root 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **パスワード暗号化アルゴリズムを NIS クライアント上の /etc/security/policy.conf ファイルに指定します。**
2. **変更された /etc/security/policy.conf ファイルを NIS ドメインのすべてのクライアントシステムにコピーします。**

3. **混乱をできるだけ少なくするために、変更された `/etc/security/policy.conf` ファイルを NIS ルートサーバーとスレーブサーバーにコピーします。**

▼ LDAP ドメイン用の新しいパスワードアルゴリズムを指定する方法

適切に構成された LDAP クライアントでは、新しいパスワードアルゴリズムを使用できます。LDAP クライアントは NIS クライアントと同じように動作します。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』の「Using Your Assigned Administrative Rights」を参照してください。

1. **パスワード暗号化アルゴリズムを LDAP クライアント上の `/etc/security/policy.conf` ファイルに指定します。**

注記 - `account-policy` SMF ステンシルを使用している場合で、`config/etc_default_passwd` プロパティーが有効になっているときは、この新しいアルゴリズムを使用するすべてのシステムで、対応する SMF プロパティーを変更する必要があります。例については、『Securing Users and Processes in Oracle Solaris 11.4』の「Modifying Rights System-Wide As SMF Properties」の手順を参照してください。`account-policy(8S)` のマニュアルページも参照してください。

---

2. **変更された `policy.conf` ファイルを LDAP ドメインのすべてのクライアントシステムにコピーします。**

3. **クライアントの `/etc/pam.conf` ファイルが `pam_ldap` モジュールを使用していないことを確認します。**

   `pam_ldap.so.1` を含むエントリの前にコメント記号 (#) があることを確認します。また、`pam_authtok_store.so.1` モジュールには `server_policy` オプションを使用しないでください。

   ローカルアルゴリズム構成に基づくパスワードの暗号化は、クライアントの `pam.conf` ファイルの PAM エントリに従って行われます。パスワードの認証もこの PAM エントリによって行われます。

   LDAP ドメインのユーザーがパスワードを変更すると、LDAP クライアントは、`/etc/security/policy.conf` ファイルにある自身のローカルアルゴリズム構成を調べ、LDAP クライアントシステムでパスワードを暗号化します。続いてクライアントは、`{crypt}` タグ付きの暗号化パスワードをサーバーに送信します。このタグは、パスワードが暗号化済みであることをサーバーに知らせます。パスワードはそのままの形でサーバーに格納されます。認証時に、クライアントはこのパスワードをサーバーから取り出します。クライアントは、このパスワードと、入力されたユーザーのパスワードからクライアントが暗号化したばかりのパスワードとを比較します。

注記 - LDAP サーバーでパスワードポリシー制御を使用するには、pam.conf ファイルの pam_authtok_store エントリに server_policy オプションを指定します。パスワードはそのあと、LDAP サーバー上で暗号化されます。手順については、『Oracle Solaris 12 ディレクトリサービスとネームサービスでの作業: LDAP』 の 第 4 章, 「Setting Up an Oracle Unified Directory Server or OpenLDAP Server」を参照してください。

root アクセスのモニタリングと制限

デフォルトでは、root 役割は初期ユーザーに割り当てられ、ローカルシステムに直接ログインしたり、Oracle Solaris システムにリモートログインしたりすることはできません。

▼ だれが su コマンドを使用しているかをモニターする方法

sulog ファイルには、ユーザーから root に切り替えるために使用される su の試行だけでなく、ユーザー切替え (su) コマンドのすべての使用が記録されます。

このファイルへの su ログの記録は、デフォルトで、 /etc/default/su ファイルの次のエントリで有効になっています。

```
SULOG=/var/adm/sulog
```

注記 - account-policy SMF ステンシルを使用している場合で、config/etc_default_passwd プロパティーが有効になっているときは、この新しいアルゴリズムを使用するすべてのシステムで、対応する SMF プロパティーを変更する必要があります。例については、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」の手順を参照してください。account-policy(8S) のマニュアルページも参照してください。

始める前に root 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

- **/var/adm/sulog ファイルの内容を定期的にモニタリングします。**

```
# more /var/adm/sulog
SU 12/20 16:26 + pts/0 stacey-root
SU 12/21 10:59 + pts/0 stacey-root
```

```
SU 01/12 11:11 + pts/0 root-rimmer
SU 01/12 14:56 + pts/0 jdoe-root
SU 01/12 14:57 + pts/0 jdoe-root
```

ここには、次のような情報が表示されます。

- コマンドが入力された日時。
- 試行に成功した場合。プラス記号 (+) は成功を示し、マイナス記号 (-) は失敗を示します。
- コマンドが実行されたポート。
- ユーザー名と切り替えたユーザー ID。

注意事項 `???` を含むエントリは、`su` コマンドの制御端末を識別できないことを示しています。通常、デスクトップが表示される前の `su` コマンドのシステム呼び出しには、`???` が含まれます。たとえば、`SU 10/10 08:08 + ??? root-root` です。ユーザーがデスクトップセッションを開始すると、`ttynam` コマンドは、次のように制御端末の値を `sulog` に返します。 `SU 10/10 10:10 + pts/3 jdoe-root`。

次のようなエントリは、`su` コマンドがコマンド行で呼び出されなかったことを示している場合があります。`SU 10/10 10:20 + ??? root-oracle`。Trusted Extensions のユーザーが GUI を使用して `oracle` 役割に切り替えた可能性があります。

▼ root ログインを制限およびモニターする方法

この方法では、ローカルシステムにアクセスしようとする `root` をただちに検出できます。

始める前に `root` 役割になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **`/etc/default/login` ファイルの `CONSOLE` エントリを確認します。**

注記 - `account-policy` SMF ステンシルを使用している場合で、`config/etc_default_login` プロパティーが有効になっているときは、`login_policy/root_login_device` プロパティーを表示して変更する必要があります。例については、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Modifying Rights System-Wide As SMF Properties」の手順を参照してください。`account-policy(8S)` のマニュアルページも参照してください。

```
CONSOLE=/dev/console
```

デフォルトのコンソールデバイスは `/dev/console` に設定されています。このように設定されていると、`root` はコンソールにログインできます。`root` はリモートログインを行うことはできません。

2. **root がリモートログインできないことを検証します。**

   リモートシステムから、root としてログインを試みます。

   ```
   system2 $ ssh -l root system1
   Password:      system1 の root パスワードを入力します
   Password:
   Password:
   Permission denied (gssapi-keyex,gssapi-with-mic,publickey,keyboard-interactive).
   ```

   デフォルト構成では、root は役割であり、役割はログインできません。また、デフォルトの構成では、ssh プロトコルによって root ユーザーのログインが阻止されます。

3. **root になろうとする試みをモニターします。**

   デフォルトでは、root になろうとする試みが SYSLOG ユーティリティーによってコンソールに表示されます。

   a. **デスクトップに端末コンソールを開きます。**

   b. **別のウィンドウで、su コマンドを使用して root になります。**

      ```
      $ su -
      Password:      root パスワードを入力します
      #
      ```

      端末コンソールにメッセージが表示されます。

      ```
      Sep 7 13:22:57 system1 su: 'su root' succeeded for jdoe on /dev/pts/6
      ```

例 10 root アクセスの試行のログ記録

この例では、root の試行は SYSLOG によってログに記録されていません。そのため、管理者は、/etc/default/su ファイル内の #CONSOLE=/dev/console エントリからコメントを削除することによって、これらの試行をログに記録します。

```
# CONSOLE determines whether attempts to su to root should be logged
# to the named device
#
CONSOLE=/dev/console
```

ユーザーが root になろうとすると、この試行が端末コンソールに出力されます。

```
SU 09/07 16:38 + pts/8 jdoe-root
```

注意事項 /etc/default/login ファイルにデフォルトの CONSOLE エントリが含まれている場合にリモートシステムから root になるには、ユーザーはまず、自分のユーザー名を使用してログインする必要があります。自分のユーザー名を使用してログインしたあと、ユーザーは su コマンドを使用して root になることができます。

コンソールに Last login: Thu Sep 7 15:13:11 2017 from system2 のようなエントリが表示された場合、システムは、リモート root ログインを許可するように構成

されています。リモート root アクセスを防止するには、/etc/default/login ファイル内の #CONSOLE=/dev/console エントリを CONSOLE=/dev/console に変更します。ssh プロトコルをデフォルトに戻す方法については、sshd_config(5) のマニュアルページを参照してください。

システムハードウェアアクセスの制御

物理的なマシンは、ハードウェア設定にアクセスする際にパスワードを入力させることで保護できます。また、ユーザーがアボートシーケンスを使ってウィンドウ表示システムから離れるのを防ぐことでシステムを保護することもできます。

BIOS を保護するには、ベンダーのドキュメントを参照してください。ブート時に暗号化と検証を必要にするには、42 ページの「Trusted Platform Module の使用」および 37 ページの「ベリファイドブートの使用」を参照してください。

▼ SPARC ハードウェアへのアクセスにパスワードを必要にする方法

始める前に Device Security、Maintenance and Repair、または System Administrator 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **端末ウィンドウで、PROM セキュリティーモードを有効にします。**

```
# eeprom security-mode=command

Changing PROM password:
New password:       <Type password>
Retype new password:       <Retype password>
```

値として command か full を選択します。詳細は、eeprom(8) のマニュアルページを参照してください。

前述のコマンドを入力する際に PROM パスワードを要求されない場合は、システムがすでに PROM パスワードを持っています。

2. **(オプション) PROM パスワードを変更します。**

注意 - PROM パスワードを忘れないでください。このパスワードがないと、ハードウェアは使用できません。

```
# eeprom security-password=       Return キーを押します
Changing PROM password:
New password:       <Type password>
Retype new password:       <Retype password>
```

新しい PROM セキュリティーモードとパスワードはただちに有効になりますが、それが認識できるのは、ほとんどの場合、次回のブート時です。

▼ システムのアボートシーケンスを無効にする方法

注記 - 一部のサーバーシステムにはキースイッチがあります。このキースイッチを安全な位置に設定すると、ソフトウェアキーボードのアボート設定がオーバーライドされます。そのため、次の手順で行なった変更が実装されないことがあります。

始める前に `solaris.admin.edit/etc/default/kbd` 承認が割り当てられている管理者になる必要があります。デフォルトでは、`root` 役割がこの承認を持っています。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』の「Using Your Assigned Administrative Rights」を参照してください。

1. **`KEYBOARD_ABORT` の値を `disable` に変更します。**

   `/etc/default/kbd` ファイルの `enable` 行をコメントにします。次に `disable` 行を追加します。

```
# cat /etc/default/kbd
…
# KEYBOARD_ABORT affects the default behavior of the keyboard abort
# sequence, see kbd(1) for details.  The default value is "enable".
# The optional value is "disable".  Any other value is ignored.
…
#KEYBOARD_ABORT=enable
KEYBOARD_ABORT=disable
```

2. **キーボードのデフォルトを更新します。**

```
# kbd -i
```

♦ ♦ ♦ 第 4 章

4 デバイスアクセスの制御

この章では、システムに接続されているデバイスを保護するための作業について説明するとともに、参考となるセクションを示します。この章の内容は次のとおりです。

デバイスの保護についての概要は、29 ページの「デバイスアクセスの制御」を参照してください。

デバイスポリシーの構成

デバイスポリシーは、システムに不可欠なデバイスに対するアクセスの制限または防止を行うものです。デバイスポリシーはカーネル内で適用されます。

次のタスクマップは、デバイスポリシーに関連するデバイス構成作業の参照先を示しています。

表 5 デバイスポリシーの構成タスクマップ

タスク	説明	手順
システム上のデバイスのデバイスポリシーを表示します。	デバイスとそれらのデバイスポリシーの一覧を表示します。	78 ページの「デバイスポリシーを表示する方法」
デバイスポリシーの変更を監査します。	デバイスポリシーの変更を監査トレール内に記録します。	78 ページの「デバイスポリシーの変更を監査する方法」
`/dev/arp` にアクセスします。	Oracle Solaris IP MIB-II 情報を取得します。	79 ページの「`/dev/*` デバイスから IP MIB-II 情報を取得する方法」

▼ デバイスポリシーを表示する方法

- **システム上のすべてのデバイスのデバイスポリシーを表示します。**

```
$ getdevpolicy | more
DEFAULT
read_priv_set=none
write_priv_set=none
ip:*
read_priv_set=net_rawaccess
write_priv_set=net_rawaccess
…
```

例 11 特定のデバイスのデバイスポリシーを表示する

この例では、3 つのデバイスのデバイスポリシーが表示されています。

```
$ getdevpolicy /dev/allkmem /dev/ipsecesp /dev/net0
/dev/allkmem
read_priv_set=all
write_priv_set=all
/dev/ipsecesp
read_priv_set=sys_net_config
write_priv_set=sys_net_config
/dev/bge
read_priv_set=net_rawaccess
write_priv_set=net_rawaccess
```

▼ デバイスポリシーの変更を監査する方法

デフォルトでは、as 監査クラスに、AUE_MODDEVPLCY 監査イベントが含まれます。

始める前に Audit Configuration 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

- **AUE_MODDEVPLCY 監査イベントを含む監査クラスをあらかじめ選択します。**

```
# auditconfig -getflags
current-flags
# auditconfig -setflags current-flags,as
```

詳細な手順については、『Managing Auditing in Oracle Solaris 11.4』 の 「How to Preselect Audit Classes」を参照してください。

▼ /dev/* デバイスから IP MIB-II 情報を取得する方法

Oracle Solaris IP MIB-II 情報を取得するアプリケーションは、/dev/ip ではなく /dev/arp を開く必要があります。

1. **/dev/ip および /dev/arp のデバイスポリシーを決定します。**

```
$ getdevpolicy /dev/ip /dev/arp
/dev/ip
read_priv_set=net_rawaccess
write_priv_set=net_rawaccess
/dev/arp
read_priv_set=none
write_priv_set=none
```

/dev/ip の読み取りおよび書き込みには、net_rawaccess 特権が必要であることに注意してください。/dev/arp は特権を必要としません。

2. **/dev/arp を開き、tcp モジュールと udp モジュールをプッシュします。**

特権は不要です。この方法は、/dev/ip を開いて arp、tcp、および udp モジュールをプッシュするのと同じです。現在、/dev/ip を開くには特権が必要なため、/dev/arp メソッドを推奨します。

デバイス割り当ての管理

pkg:/system/device-allocation パッケージがシステムに存在する場合は、デバイス割り当てを制御できます。デバイス割り当ては一般に、デバイスセキュリティーの追加の層が必要なサイトで実装されます。通常、割り当て可能なデバイスにアクセスするユーザーには承認が必要です。

次のタスクマップは、デバイス割り当ての有効化、構成、およびトラブルシューティングを行うための手順とコマンドオプションを示しています。デフォルトではデバイス割り当ては有効になっていません。デバイス割り当てを有効にしたあとで、デバイスを割り当てるための手順について、85 ページの「デバイスの割り当て」を参照してください。

表 6 デバイス割り当ての管理タスクマップ

タスク	説明	手順
デバイスを割り当て可能にします。 デバイス割り当てを無効にします。	デバイスを一度に 1 人のユーザーに割り当てられるようにします。 すべてのデバイスの割り当て制限を解除します。	80 ページの「デバイス割り当ての有効化または無効化」

タスク	説明	手順
ユーザーによるデバイス割り当てを承認します。	デバイス割り当ての承認をユーザーに与えます。	81 ページの「ユーザーによるデバイス割り当てを承認する方法」
システム上の割り当て可能なデバイスを表示します。	割り当てが可能なデバイスと、そのデバイスの状態を一覧表示します。	82 ページの「デバイスの割り当て情報の表示」
デバイスを強制的に割り当てるか割り当て解除します。	デバイスを、ただちに必要とするユーザーに割り当てるか、割り当て解除します。	82 ページの「デバイスの強制的な割り当てまたは割り当て解除」
デバイスの割り当てプロパティーを変更します。	デバイスを割り当てるための要件を変更します。	83 ページの「割り当て可能なデバイスの変更」
デバイス割り当てを監査します。	デバイス割り当てを監査トレールに記録します	84 ページの「デバイス割り当ての監査」
デバイスクリーンスクリプトを作成します。	物理デバイスからデータを一掃します。	96 ページの「新しいデバイスクリーンスクリプトの作成」

デバイス割り当ての有効化または無効化

注記 - Trusted Extensions がシステムにインストールされて有効になっている場合、`svc:/system/device/allocate` パッケージはすでにインストールされて有効になっています。

これらのアクションを実行するには、Device Security 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

デバイス割り当てサービスを有効にし、このサービスが有効になっていることを確認するには:

```
# svcadm enable svc:/system/device/allocate
svcadm: Pattern 'svc:/system/device/allocate' doesn't match any instances
# pkg install system/device-allocation
...
# svcs -x allocate
svc:/system/device/allocate:default (device allocation)
State: online since September 10, 2016 01:10:11 PM PDT
See: allocate(1)
See: deallocate(1)
See: list_devices(1)
See: device_allocate(8)
See: mkdevalloc(8)
See: mkdevmaps(8)
```

```
See: dminfo(8)
See: device_maps(5)
See: /var/svc/log/system-device-allocate:default.log
Impact: None.
```

デバイス割り当てサービスを無効にするには:

```
# svcadm disable device/allocate
```

ユーザーによるデバイス割り当ての承認

システム管理者は、ユーザーによるデバイスの割り当てを有効にできます。

▼ ユーザーによるデバイス割り当てを承認する方法

始める前に User Security 権利プロファイルが割り当てられている管理者になる必要があります。権利プロファイルには、`solaris.auth.delegate` 承認が含まれている必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **適切な承認とコマンドが入った権利プロファイルを作成します。**

   一般には、`solaris.device.allocate` 承認を含む権利プロファイルを作成します。『Securing Users and Processes in Oracle Solaris 11.4』 の 「How to Create a Rights Profile」の指示に従います。権利プロファイルに、次に示すような適切なプロパティーを指定します。

   - 権利プロファイル名: `Device Allocation`
   - 付与される承認: `solaris.device.allocate`
   - 特権を持つコマンド: `sys_mount` 特権を持つ `mount`、および `sys_mount` 特権を持つ `umount`

2. **(オプション) 権利プロファイルの役割を作成します。**

   『Securing Users and Processes in Oracle Solaris 11.4』 の 「Assigning Rights to Users」の指示に従います。次に示す役割プロパティーを参考にしてください。

   - 役割名: `devicealloc`
   - 役割の完全名: `Device Allocator`
   - 役割の説明: `Allocates and mounts allocated devices`
   - 権利プロファイル: `Device Allocation`

     この権利プロファイルは、この役割に含まれているプロファイルのリストの先頭に存在する必要があります。

3. **権利プロファイルを承認されたユーザーまたは承認された役割に割り当てます。**

次の手順 これらのユーザーにデバイス割り当ての方法を教えます。

リムーバブルメディアの割り当て例は、85 ページの「デバイスを割り当てる方法」を参照してください。

デバイスの割り当て情報の表示

この情報を表示するには、Device Security 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

システム上の割り当て可能デバイスについての情報を表示するには:

```
# list_devices device-name
```

device-name は次のいずれかです。

- `audio`[*n*] – マイクとスピーカー。
- `rmdisk`[*n*] – リムーバブルメディアデバイス (USB など)。
- `sr`[*n*] – CD-ROM ドライブ。
- `st`[*n*] – テープドライブ。

`list_devices` コマンドが次のようなエラーメッセージを返す場合は、デバイス割り当てが有効になっていないか、情報を取得するために必要なアクセス権がありません。

```
list_devices: No device maps file entry for specified device.
```

コマンドを実行するには、デバイス割り当てを有効にし、`solaris.device.revoke` 承認のある役割になります。

デバイスの強制的な割り当てまたは割り当て解除

デバイスを強制的に割り当てるか割り当て解除できます。

これらのアクションを実行するには、`solaris.device.revoke` 承認が割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

自分の役割に適切な承認が含まれているかどうかを確認します。

```
$ auths
```

```
solaris.device.allocate solaris.device.revoke
```

デバイスの強制的な割り当て

強制的な割り当ては、誰かがデバイスの割り当て解除を忘れた場合や、デバイスをただちに使用する必要がある場合などに行います。

`allocate -U` コマンドを使用して、デバイスを必要としているユーザーにデバイスを強制的に割り当てます。この例では、USB フラッシュドライブがユーザー `jdoe` に強制的に割り当てられます。

```
$ allocate -U jdoe
```

デバイスの強制的な割り当て解除

ユーザーが割り当てたデバイスは、プロセスの終了時やそのユーザーのログアウトの際に自動的に割り当て解除されないため、ユーザーがデバイスを割り当て解除するのを忘れた場合は強制的な割り当て解除を使用する必要が生じることがあります。

次のように、`deallocate -f` コマンドを使用してデバイスを強制的に割り当て解除します。

```
$ deallocate -f /dev/lp/printer-1
```

この例では、別のユーザーが割り当てられるように、プリンタが強制的に割り当て解除されます。

割り当て可能なデバイスの変更

このタスクを行うには、デバイス割り当てが有効になっている必要があります。デバイス割り当てを有効にするには、80 ページの「デバイス割り当ての有効化または無効化」を参照してください。`root` 役割になる必要があります。

割り当て可能デバイスを変更するには、`device_allocate` ファイルでデバイスエントリの 5 番目のフィールドを変更して、承認が必要であるかどうかを指定したり、`solaris.device.allocate` 承認を指定したりします。

```
audio;audio;reserved;reserved;solaris.device.allocate;/etc/security/lib/audio_clean
fd0;fd;reserved;reserved;solaris.device.allocate;/etc/security/lib/fd_clean
sr0;sr;reserved;reserved;solaris.device.allocate;/etc/security/lib/sr_clean
```

`solaris.device.allocate` は、デバイスの使用に `solaris.device.allocate` 承認が必要であることを示します。

例 12 任意のユーザーによるデバイス割り当てを許可する

次の例では、システム上の任意のユーザーが任意のデバイスを割り当てることができます。`device_allocate` ファイルの各デバイスエントリ内にある 5 番目のフィールドは、「単価」記号 (@) に変更されました。

```
# pfedit /etc/security/device_allocate
audio;audio;reserved;reserved;@;/etc/security/lib/audio_clean
fd0;fd;reserved;reserved;@;/etc/security/lib/fd_clean
sr0;sr;reserved;reserved;@;/etc/security/lib/sr_clean
…
```

例 13 一部の周辺機器の使用を防止する

次の例では、オーディオデバイスの使用が禁止されています。`device_allocate` ファイルのオーディオデバイスエントリにある 5 番目のフィールドは、アスタリスク (*) に変更されました。

```
# pfedit /etc/security/device_allocate
audio;audio;reserved;reserved;*;/etc/security/lib/audio_clean
fd0;fd;reserved;reserved;solaris device.allocate;/etc/security/lib/fd_clean
sr0;sr;reserved;reserved;solaris device.allocate;/etc/security/lib/sr_clean
…
```

例 14 すべての周辺機器の使用を防止する

次の例では、使用できる周辺機器はありません。`device_allocate` ファイルの各デバイスエントリにある 5 番目のフィールドは、アスタリスク (*) に変更されました。

```
# pfedit /etc/security/device_allocate
audio;audio;reserved;reserved;*;/etc/security/lib/audio_clean
fd0;fd;reserved;reserved;*;/etc/security/lib/fd_clean
sr0;sr;reserved;reserved;*;/etc/security/lib/sr_clean
…
```

デバイス割り当ての監査

デフォルトでは、デバイス割り当てコマンドは、監査クラス `other` の状態です。

注記 - Audit Configuration 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

次のように `ot` 監査クラスを事前選択できます。

```
$ auditconfig -getflags
```

current-flags
```
$ auditconfig -setflags current-flags,ot
```

詳細な手順については、『Managing Auditing in Oracle Solaris 11.4』 の 「How to Preselect Audit Classes」を参照してください。

デバイスの割り当て

デバイス割り当ては、一度に 1 人のユーザーだけが使用できるようにデバイスを予約 (確保) する作業です。マウントポイントが必要なデバイスはマウントする必要があります。次の手順でユーザーに、デバイスを割り当てる方法を示します。

▼ デバイスを割り当てる方法

始める前に 80 ページの「デバイス割り当ての有効化または無効化」の説明に従って、デバイス割り当てを有効にする必要があります。ただし、Trusted Extensions がシステムにインストールされて有効になっている場合、デバイス割り当てはすでに有効になっています。デバイス割り当ては通常有効になっています。

注記 - 承認が必要な場合は、そのユーザーは承認を得ていなければなりません。

1. **デバイスを割り当てます。**
   デバイス名でデバイスを指定します。

   ```
   $ allocate device-name
   ```

2. **コマンドを繰り返して、デバイスが割り当てられていることを確認します。**

   ```
   $ allocate device-name
   allocate. Device already allocated.
   ```

例 15 プリンタを割り当てる

この例では、ユーザーがプリンタを割り当てます。このユーザーが `printer-1` の割り当てを解除するか、このプリンタが強制的にほかのユーザーに割り当てられるまで、ほかのユーザーはこのプリンタを使用できません。

```
$ allocate /dev/lp/printer-1
```

強制的な割り当て解除の例については、82 ページの「デバイスの強制的な割り当てまたは割り当て解除」を参照してください。

例 16 USB フラッシュドライブを割り当てる

この例では、ユーザーが USB フラッシュドライブ `rmdisk1` を割り当てます。

```
$ allocate rmdisk1
```

注意事項 `allocate` コマンドがデバイスを割り当てることができない場合は、コンソールウィンドウにエラーメッセージが表示されます。割り当てのエラーメッセージのリストについては、`allocate`(1) のマニュアルページを参照してください。

▼ 割り当て済みデバイスをマウントする方法

適切な特権が付与されている場合、デバイスは自動的にマウントします。デバイスがマウントに失敗した場合は、この手順に従います。

始める前に デバイスをすでに割り当てている必要があります。81 ページの「ユーザーによるデバイス割り当てを承認する方法」の説明に従って、デバイスをマウントするために必要な特権が割り当てられています。

1. **デバイスの割り当てまたはマウントが行える役割になります。**

   ```
   $ su - role-name
   Password:       <Type role-name password>
   $
   ```

2. **この役割のホームディレクトリにマウントポイントを作成し、このマウントポイントを保護します。**

   この手順を実行する必要があるのは、マウントポイントがはじめて必要になったときだけです。

   ```
   $ mkdir mount-point ; chmod 700 mount-point
   ```

3. **割り当てが可能なデバイスを一覧表示します。**

   ```
   $ list_devices -l
   List of allocatable devices
   ```

4. **デバイスを割り当てます。**

   デバイス名でデバイスを指定します。

   ```
   $ allocate device-name
   ```

5. **デバイスをマウントします。**

   ```
   $ mount -o ro -F filesystem-type device-path mount-point
   ```

`-o ro`	デバイスは読み取り専用としてマウントされることを示します。デバイスを書き込み可能にするには、`-o rw` を使用します。
`-F` *filesystem-type*	デバイスのファイルシステムフォーマットを示します。一般に、CD-ROM は HSFS ファイルシステムでフォーマットされています。
device-path	デバイスへのパスを示します。`list_devices -l` コマンドの出力には、*device-path* が含まれます。
mount-point	ステップ 2 で作成したマウントポイントを示します。

例 17 CD-ROM ドライブを割り当てる

この例では、ユーザーは CD-ROM ドライブ `sr0` の割り当てとマウントが行える役割を引き受けます。このドライブは、HSFS ファイルシステムでフォーマットされています。

```
$ roles
devicealloc
$ su - devicealloc
Password:       <Type devicealloc password>
$ mkdir /home/devicealloc/mymnt
$ chmod 700 /home/devicealloc/mymnt
$ list_devices -l
...
device: sr0 type: sr files: /dev/sr0 /dev/rsr0 /dev/dsk/c0t2d0s0 ...
...
$ allocate sr0
$ mount -o ro -F hsfs /dev/sr0 /home/devicealloc/mymnt
$ cd /home/devicealloc/mymnt ; ls
List of the contents of CD-ROM
```

注意事項 `mount` コマンドがデバイスをマウントできない場合は、「`mount: insufficient privileges`」というエラーメッセージが表示されます。次の点を確認してください。

- `mount` コマンドをプロファイルシェルで実行していることを確認します。役割を引き受けた場合は、その役割にプロファイルシェルがあります。`mount` コマンドでプロファイルを割り当てられたユーザーの場合、プロファイルシェルを作成する必要があります。使用可能なプロファイルシェルのリストについては、`pfexec(1)` のマニュアルページを参照してください。
- 指定されたマウントポイントを所有していることを確認します。このマウントポイントに対する読み取り、書き込み、および実行のアクセス権が必要です。

以上の要件を満たしているにもかかわらず割り当て済みデバイスをマウントできないという場合は、管理者に問い合わせてください。まず、『Securing Users and Processes in Oracle Solaris 11.4』 の 「How to Troubleshoot Rights Assignments」を参照してください。

▼ デバイスの割り当てを解除する方法

割り当てを解除すると、ほかのユーザーもユーザーの使用後にそのデバイスを割り当てて使用できるようになります。

始める前に デバイスをすでに割り当てていなければなりません。詳細は、85 ページの「デバイスを割り当てる方法」を参照してください。

1. **デバイスがマウントされている場合は、デバイスのマウントを解除します。**

```
$ cd $HOME
$ umount mount-point
```

2. **デバイスの割り当てを解除します。**

```
$ deallocate device-name
```

例 18 マイクの割り当てを解除する

この例では、ユーザー `jdoe` がマイク `audio` の割り当てを解除します。

```
$ whoami
jdoe
$ deallocate audio0
```

例 19 CD-ROM ドライブの割り当てを解除する

この例では、Device Allocator 役割が CD-ROM ドライブの割り当てを解除します。次のメッセージが表示されたあとで、CD-ROM が取り出されます。

```
$ whoami
devicealloc
$ cd /home/devicealloc
$ umount /home/devicealloc/mymnt
$ ls /home/devicealloc/mymnt
$
$ deallocate sr0
/dev/sr0:      326o
/dev/rsr0:     326o
…
sr_clean: Media in sr0 is ready.  Please, label and store safely.
```

デバイス保護リファレンス

Oracle Solaris でのデバイスは、カーネルのデバイスポリシーによって保護されます。周辺機器は、デバイス割り当てによって保護できます。デバイス割り当ては、ユーザーレベルで任意に有効化と適用が行われます。

デバイスポリシーコマンド

デバイス管理コマンドは、ローカルファイル上のデバイスポリシーを管理します。デバイスポリシーは特権要件を含むことができます。Device Management および Device Security 権利プロファイルが割り当てられているユーザーはデバイスを管理できます。

次の表は、デバイス管理コマンドを示しています。

表 7 デバイス管理コマンド

コマンド	目的
add_drv(8)	稼働中のシステムに新しいデバイスドライバを追加します。新しいデバイスにデバイスポリシーを追加するオプションを含みます。一般に、このコマンドはデバイスドライバのインストール中にスクリプト内で呼び出されます。
devfsadm(8)	稼働しているシステム上のデバイスとデバイスドライバを管理します。また、デバイスポリシーの読み込みも行います。devfsadm コマンドは、ディスクデバイス、テープデバイス、ポートデバイス、オーディオデバイス、および擬似デバイスに対する /dev リンクのクリーンアップにも使用できます。名前付きドライバのデバイスの再構成も行えます。
Unresolved link to "getdevpolicy8"	1 つ以上のデバイスに関連付けられたポリシーを表示します。このコマンドはどのユーザーでも実行できます。
rem_drv(8)	デバイスまたはデバイスドライバを削除します。
update_drv(8)	既存のデバイスドライバの属性を更新します。デバイスのデバイスポリシーを更新するオプションを含みます。一般に、このコマンドはデバイスドライバのインストール中にスクリプト内で呼び出されます。

デバイスの割り当て

デバイス割り当てによって、データの消失、コンピュータウイルス、セキュリティー侵害などからサイトを保護できます。デバイスポリシーと違い、デバイス割り当ては任意です。デバイス割り当ては、割り当て可能デバイスへのアクセスを制限するのに承認を使用します。

デバイス割り当てのコンポーネント

デバイス割り当てメカニズムのコンポーネントは、次のとおりです。

- svc:/system/device/allocate サービス。詳細は、smf(7) のマニュアルページおよびデバイス割り当てコマンドのマニュアルページを参照してください。

- `allocate`、`deallocate`、`dminfo`、`list_devices` コマンド。詳細は、91 ページの「デバイス割り当てコマンド」を参照してください。
- Device Management および Device Security 権利プロファイル。詳細は、90 ページの「デバイス割り当て権利プロファイル」を参照してください。
- 各割り当て可能デバイスのデバイスクリーンスクリプト。

これらのコマンドとスクリプトは、次のローカルファイルを使用してデバイス割り当てを実装します。

- `/etc/security/device_allocate` ファイル。詳細は、`device_allocate(5)` のマニュアルページを参照してください。
- `/etc/security/device_maps` ファイル。詳細は、`device_maps(5)` のマニュアルページを参照してください。
- ロックファイル。割り当て可能デバイスごとに `/etc/security/dev` ディレクトリに配置します。
- 各割り当て可能デバイスに関連付けられたロックファイルの変更後の属性。

デバイス割り当てサービス

`svc:/system/device/allocate` サービスは、デバイス割り当てを制御します。このサービスはデフォルトで無効になっています。

デバイス割り当て権利プロファイル

デバイスおよびデバイス割り当てを管理するには、Device Management および Device Security 権利プロファイルが必要です。

これらの権利プロファイルには、次の承認が含まれています。

- `solaris.device.allocate` – デバイスを割り当てるために必要
- `solaris.device.cdrw` – CD-ROM の読み取りと書き込みを行うために必要
- `solaris.device.config` – デバイスの属性を構成するために必要
- `solaris.device.mount.alloptions.fixed` – 固定デバイスのマウント時にマウントオプションを指定するために必要
- `solaris.device.mount.alloptions.removable` – リムーバブルデバイスのマウント時にマウントオプションを指定するために必要
- `solaris.device.mount.fixed` – 固定デバイスをマウントするために必要
- `solaris.device.mount.removable` – リムーバブルデバイスをマウントするために必要

- `solaris.device.revoke` – デバイスを取り消すか、または再利用するために必要

デバイス割り当てコマンド

大文字のオプションが指定された `allocate`、`deallocate`、および `list_devices` コマンドは管理用コマンドです。それ以外ではこれらのコマンドはユーザーコマンドです。次の表は、デバイス割り当てコマンドを示しています。

表 8 デバイス割り当てコマンド

コマンドのマニュアルページ	目的
`allocate`(1)	1 人のユーザーだけが使用できるように割り当て可能デバイスを予約します。 デフォルトでは、ユーザーがデバイスを割り当てるには `solaris.device.allocate` 承認が必要です。ユーザー承認を必要としないように、`device_allocate` ファイルを変更することもできます。そのように変更した場合、システム上のどのユーザーでもデバイスの使用割り当てを要求できます。
`deallocate`(1)	デバイスから割り当て予約を削除します。
Unresolved link to "dminfo8"	デバイスタイプ、デバイス名、またはフルパス名を指定して、割り当て可能デバイスを検索します。
`list_devices`(1)	割り当て可能なデバイスのステータスを表示します。 `device_maps` ファイルにリストされたデバイスに関連付けられている、デバイス特殊ファイルを列挙します。 `-U` オプションがあると、割り当て可能なデバイス、または指定されたユーザー ID に割り当てられているデバイスを一覧表示します。このオプションを使用すると、別のユーザーに割り当てることができるデバイスまたは割り当て済みのデバイスを確認できます。このコマンドを実行するには、ユーザーまたは役割に `solaris.device.revoke` 承認がなければなりません。

割り当てコマンドの承認

デフォルトでは、ユーザーが割り当て可能デバイスを予約するには `solaris.device.allocate` 承認を必要とします。`solaris.device.allocate` 承認を含める権利プロファイルを作成する方法については、81 ページの「ユーザーによるデバイス割り当てを承認する方法」を参照してください。

管理者がデバイスの割り当て状態を変更するには、`solaris.device.revoke` 承認が必要です。たとえば、`allocate` および `list_devices` コマンドの `-U` オプションや、`deallocate` コマンドの `-F` オプションは、`solaris.device.revoke` 承認が必要です。

詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Selected Commands That Require Authorizations」を参照してください。

割り当てエラー状態

deallocate コマンドが割り当ての解除に失敗する場合、または allocate コマンドが割り当てに失敗する場合は、デバイスは「割り当てエラー状態」になります。割り当て可能デバイスが割り当てエラー状態となった場合、そのデバイスの割り当てを強制的に解除する必要があります。割り当てエラー状態を処理できるのは、Device Management 権利プロファイルまたは Device Security 権利プロファイルを持つユーザーまたは役割だけです。

F オプションを指定した -deallocate コマンドは、割り当て解除を強制します。あるいは、allocate -U を実行してデバイスを特定のユーザーに割り当てることもできます。いったんデバイスが割り当てられると、発生したエラーメッセージを調査できます。デバイスに関する問題が解決されたあとで、そのデバイスの割り当てを強制的に解除できます。

device_maps ファイル

システムのデバイス割り当てを作成すると、デバイスマップが作成されます。/etc/security/device_maps ファイルには、割り当て可能な各デバイスに関連付けられたデバイス名、デバイスタイプ、およびデバイス特殊ファイルが含まれています。

直観的にはわかりにくい各デバイスのために、device_maps ファイルはデバイス特殊ファイルのマッピングを定義します。このファイルによって、プログラムはどのデバイス特殊ファイルがどのデバイスに割り当てられているかを検出できます。たとえば、dminfo コマンドを使用すると、デバイス名、デバイスの種類およびデバイス特殊ファイルを取得して、割り当て可能なデバイスを設定するときに指定できます。dminfo コマンドは、device_maps ファイルを使用してデバイス割り当て情報を報告します。

各デバイスは、次の形式の 1 行のエントリで表されます。

device-name:*device-type*:*device-list*

例 20 device_maps エントリの例

次の例は、device_maps ファイルのエントリを示したものです。

```
audio0:\
audio:\
/dev/audio /dev/audioctl /dev/dsp /dev/dsp0 /dev/mixer0 /dev/sound/0
/dev/sound/0ctl /dev/sound/audio810\:0mixer /dev/sound/audio810\:0dsp
/dev/sound/audio810\:0 /dev/sound/audio810\:0ctl
```

device_maps ファイル内の行は、バックスラッシュ (\) で終了することにより、エントリを次の行に続けることができます。コメントも挿入できます。ポンド記号 (#) を付けると、1 つ前の行末にバックスラッシュのない改行まで、それに続くすべてのテ

キストはコメントになります。どのフィールドでも先行ブランクと後続ブランクを使用できます。フィールドの定義は次のとおりです。

device-name デバイスの名前を指定します。現在のデバイス名のリストについては、82 ページの「デバイスの割り当て情報の表示」を参照してください。

device-type 汎用デバイスタイプを指定します。汎用名は、`st`、`fd`、`rmdisk`、`audio` などの、デバイスのクラスの名前です。*device-type* では、関連するデバイスが論理的にグループ化されます。

device-list 物理デバイスに関連付けられたデバイス特殊ファイルを一覧表示します。*device-list* には、特定のデバイスにアクセスできるすべての特殊ファイルが含まれている必要があります。リストが不完全な場合は、悪意を持ったユーザーでも個人情報を入手または変更できます。*device-list* フィールドには、`/dev` ディレクトリに入っているデバイスファイルを指定します。

`device_allocate` ファイル

デバイスを割り当て可能から割り当て不可能に変更したり、新しいデバイスを追加したりするために、`/etc/security/device_allocate` ファイルを変更できます。

`device_allocate` ファイル内のエントリは、デバイスが割り当て可能であると特に記載していないかぎり、そのデバイスが割り当て可能であることを示しません。

`device_allocate` ファイルでは、各デバイスは次の形式の 1 行のエントリで表されます。

device-name`;`*device-type*`;reserved;reserved;`*auths*`;`*device-exec*

次の例は、`device_allocate` ファイルのサンプルを示しています。

```
st0;st;;;;/etc/security/lib/st_clean
fd0;fd;;;;/etc/security/lib/fd_clean
sr0;sr;;;;/etc/security/lib/sr_clean
audio;audio;;;*;/etc/security/lib/audio_clean
```

`audio` デバイスエントリの 5 番目のフィールドにあるアスタリスク (*) に注意してください。

`device_allocate` ファイル内の行は、バックスラッシュ (\) で終了することにより、エントリを次の行に続けることができます。コメントも挿入できます。ポンド記号 (#) を付けると、1 つ前の行末にバックスラッシュのない改行まで、それに続くすべてのテキストはコメントになります。どのフィールドでも先行ブランクと後続ブランクを使用できます。フィールドの定義は次のとおりです。

device-name デバイスの名前を指定します。現在のデバイス名のリストについては、82 ページの「デバイスの割り当て情報の表示」を参照してください。

device-type 汎用デバイスタイプを指定します。汎用名は、`st`、`fd`、`sr` などのデバイスクラス名です。*device-type* では、関連するデバイスが論理的にグループ化されます。デバイスを割り当て可能にするときは、`device_maps` ファイルの *device-type* フィールドからデバイス名を取得します。

`reserved` Oracle では、`reserved` で示される 2 つのフィールドを将来の使用に予約しています。

auths デバイスが割り当て可能であるかどうかを指定します。このフィールドにアスタリスク (*) が入っている場合は、デバイスが割り当て不可能であることを示します。承認を示す文字列が入っている場合や、空の場合は、デバイスが割り当て可能であることを示します。たとえば、*auths* フィールド内の文字列 `solaris.device.allocate` は、そのデバイスを割り当てるには `solaris.device.allocate` 承認が必要であることを示します。このフィールドに単価記号 (@) が入っている場合は、どのユーザーでもそのデバイスを割り当てることができることを示します。

device-exec 割り当てプロセス中にクリーンアップやオブジェクト再利用防止などの特殊処理のために呼び出されるスクリプトのパス名を指定します。*device-exec* スクリプトは、デバイスに対して `deallocate` コマンドを実行するたびに実行されます。

たとえば、`sr0` デバイスについての次のエントリは、CD-ROM ドライブが `solaris.device.allocate` 承認を得たユーザーによって割り当て可能であることを示します。

```
sr0;sr;reserved;reserved;solaris.device.allocate;/etc/security/lib/sr_clean
```

デフォルトのデバイスとそれらの定義された特性を受け入れることを決定できます。新しいデバイスをインストールしたあとで、エントリを変更できます。使用前に割り当てが必要なデバイスはすべて、そのデバイスのシステムの `device_allocate` ファイルと `device_maps` ファイルで定義する必要があります。現在は、カートリッジテープドライブ、CD-ROM ドライブ、リムーバブルメディアデバイス、およびオーディオチップが、割り当て可能とみなされます。これらのデバイスタイプには、デバイスクリーンスクリプトが用意されています。

注記 - Xylogics および Archive テープドライブもまた、SCSI デバイスのために提供されている `st_clean` スクリプトを使用します。端末、グラフィックスタブレット、その他の割り当て可能なデバイスなどのほかのデバイスについては、ユーザー独自のデバイスクリーンスクリプトを作成する必要があります。このスクリプトは、そのデバイスタイプのオブジェクト再利用の要件を満たしている必要があります。

デバイスクリーンスクリプト

デバイス割り当てによって、セキュリティー監査者がオブジェクト再利用の要件と呼ぶものの一部が満たされます。デバイスクリーンスクリプトは、使用可能なすべてのデータを、再利用の前に物理デバイスから消去するというセキュリティー要件に対応します。データのクリアは、そのデバイスが別のユーザーによって割り当て可能になる前に実行されます。デフォルトでは、カートリッジテープドライブ、CD-ROM ドライブ、オーディオデバイスには、Oracle Solaris で提供されるデバイスクリーンスクリプトが必要です。このセクションでは、デバイスクリーンスクリプトが実行する処理について説明します。

テープ用のデバイスクリーンスクリプト

`st_clean` デバイスクリーンスクリプトでは、3 つのテープデバイスがサポートされます。

- SCSI ¼ インチテープ
- アーカイブ ¼ インチテープ
- オープンリール ½ インチテープ

`st_clean` スクリプトは、`mt` コマンドの `rewoffl` オプションを使用してデバイスをクリーンアップします。詳細は、`mt(1)` のマニュアルページを参照してください。このスクリプトは、システムブート中に実行されると、デバイスを照会し、デバイスがオンラインであるかどうかを確認します。デバイスがオンラインの場合、スクリプトは、そのデバイスにメディアが挿入されているかどうかを調べます。¼ インチのテープデバイスにメディアが挿入されていた場合、このデバイスは割り当てエラー状態になります。この場合、管理者はそのデバイスを手動でクリーンアップする必要があります。

通常のシステム操作中に、`deallocate` コマンドを対話型モードで実行すると、メディアを取り出すように求めるプロンプトが表示されます。割り当て解除は、デバイスからメディアが取り出されるまで見送られます。

CD-ROM ドライブ用のデバイスクリーンスクリプト

CD-ROM ドライブ用の **`sr_clean`** デバイスクリーンスクリプトが用意されています。

スクリプトは、`eject` コマンドを使用してドライブからメディアを取り出します。`eject` コマンドが失敗すると、デバイスは割り当てエラー状態になります。詳細は、`eject(1)` のマニュアルページを参照してください。

オーディオ用のデバイスクリーンスクリプト

オーディオデバイスは、audio_clean スクリプトを使用してクリーンアップします。スクリプトは、AUDIO_GETINFO ioctl システムコールを実行してデバイスを読み取ります。AUDIO_SETINFO ioctl システムコールを実行してデバイス構成をデフォルトにリセットします。

新しいデバイスクリーンスクリプトの作成

システムに新しく割り当て可能デバイスを追加する場合は、独自のデバイスクリーンスクリプトを作成する必要があります。deallocate コマンドは、デバイスクリーンスクリプトにパラメータを渡します。次に示すように、パラメータはデバイス名を含む文字列です。詳細は、device_allocate(5) のマニュアルページを参照してください。

clean-script `-[I|i|f|S]` *device-name*

デバイスクリーンスクリプトは、成功時には「0」を、失敗時には「0」より大きな値を返す必要があります。オプション -I、-f、および -S は、スクリプトの実行モードを決定します。

`-I` システムのブート時にのみ必要です。すべての出力は、システムコンソールに送られます。失敗した場合や、メディアを強制的に取り出せない場合は、デバイスを割り当てエラー状態にします。

`-i` 出力が抑止される点を除き、-I オプションと同じです。

`-f` 強制的なクリーンアップ用。このオプションは対話型であり、ユーザーがプロンプトに応答するものとみなします。このオプションが付いたスクリプトは、クリーンアップの一部に失敗した場合に、クリーンアップ全体を完了しようとします。

`-S` 標準クリーンアップ。このオプションは対話型であり、ユーザーがプロンプトに応答するものとみなします。

♦ ♦ ♦ 第 5 章

5 ウイルスのスキャン

この章では、ウイルス対策ソフトウェアの使用についての情報を提供します。この章の内容は次のとおりです。

ウイルススキャンについて

データは、各種スキャンエンジンを使用するスキャンサービス vscan によってウイルスから保護されます。スキャンエンジンとは、外部ホストに常駐するサードパーティーのアプリケーションであり、ファイルで既知のウイルスを調べます。ファイルがウイルススキャンの候補となるのは、そのファイルシステムが vscan サービスをサポートし、そのサービスが有効になっていて、ファイルのタイプが対象外になっていない場合です。そして、ファイルが最新のウイルス定義でまだスキャンされていない場合、またはファイルが最後にスキャンされた以後に変更されている場合、ファイルのオープンおよびクローズ操作中にウイルススキャンが実行されます。

vscan サービスは、複数のスキャンエンジンを使用するように構成できます。vscan サービスで 2 つ以上のスキャンエンジンを使用することをお勧めします。ウイルススキャンの要求は、使用できるすべてのスキャンエンジンに配信されます。表9に、最新のパッチを使って構成された場合にサポートされるスキャンエンジンを示します。

表 9 ウイルス対策スキャンエンジンソフトウェア

ウイルス対策ソフトウェア	ICAP のサポート
Symantec Critical System Protection 5.2.9	サポートされています
Symantec Protection Engine 7.0	SPARC でのみサポートされています
Trend Micro Deep Security Agent 9.0 SP1	サポートされています
Trend Micro Deep Security Agent (DSA) 8.0 SP1	x86 でのみサポートされています

ウイルス対策ソフトウェア	ICAP のサポート
McAfee VirusScan Command Line scanner (VSCL) 6.06	サポートされています
McAfee Anti-Malware Engine 5800	サポートされています
McAfee Anti-Malware Engine 5700	x86 でのみサポートされています

vscan サービスについて

リアルタイムスキャン方法の利点は、ファイルが使用される前に最新のウイルス定義でスキャンされることです。この方法を使用することで、ウイルスがデータを危険にさらす前にそれらを検出できます。

次は、ウイルススキャンプロセスについて説明したものです。

1. ユーザーがクライアントでファイルを開くと、vscan サービスにより、そのファイルが最新のウイルス定義ですでにスキャンされているかどうか、およびそのファイルが最後にスキャンされた以後に変更されたかどうかに基づいて、ファイルのスキャンが必要かどうかが判断されます。
   - ファイルのスキャンが必要な場合、そのファイルはscan engineに転送されます。スキャンエンジンへの接続に失敗した場合、そのファイルは別のスキャンエンジンに送信されます。使用できるスキャンエンジンがない場合、ウイルススキャンは失敗し、そのファイルへのアクセスが拒否される可能性があります。
   - ファイルのスキャンが必要ない場合、クライアントはそのファイルへのアクセスを許可されます。
2. スキャンエンジンが最新のウイルス定義を使用してファイルをスキャンします。
   - ウイルスが検出された場合、そのファイルには隔離されたことを示すマークが付けられます。隔離されたファイルの読み取り、実行、または名前の変更はできませんが、削除はできます。システムログには、隔離されたファイルの名前とウイルスの名前が記録され、さらに監査が有効になっていた場合は、同じ情報が含まれた監査レコードが作成されます。
   - ファイルが感染していない場合、そのファイルはスキャンスタンプでタグ付けされ、クライアントはそのファイルへのアクセスを許可されます。

vscan サービスの使用

ファイルのウイルススキャンは、次の要件が満たされたときに使用できます。

- ウイルススキャンパッケージがインストールされている。

- 1 つ以上のスキャンエンジンがインストールされ、構成されている。
- ファイルが、ウイルススキャンをサポートしているファイルシステム上に存在する。
- そのファイルシステムでウイルススキャンが有効になっている。
- vscan サービスが有効になっている。
- vscan サービスが、指定されたファイルタイプのファイルをスキャンするように構成されている。

次の表は、vscan サービスを設定するために行うタスクを示しています。

タスク	説明	手順
スキャンエンジンをインストールします。	表9に示す、サポートされているサードパーティー製品を 1 つ以上インストールし、構成します。	製品のドキュメントを参照してください。
virus-scan IPS パッケージをインストールします。	ウイルススキャンパッケージをインストールします。	99 ページの「ウイルススキャンソフトウェアをインストールする方法」
ファイルシステムでウイルススキャンを使用できるようにします。	ZFS ファイルシステムでのウイルススキャンを有効にします。デフォルトでは、スキャンは無効になっています。	100 ページの「ファイルシステムでウイルススキャンを有効にする方法」
vscan サービスを有効にします。	スキャンサービスを開始します。	100 ページの「vscan サービスを有効にする方法」
スキャンエンジンを vscan サービスに追加します。	特定のスキャンエンジンを vscan サービスに組み込みます。	101 ページの「スキャンエンジンを追加する方法」
vscan サービスを構成します。	vscan プロパティーを表示および変更します。	101 ページの「vscan プロパティーを表示する方法」 102 ページの「vscan プロパティーを変更する方法」
特定のファイルタイプ向けに vscan サービスを構成します。	スキャンに組み込んだり、スキャンから除外したりするファイルタイプを指定します。	102 ページの「ウイルススキャンからファイルを除外する方法」

▼ ウイルススキャンソフトウェアをインストールする方法

始める前に ソフトウェアインストールに関連する権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』の「Using Your Assigned Administrative Rights」を参照してください。

1. **ウイルススキャンパッケージをインストールします。**

```
$ pkg install virus-scan
```

2. **(オプション) インストールを確認します。**

```
$ pkg list virus-scan
NAME (PUBLISHER)                          VERSION                    IFO
service/storage/virus-scan                0.11.4.0.............      i--

$ pkg verify -v virus-scan
PACKAGE                                                             STATUS
pkg://solaris/service/storage/virus-scan                              OK
```

▼ ファイルシステムでウイルススキャンを有効にする方法

ファイルのウイルススキャンを可能にするには、ファイルシステムコマンドを使用します。たとえば、ZFS ファイルシステムをウイルススキャンに組み込むには、zfs(8) コマンドを使用します。

ZFS ファイルシステムでは、一部の管理タスクを特定のユーザーに委託できます。委託管理の詳細は、『Managing ZFS File Systems in Oracle Solaris 11.4』 の 第 9 章, 「Oracle Solaris ZFS Delegated Administration」 を参照してください。

始める前に ZFS File System Management または ZFS Storage Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」 を参照してください。

- **ZFS ファイルシステム、たとえば `pool/volumes/vol1` でのウイルススキャンを有効にします。**

```
$ zfs set vscan=on path/pool/volumes/vol1
```

▼ vscan サービスを有効にする方法

始める前に VSCAN Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」 を参照してください。

- **ウイルススキャンサービスを有効にします。**

```
$ svcadm enable vscan
```

▼ スキャンエンジンを追加する方法

始める前に VSCAN Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

- **デフォルトのプロパティーを使用してスキャンエンジンを vscan サービスに追加するには、次のように入力します。**

  ```
  $ vscanadm add-engine engine-ID
  ```

  詳細は、vscanadm(8) のマニュアルページを参照してください。

▼ vscan プロパティーを表示する方法

始める前に VSCAN Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

- **すべてのスキャンエンジンまたは特定のスキャンエンジンについて、vscan サービスのプロパティーを表示します。**

  - **特定のスキャンエンジンのプロパティーを表示するには、次を入力します。**

    ```
    $ vscanadm get-engine engine-ID
    ```

  - **すべてのスキャンエンジンのプロパティーを表示するには、次を入力します。**

    ```
    $ vscanadm get-engine
    ```

  - **vscan サービスのいずれかのプロパティーを表示するには、次のように入力します。**

    ```
    $ vscanadm get -p property
    ```

    ここで、*property* は vscanadm(8) コマンドのマニュアルページに記載されているパラメータの 1 つです。

    たとえば、スキャンできるファイルの最大サイズを表示する場合は次を入力します。

    ```
    $ vscanadm get max-size
    ```

▼ vscan プロパティーを変更する方法

特定のスキャンエンジンのプロパティーや vscan サービスの一般プロパティーを変更できます。多くのスキャンエンジンではスキャンできるファイルのサイズが制限されているため、vscan サービスの max-size プロパティーをスキャンエンジンの最大許容サイズ以下の値に設定する必要があります。その際、最大サイズよりも大きく、そのためにスキャンされないファイルをアクセス可能にするかどうかを定義します。

始める前に VSCAN Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **vscanadm show コマンドを使用して現在のプロパティーを表示します。**

```
$ vscanadm show
max-size=1GB
max-size-action=allow
timeout=30
...
```

2. **タイムアウト値を設定します。**
スキャン時間がこのタイムアウト値を超えると、ファイルへのアクセスが拒否されます。

```
$ vscanadm set -p timeout=60
```

3. **ウイルススキャンの最大サイズを、たとえば 128M バイトに設定します。**

```
$ vscanadm set -p max-size=128M
```

4. **そのサイズのせいでスキャンされないファイルへのアクセスが拒否されるように指定します。**

```
$ vscanadm set -p max-size-action=deny
```

詳細は、vscanadm(8) のマニュアルページを参照してください。

▼ ウイルススキャンからファイルを除外する方法

ウイルス対策保護を有効にした場合、特定のタイプのすべてのファイルがウイルススキャンから除外されるように指定できます。vscan サービスはシステムのパフォーマンスに影響を与えるため、特定のファイルタイプをウイルススキャンの対象とすることで、システムリソースを節約できます。

始める前に VSCAN Management 権利プロファイルが割り当てられている管理者になる必要があります。詳細は、『Securing Users and Processes in Oracle Solaris 11.4』 の 「Using Your Assigned Administrative Rights」を参照してください。

1. **ウイルススキャンに含まれているすべてのファイルタイプの一覧を表示します。**

```
$ vscanadm get -p types
```

2. **ウイルススキャンの対象となるファイルのタイプを指定します。**

   - **特定のファイルタイプ、たとえば JPEG タイプをウイルススキャンから除外します。**

```
$ vscanadm set -p types=-jpg,+*
```

   - **特定のファイルタイプ、たとえば実行可能ファイルをウイルススキャンに含めます。**

```
$ vscanadm set -p types=+exe,-*
```

   詳細は、vscanadm(8) のマニュアルページを参照してください。

システムおよびデバイスの用語集

この用語集には、オペレーティングシステムのさまざまな部分で用法が異なっていたり、Oracle Solaris ではほかのオペレーティングシステムとは異なる意味を持っていたりするために、あいまいになる可能性のある用語が収録されています。

権利 すべての機能を持つスーパーユーザーの代替アカウント。ユーザー権利の管理およびプロセス権利の管理で、組織はスーパーユーザーの特権を分割して、ユーザーまたは役割に割り当てることができます。Oracle Solaris の権利は、カーネル特権、承認、または特定の UID や GID としてプロセスを実行する機能として実装されています。権利は権利プロファイルにまとめることができます。

権利プロファイル プロファイルとも呼ばれます。役割またはユーザーに割り当てることができるセキュリティーオーバーライドの集合。権利プロファイルには、承認、特権、セキュリティー属性が割り当てられたコマンド、および補足プロファイルと呼ばれるその他の権利プロファイルを含めることができます。

最小特権 指定されたプロセスにスーパーユーザー権限のサブセットのみを提供するセキュリティーモデル。最小特権モデルでは、通常のユーザーに、ファイルシステムのマウントやファイルの所有権の変更などの個人の管理タスクを実行できる十分な特権を割り当てます。これに対して、プロセスは、スーパーユーザーの完全な権限 (つまり、すべての特権) ではなく、タスクを完了するために必要な特権のみで実行されます。バッファーオーバーフローなどのプログラミングエラーによる損害を、保護されたシステムファイルの読み取りまたは書き込みやシステムの停止などの重要な機能にはアクセスできない root 以外のユーザーに封じ込めることができます。

信頼できるユーザー ある程度の信頼レベルで管理タスクを実行できるように決定されたユーザー。一般に、管理者は最初に信頼できるユーザーのログインを作成してから、ユーザーの信頼および能力レベルに合致した管理者権利を割り当てます。その後、これらのユーザーはシステムの構成および保守を支援します。特権ユーザーとも呼ばれます。

スーパーユーザーモデル コンピュータシステムにおける典型的な UNIX セキュリティーモデル。スーパーユーザーモデルでは、管理者は絶対的なシステム制御権を持ちます。一般に、システム管理のために 1 人のユーザーがスーパーユーザー (`root`) になり、すべての管理作業を行える状態となります。

セキュリティー属性 セキュリティーポリシーをオーバーライドし、スーパーユーザー以外のユーザーによって実行されても成功する管理コマンドを有効にします。スーパーユーザーモデ

ルでは、setuid root プログラムと setgid プログラムがセキュリティー属性です。これらの属性がコマンドで指定されると、そのコマンドがどのようなユーザーによって実行されているかにかかわらず、コマンドは正常に処理されます。特権モデルでは、セキュリティー属性として setuid root プログラムがカーネル特権およびその他の権利によって置き換えられます。特権モデルは、スーパーユーザーモデルと互換性があります。このため、特権モデルは setuid プログラムと setgid プログラムをセキュリティー属性として認識します。

セキュリティーポリシー ポリシーを参照してください。

デバイスの割り当て システムに対するユーザーレベルでのデバイス保護。デバイス割り当ては、一度に 1 人のユーザーだけが使用できるようにデバイスを設定する作業です。デバイスデータは、デバイスが再使用される前に消去されます。誰にデバイス割り当てを許可するかは、承認を使用して制限できます。

デバイスポリシー システムに対するカーネルレベルでのデバイス保護。デバイスポリシーは、2 つの特権セットとしてデバイスに実装されます。この 1 つはデバイスに対する読み取り権を制御し、もう 1 つはデバイスに対する書き込み権を制御します。ポリシーも参照してください。

特権 1. 一般に、コンピュータシステム上で通常のユーザーの能力を超える操作を実行する能力または機能。スーパーユーザー特権は、スーパーユーザーに付与されているすべての権利です。特権ユーザーまたは特権アプリケーションは、追加の権利が付与されているユーザーまたはアプリケーションです。

2. Oracle Solaris システムにおいてプロセスに対する個々の権利。特権を使用すると、root を使用するよりもきめ細かなプロセス制御が可能です。特権の定義と適用はカーネルで行われます。特権は、プロセス特権やカーネル特権とも呼ばれます。特権の詳細は、privileges(7) のマニュアルページを参照してください。

特権モデル コンピュータシステムにおいてスーパーユーザーモデルより厳密なセキュリティーモデル。特権モデルでは、プロセスの実行に特権が必要です。システムの管理は、管理者が各自のプロセスで与えられている特権に基づいて複数の個別部分に分割できます。特権は、管理者のログインプロセスに割り当てることも、特定のコマンドだけで有効なように割り当てることも可能です。

パスワードポリシー パスワードの生成に使用できる暗号化アルゴリズム。パスワードをどれぐらいの頻度で変更すべきか、パスワードの試行を何回まで認めるかといったセキュリティー上の考慮事項など、パスワードに関連した一般的な事柄を指すこともあります。セキュリティーポリシーにはパスワードが必要です。パスワードポリシーでは、AES アルゴリズムを使用してパスワードを暗号化することを要求したり、パスワードの強度に関連したそれ以上の要件を設定したりすることもできます。

ポリシー 一般には、意思やアクションに影響を与えたり、これらを決定したりする計画や手続き。コンピュータシステムでは、多くの場合セキュリティーポリシーを指します。実

際のサイトのセキュリティーポリシーは、処理される情報の重要度や未承認アクセスから情報を保護する手段を定義する規則セットです。たとえば、セキュリティーポリシーで、システムの監査、使用するシステムデバイスの割り当て、6 週ごとのパスワード変更といったことを設定できます。

Oracle Solaris OS の特定の領域におけるポリシーの実装については、デバイスポリシーおよびパスワードポリシーを参照してください。

Secure Shell セキュリティー保護されていないネットワークを通して、セキュアなリモートログインおよびその他のセキュアなネットワークサービスを使用するための特別なプロトコル。

索引

か

さ

た

な

は

ま

や

ら

わ

A

B

S

T

U

V

X