Hewlett Packard Enterprise

知っておくべき セキュリティ対策

-HP-UX のセキュリティを極める-

自分のシステムのセキュリティに絶対の自信を持っている――という IT エンジニアはそれほど多くないでしょう。「セキュリティ」と一言で言っても、IT システムにまつわる多種多様な要素に個別のセキュリティ対策が必要とされます。例えば「SSL だけは自信がある」、「ファイアウォール設定には時間を掛けた」という状態ではセキュリティ強度は向上しません。各階層のそれぞれについてバランス良くセキュリティ強化を施す「多層防御」こそが、セキュリティのかなめと言えます。そこで本連載では、HP-UX に備わるセキュリティ機能のうち、これまであまり紹介されていないものや新しい機能にスポットをあてて紹介します。

《連載期間：2008 年 1 月～2008 年 7 月》

―目次―

して、HP-UX 11i v3 では「Role-based Access Control（RBAC）」を提供しています。RBAC により、root が持つすべての権限のうち個々の作業に必要な権限だけをユーザに付与することで、root アカウントの乱用によるセキュリティ・リスクの増加を抑えられるのです。

最終回　LDAP によるアカウント統合化

HP-UX を搭載したサーバーの台数が増えてくると、アカウントの管理が懸案となります。例えば数台、数 10 台といったサーバーのそれぞれにユーザ・アカウント登録を行い、整合性を保持するのは大変面倒なうえ、古いアカウントの放置による脆弱性のリスクも生じます。これまで UNIX 環境では NIS や NIS+による一元管理が一般的でしたが、セキュリティ脆弱性の問題などから最近では LDAP（Lightweight Directory Access Protocol）ベースのディレクトリ・サービスがアカウント統合に利用されつつあります。そこで本連載の最終回となる今回は、HP-UX に備わる LDAP クライアント機能「LDAP-UX」を利用したアカウント一元管理の実際を紹介します。

第 1 回

セキュリティの『多層防御』と HP-UX

2008 年 1 月　テクニカルライター 吉川和巳

自分のシステムのセキュリティに絶対の自信を持っている――という IT エンジニアはそれほど多くないだろう。J-SOX の対応など IT 統制の対策として IT システムのセキュリティ強化の優先順位が依然として高い現状であるが、「セキュリティ」と一言で言っても、IT システムにまつわる多種多様な要素に個別のセキュリティ対策が必要とされる。例えば「SSL だけは自信がある」、「ファイアウォール設定には時間を掛けた」という状態ではセキュリティ強度は向上しない。各階層のそれぞれについてバランス良くセキュリティ強化を施す「多層防御」こそが、セキュリティのかなめと言えるのである。そこで本連載では、HP-UX に備わるセキュリティ機能のうち、これまであまり紹介されていないものや新しい機能にスポットをあてて紹介していこう。

セキュリティのかなめは「多層防御」

自分のシステムのセキュリティに絶対の自信を持っている――という IT エンジニアはそれほど多くないだろう。「セキュリティ」と一言で言っても、ネットワークをはじめ、ハードウェア、OS、Web サーバーやミドルウェア、データベース、アプリケーション、運用体制や業務ポリシーなど、IT システムにまつわる多種多様な要素のそれぞれについて、個別のセキュリティ対策が必要とされる。例えば、OS 上で動作する Web サーバーやデータベースなどのセキュリティ設定ばかりに時間を取られ、そこだけ万全の対策を施していても、OS 自体の設定不備があれば、IT システム全体としては脆弱なものとなり、IT 統制がなされているとは言えないだろう。そのうえ、個々のセキュリティ対策について「ここまでやれば万全」というポイントを見極めるのは容易ではない。「現実的な実装効率や運用効率」と「セキュリティ強度」のトレードオフがつねに要求される。

このようにセキュリティの確保は一筋縄ではいかない作業だが、ひとつ言えるのは「セキュリティのかなめは多層防御」ということだ。つまり、IT システムを構成する各階層のそれぞれについてバランス良くセキュリティ強化を施すことが、どのような IT システムでも不可欠となる。「SSL だけは自信がある」、「ファイアウォール設定には時間を掛けた」という状態では、IT システム

全体のセキュリティ強度を高めることはできない。たとえて言うならば、セキュリティに関しては「スペシャリスト」ではなく「ゼネラリスト」が求められるのである。

HP-UX における多層防御

よって HP-UX を扱う IT エンジニアにとっては、この多層防御を実現する上で HP-UX がどのようなセキュリティ機能を備えているか、ひととおり理解しておくことが肝要となる。HP-UX では、大きく分けて以下の 3 種類の観点から多層防御を実現する各種セキュリティ機能を提供している。また、これらの機能は HP-UX のオペレーティング環境に含まれており、セキュリティ強化に伴う余計なコストを最小限に抑えることができる。

- データ保護
- システム保護
- アイデンティティ保護

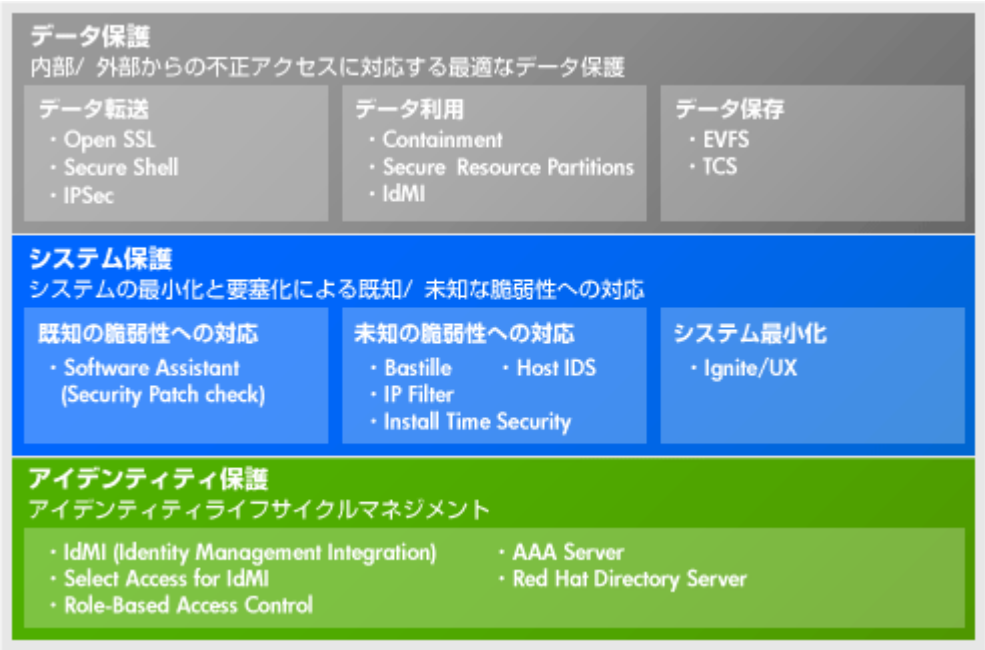

図 1：HP-UX に備わる 3 種類のセキュリティ機能

本連載では、これらのセキュリティ機能のうち、これまであまり紹介されていないものや新しい機能にスポットをあてて紹介していく予定だ。まずは、今後取り上げる予定のセキュリティ機能について簡単に説明しておきたい。

データ保護

Protected Systems Web Server - Web サーバーのセキュリティ強化アーキテクチャ

Protected Systems-Web Server（PS-Webserver）は、外部からの攻撃を受けやすい Web サーバーを強固なセキュリティで保護する機能である。具体的には、HP-UX の内部をいくつかのコンパートメント（区域）に分割し、それぞれのコンパートメントに Web サーバーやアプリケーション・サーバーを分散配置する。個々のコンパートメントは単独のサーバーに匹敵する高い隔離性を備えているため、万が一に Web サーバーへのクラッキングが成功したとしても、クラッカーがほかのコンパートメントに侵入することは非常に困難となる。PS-Webserver は、こうしたコンパートメントを利用したセキュアな Web サーバーを構築す

るための、リファレンス・アーキテクチャを提供する。また、セットアップ・スクリプトが付属しており、導入が容易な点も特徴だ。

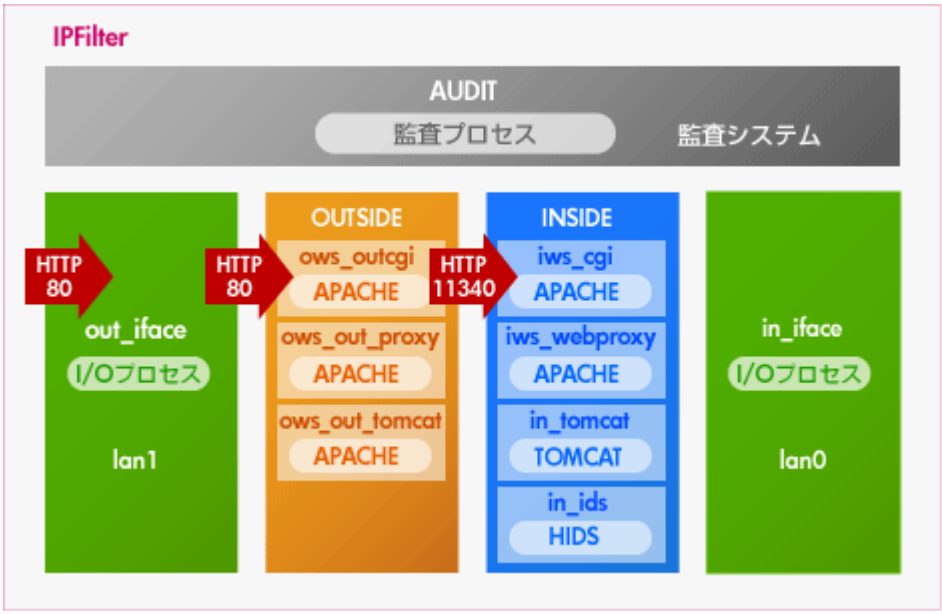

図 2：PS-Webserver のアーキテクチャ

Trusted Computing Services - セキュリティチップを使用した暗号化強化

Trusted Computing Services（TCS）は、Integrity サーバーのハードウェアに備わるセキュリティ・チップ TPM（Trusted Platform Module）を利用した暗号化機能である。TPM にアクセスするカーネル・ドライバを提供するほか、HP-UX のデータ暗号化機能 EVFS との連携に対応する。例えば EVFS では暗号化に用いる秘密鍵をディスク上に保存するが、TCS の利用によりこの秘密鍵を TPM で保護することができる。これにより、もしサーバー本体やディスクがまるごと盗難された場合でも、秘密鍵を取り出して用いることは不可能となる。また、ファイル暗号化ユーティリティも提供しており、TPM のない信頼されないサーバーではファイルを復号できないようにすることもできるため、より強力な情報漏えい対策が可能になる。

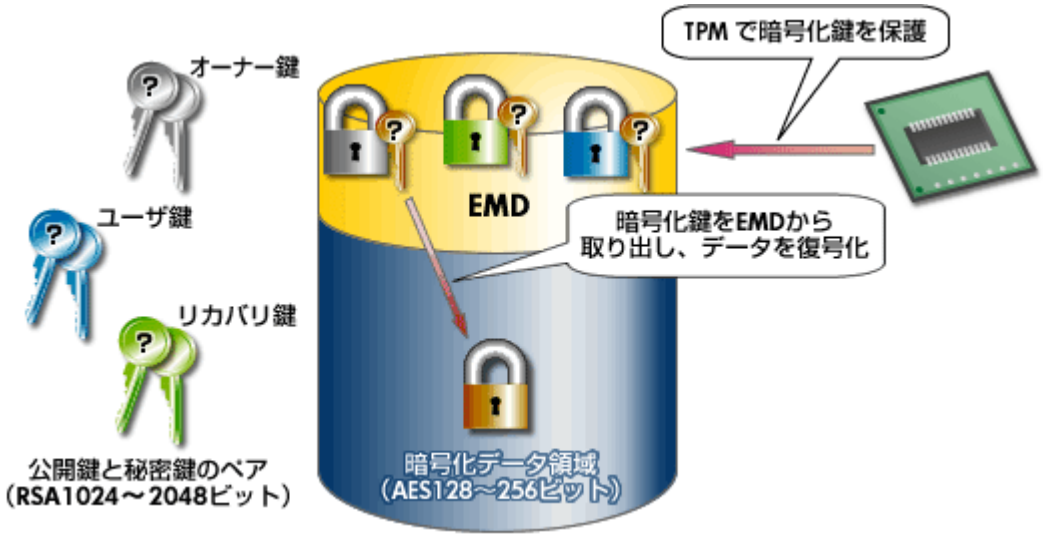

図 3：EVFS での鍵管理のメカニズム

システム保護

Host Intrusion Detection System - 侵入検知システム

Host Intrusion Detection System（HIDS）は、HP-UX に備わるホストベースの侵入検知システムである。HP-UX が管理する多種多様なイベントを管理者に代わって監視し、クラッカーによる HP-UX への侵入の試みを検知した場合には管理者へリアルタイムに通知する。あらかじめ必要な設定が定義済みのテンプレートを備えるほか、管理用 GUI を提供する。

Standard Mode Security Extensions - 監査ログ＆システムセキュリティポリシー設定

Standard Mode Security Extensions（SMSE）は、HP-UX の高信頼性モード（trusted mode）でサポートされていた高度な監査機能やセキュリティ・データベース機能を、標準モードでも利用可能とする機能である。例えば、システムコール・レベルでの監査をはじめ、改ざんの困難なバイナリ・ファイルへのログ保存、認証失敗の多発時のアカウント・ロック、パスワード履歴管理による既知のパスワードの利用禁止、未使用アカウントのロックなどの機能を提供する。Web ベースの管理ツールである SMH（System Management Homepage）から設定や管理が可能だ。

Bastille - システム要塞化ツール

Bastille は、HP-UX のセキュリティを強化するロック・ダウン（要塞化）を支援する対話型のツールである。Bastille を起動すると、内蔵するチェックリストに基づいて、多数の質問が管理者に向けて表示される。これらにひとつずつ答えていくことで、セキュリティの甘いデフォルト設定の HP-UX から、強固なセキュリティ設定が施され要塞化された HP-UX へと変えることができる。例えば、不要なサービスやデーモン、システム設定の無効化をはじめ、セキュリティ・パッチ・チェックの自動起動設定、IPFilter ベースのファイアウォール設定を実施する。また、セキュリティ設定の結果を HTML やテキスト形式のレポートに出力できる。さらに、これら Bastille の各機能は、統合管理ツールである SIM（Systems Insight Manager）に統合されている。

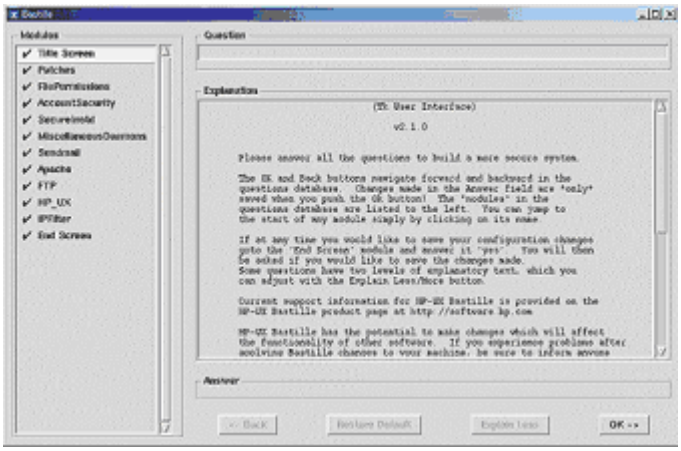

図 4：Bastille の設定画面

アイデンティティ保護

Role-based Access Control - 職務分掌に従ったアクセス管理

RBAC（Role-Based Access Control）は、いわゆる「root アカウントの乱用」を防ぐ機能である。その名が示すとおり、一般アカウントに細かなロール（役割）を割り当て、そのロール単位で root 権限の一部を委譲する。これにより、root アカウント権限を持たなくても必要な作業を進められる仕組みだ。これまで root アカウントでしか実行できなかった管理機能を一般アカウン

トでも必要に応じて実行可能とする、きめ細かな権限委譲のメカニズムを備える。この機能により実際の職務に応じた権限を適切に設定できる。この RBAC は SMH に統合されており、ロールの登録や修正は SMH の Web 画面上で簡単に設定できる。

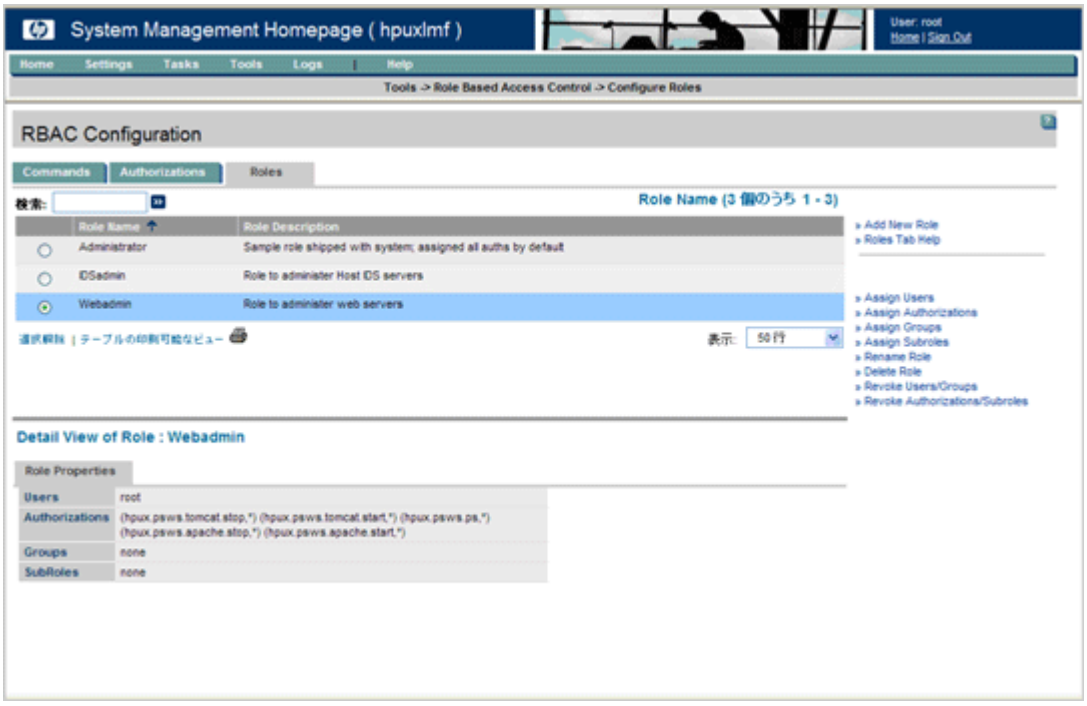

図 5：SMH での RBAC 設定例

以上、今回は HP-UX に備わるセキュリティ機能のうち、今後取り上げる予定のものを簡単に紹介した。これらのセキュリティ機能をひととおり理解し、「HP-UX セキュリティのゼネラリスト」となることが、多層防御を実現する近道となるはずである。

第 2 回

EVFS と TCS によるデータ暗号化

2008 年 2 月　テクニカルライター 吉川和巳

ディスク上のファイルやボリュームの内容をまるごと暗号化したい。そうしたニーズに応えるべく、HP-UX では暗号化ファイルシステム「EVFS」を提供している。EVFS では、既存のアプリケーションを変更することなく、そのまま暗号化ファイルシステムに移行することができる。また HP-UX の暗号化機能「TCS」では、Integrity サーバーに搭載されたセキュリティチップ TPM の鍵を用いた暗号化が可能だ。例えばディスク・ドライブをほかのサーバーに接続したとしても、鍵がないので暗号化ボリュームには一切アクセスできなくなる。また、パスフレーズを入力せずに暗号化ボリュームを自動マウントするといった使い方も可能になる。

EVFS とは

ディスク上のファイルやボリュームの内容をまるごと暗号化したい。そうしたニーズに応えるべく、HP-UX では 2006 年より暗号化ファイルシステム「EVFS（Encrypted Volume & Filesystem）」を提供している。EVFS は、HP-UX において暗号化ボリューム機能および暗号化ファイルシステムを提供するもので、以下のような特徴を備えている。

ディスク上のデータを暗号化	EVFS では、ディスク上に記録するデータそのものを暗号化する。そのため、もしディスク・ドライブやバックアップ・メディアの紛失や盗難があったとしても、データの保護が可能だ。
アプリケーション透過性	EVFS は、アプリケーションからは HP-UX の普通のファイルシステム（VxFS/HFS）やボリューム（LVM）として認識される。よって、アプリケーション側で特別な API を実装したり改変を行ったりする必要はない。
「ボリューム単位の保護」と「ファイル単位の保護」に対応	EVFS では、ファイルごとの暗号化とボリューム全体の暗号化という 2 通りのデータ保護機能があり、用途に応じて選択できる予定だ。（現在のバージョンではボリューム単位の保護のみをサポート。ファイル単位の暗号化は将来のバージョンでサポート予定。）
故意や過失によるデータへの 不用意なアクセスを防止	暗号化されたデータにアクセスするには「鍵」と「パスフレーズ(パスワード)」が必要であるため、システム管理者が不用意にデータにアクセスすることを防止できる。

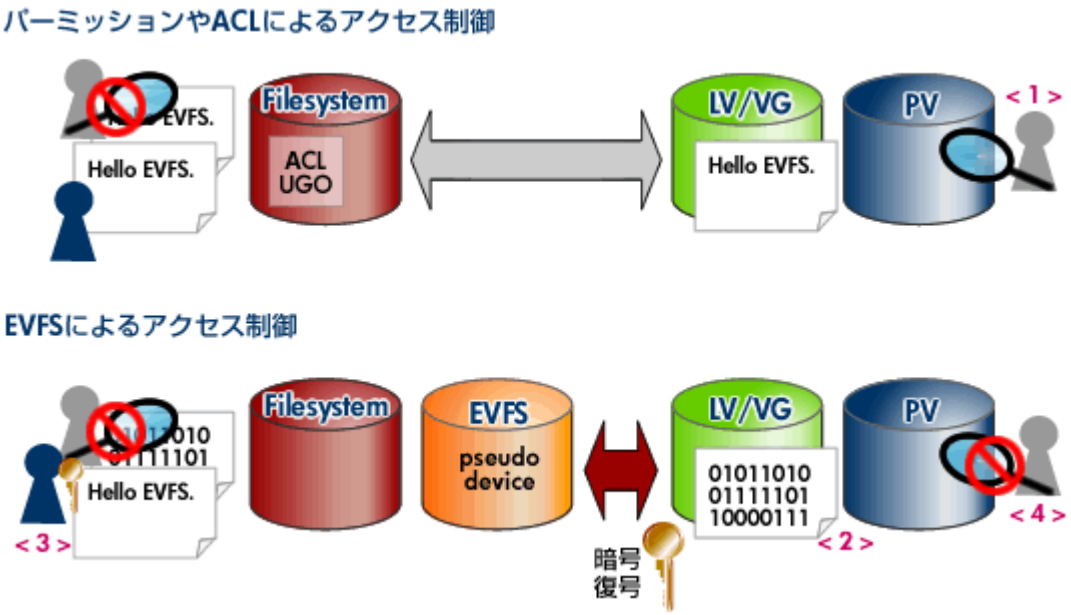

図 1：一般的なファイルシステムと EVFS との違い

図 1 は、ファイル単位の保護について、一般的なファイルシステムと EVFS の違いを示したものである。例えば HP-UX における VxFS や HFS などの一般的な UNIX ファイルシステムでは、ファイルやユーザの単位でパーミッションや ACL（アクセス制御リスト）を管理することで、ファイルへのアクセスを制限する。しかし、ディスク上のデータそのものが暗号化されているわけではない。そのため、特権を持つユーザやアプリケーションからアクセスする場合、ディスクそのものを取り外した場合、バックアップ・メディアから別ディスク上にリストアした場合などは、自由にデータにアクセスできる（＜1＞）。

一方、EVFS では、ディスク上に記録された個々のファイルの中身そのものが暗号化されている（<2>）。よって、適切な鍵を持つユーザやアプリケーションであれば、データを復号化することができる（<3>）。これに対し、たとえ特権を持つユーザでも、鍵を持たなければ物理的にどのような策を講じてもファイルの中身を見ることができない（<4>）。

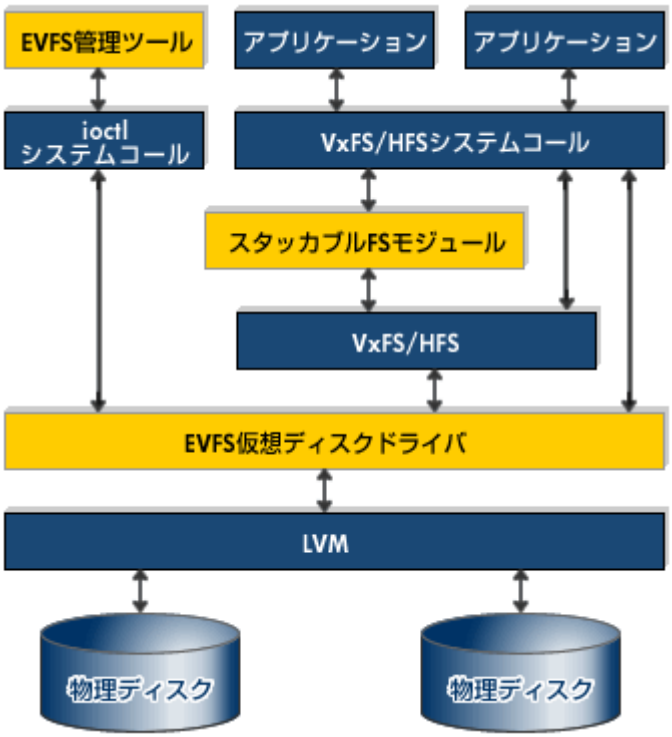

図 2：EVFS 導入時のディスク I/O の流れ

図 2 は、EVFS 導入時のディスク I/O の流れを示したものである。この図において、青色で示した部分は、HP-UX に従来から備わるファイルシステム（VxFS／HFS）やボリューム・マネージャ（LVM）を表している。一方、黄色の部分が、EVFS 導入時に追加されるコンポーネントである。この図が表すとおり、アプリケーション側ではこれまで通り VxFS または HFS のシステムコールを利用するだけであり、EVFS の存在を意識する必要はない。ファイル書き込み時の暗号化や、読み出し時の復号化は透過的に実行されるため、既存のアプリケーションをそのまま暗号化ファイルシステムに移行することが可能だ。

EVFS の暗号化機能を提供するのは、上手の中央に記された「EVFS 仮想ディスクドライバ（EVFS pseudo-disk driver）」である。同ドライバは、VxFS／HFS と LVM の中間に位置しており、LVM へのアクセス時にデータの暗号化を実行する。その名が示すとおり、仮想的なディスクドライバとしてふるまうため、VxFS や HFS といったファイルシステムには依存しない仕組みになっている。

また、上図の左上に黄色で示された「EVFS 管理ツール」は、同ドライバの動作を制御するコマンドツール群である。EVFS における暗号化機能や鍵の管理は、すべてこれらのツールを通じて実施することになる。

「鍵」の管理メカニズム

EVFS では、いくつかの「鍵」を用いることで、ファイルシステムやボリュームの暗号化を実現している。その具体的なメカニズムを紹介したい。

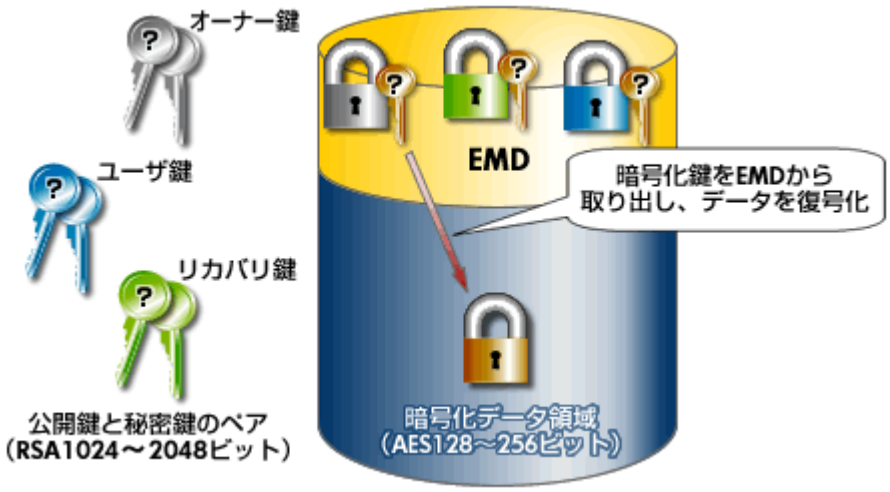

図 3：EVFS の鍵管理のメカニズム

図 3 に示すとおり、EVFS によって暗号化されたボリュームは「暗号化データ領域」と「EMD（Encryption Meta Data）」と呼ばれる 2 つの領域に分割される。前者は暗号化されたデータを保管する場所であり、後者は暗号化に用いる鍵を保管する場所として利用される。

EVFS では、データを暗号化する手段として、共通鍵暗号方式としてもっとも新しい業界標準である AES を採用している。AES は共通鍵暗号であるため、暗号化と復号化のいずれにも同じ暗号化鍵（128 ビット～256 ビット）を利用する。この暗号化鍵は、いわばデータにアクセスするためのパスワードのようなものだ（上図の金色の鍵）。

この暗号化鍵は、公開鍵暗号の標準である RSA によって暗号化されたのち EMD に記録される。この「暗号化鍵の暗号化」では、以下の 3 種類の鍵（1024～2048 ビット）を使用する。

オーナー鍵(銀色の鍵)	ボリュームの所有者が持つ鍵である。暗号化ボリュームの所有者が持つ鍵であり、1 つだけ作成される。オーナー鍵を使うことで、ボリュームへのアクセスだけでなく、ボリュームの削除などすべての管理権限を有する。また、次に説明するユーザ鍵の作成も可能になる。
ユーザ鍵(青色の鍵)	暗号化ボリュームにアクセスしたいユーザが持つ鍵である。ボリュームのマウントやアンマウント、ボリュームへのアクセスは可能だが、ボリュームの削除といった権限は持たない。ユーザ鍵は複数個作成できる。
リカバリ鍵(緑色の鍵)	オーナー鍵の破損・紛失時のために、オーナー変更の権限を持つ。リカバリ鍵は複数個作成できる。

EVFS では、これら 3 種類の鍵を用途に応じて使い分ける。例えばシステム管理者は暗号化ボリュームを作成してオーナー鍵を管理する一方で、同ボリュームを利用するユーザやアプリケーションに対してユーザ鍵を発行する、といった具合だ。またオーナー鍵ファイルが破損したり紛失したりしたときに備えて、暗号化ボリュームの所有者を切り替えできるリカバリ鍵をバックアップ保存しておく。

セキュリティチップを使用した暗号化機能を提供する TCS

このように EVFS では、適切な鍵を持たないユーザやアプリケーションによる暗号化ボリュームへのアクセスを防ぐことができる。とはいえ、この EVFS を利用する上でネックとなるのは、「鍵の保管」の問題だ。前述のとおり、EVFS で用いる各種の鍵は、ディスク上に設けられた EMD と呼ばれる領域に保管される。よって、万が一ディスク・ドライブがサーバー本体より取り外されてしまうと、その鍵も合わせて持ち運ばれることになる。

もちろん、鍵はパスフレーズによって保護されるため、すぐさま不正アクセスが可能になるわけではない。とはいえ、例えば総当たり攻撃によって鍵が破られるリスクも完全には排除できないうえ、ディスク盗難のリスクを考えると「パスフレーズの入力なしで暗号化ボリュームを自動的にマウントしたい」といった使い方は難しい。

そこで、こうした鍵の保管の問題を解決すべく提供されたのが、HP-UX の TCS（Trusted Computing Services）である。TCS は、Integrity サーバー（rx2660、rx3600、rx6600、BL860c）に搭載されたセキュリティチップ TPM（Trusted Platform Module）を使用して暗号鍵を保護する機能を提供する。つまり、ディスク上の暗号鍵をサーバー本体の TPM の鍵で暗号化し、暗号化鍵を保護する仕組みである。

セキュリティチップ TPM は、標準化団体 Trusted Computing Group が制定した規格に基づいて開発されたハードウェアチップである。暗号鍵やパスワード、デジタル署名などを保存する不揮発性メモリを備えるほか、鍵の生成や乱数生成などの機能も備える。単なるメモリ・チップとは異なり、TPM を取り外して不正に鍵を取り出すことはできない構造となっている。

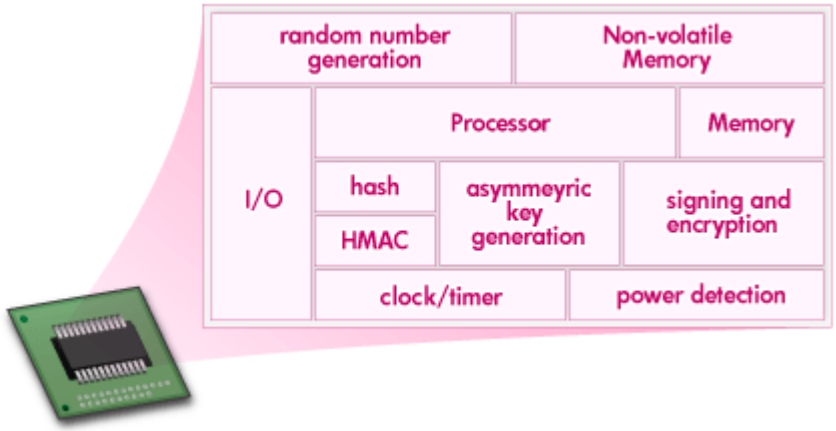

図 4：セキュリティチップ TPM の構造

HP-UX の TCS では、専用のライブラリとデバイスドライバを通じて、この TPM 上に EVFS の鍵を保護する機能を提供する。また EVFS 以外にも、ファイル単体の暗号化や複合化のためのコマンドを提供するほか、今後はサードパーティ・アプリケーションの鍵も取り扱い可能になる予定だ。

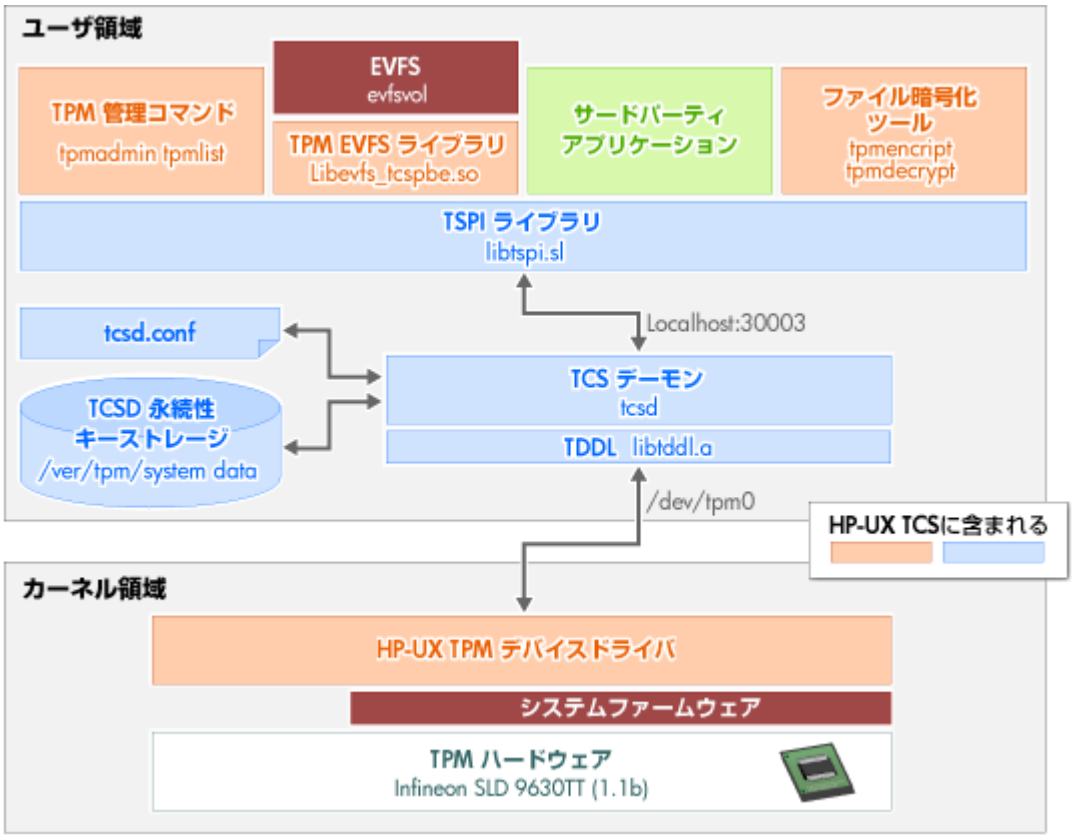

図 5：HP-UX の TCS による TPM へのアクセス

TCS 利用の実際

この TCS を用いて EVFS の暗号鍵を TPM で保護することで、上述したような鍵の保管の問題が解消される。例えばディスク・ドライブをほかのサーバーに接続したとしても、TPM で暗号化された鍵が復号できないので暗号化ボリュームには一切アクセスできなくなる。また、パスフレーズを入力せずに暗号化ボリュームを自動マウントするといった使い方も可能になる。

では、TCS の利用手順をごく簡単に紹介しよう。まずは、rx2660 や rx6600 などの TPM をサポートする Integrity サーバーに対し、TPM を装着する。この TPM の機能を有効化するには、EFI メニューより「Security Configuration -> Set Trusted Platform Module State」を選択すればよい。TPM が動作中かどうかは、EFI シェルより以下の secconfig コマンドを実行することで確認できる。

```
Shell> secconfig
SYSTEM SECURITY CONFIGURATION
 Trusted Boot:        Not Supported
 TPM:                 Enabled
 TPM Vendor ID:       0x15D1
 TPM Product ID:      0x0006
 TPM TCG Spec Version:  1.1.0.0
```

または、HP-UX の ioscan コマンドでも確認が可能だ。

```
# ioscan -fC tpm
```

```
Class    I  H/W Path  Driver S/W State H/W Type  Description
=============================================================
tpm      0  250/2    tpm  CLAIMED   INTERFACE Trusted Platform Module
```

つづいては、以下のコマンドにより TCS のソフトウェアをインストールし、インストール状態を確認する。インストールにともなうリブートは不要だ。

```
# swinstall -x autoreboot=true -s /var/tmp/TCS_A.01.01_HP-UX_B.11.31_IA.depot TCS
# swlist -l product |grep -i tcs
 TCS-PROD            A.01.01       HP-UX Trusted Computing Services
# /opt/tcs/bin/tpmadm selftest
 TPM Test Result: success
# ps -ef|grep tcs
    tss  4344     1  0 16:43:25 ?       0:00 /opt/tcs/bin/tcsd
```

以上の手順で、TCS のインストールは完了である。これにより、TPM 管理のための各種コマンドが利用可能となる。

コマンド	機能
tpmadm	TPM の無効化、有効化 オーナーシップ設定 TPM パスワード変更 鍵のバックアップ、リストア
tpmlist	TPM ステータス表示 鍵情報表示
tpmencrypt	ファイルの暗号化
tpmdecrypt	ファイルの復号化
evfs_setup	EVFS で TPM を使用するセットアップ

例えば tpmencrypt コマンドを用いることで、以下のようなファイルの暗号化が可能だ。

```
# more test.txt
This is a test file.
# tpmencrypt -o test.enc test.txt
Password :
# more test.enc
<tpmcrypt version="1.0">
<tpm_key reptype="blob" size="559">
```

AQEBBwAUAAAABgEAAAABAAMAAQAAAAwAAAgAAAAAgAAAAAAAAAAAAAABAIVLmtq+
jMYqJDVsxUImBvykI+wO5BOaZuQPFXdRlHQ4Robanbu1jjnoj6PsUrTZ94Oi5og8
vzVgtWMGQOhf/MpPO/Pejm0NZ10VNz5OeTMn1632nwSln5VvUzwRjO2YrlUCV8+l
＜以下略＞

この場合、TPM 内の鍵を暗号化に使用するため、このサーバー以外では復号することが不可能となる。

以上、今回は EVFS によるデータ暗号化機能と、TCS を使用したより強固な暗号化について説明した。Integrity サーバーと HP-UX に備わるこれらの暗号化機能をフルに活用し、IT システムのセキュリティ強化に役立てていただきたい。

第 3 回

PS-WS でつくるセキュアな Web サーバー

2008 年 3 月　テクニカルライター 吉川和巳

ひとことに「セキュアな Web サーバー」と言っても、管理者の知識や経験に基づいて、多岐にわたるセキュリティ対策を実施する必要がある。よって、Web サーバーのセキュリティは個々の管理者のスキルに大きく依存しがちだ。HP の Protected Systems Web Server（PS-WS）は、こうしたセキュリティ対策の属人性を排除し、高度なセキュリティ対策を誰もが実施可能となる「セキュアな Web サーバーのリファレンス・アーキテクチャ」だ。金融機関などで豊富な実績のある厳格なセキュリティ強化のベストプラクティスを、手軽に利用可能なテンプレートとして提供する。

“セキュアな Web サーバー”のエッセンスを凝縮した「PS-WS」

ひとくちに「セキュアな Web サーバー」と言っても、それを実現するにはいくつもの側面で多岐にわたるセキュリティ対策を講じる必要がある。それらは、大きく「OS レベルのセキュリティ強化」と「Web サーバー・レベルのセキュリティ強化」の 2 つに分類できる。

OS レベルのセキュリティ強化

- Web サーバー上の不要なアカウントやサービスを停止する（ロックダウン）
- root 権限の利用を最小限にとどめ、権限の細分化や監査を実施する
- Web サーバー、アプリケーション・サーバー、CGI などのそれぞれを独立したサーバー上で運用し、隔離性を高める
- IPFilter などを用いて、Web サーバー単体のファイアウォール設定を実施する
- IDS（侵入検知システム）を導入する

Web サーバー・レベルのセキュリティ強化

- Apache サーバーに適切なパッチを適用し、Apache そのものの脆弱性を取り除く
- Apache サーバーの設定内容を見直し、システムの脆弱性となりうる部分を排除する
- Web アプリケーションの実装時に、認証やアクセス制御、サニタイジングなどのセキュアなコーディングを徹底する

これらは、考え得る数多くのセキュリティ対策の一部分にすぎない。通常は、こうしたさまざまなセキュリティ対策のひとつひとつを、管理者や開発者が自らの知識や経験に基づいて実施していく。よって、Web サーバーのセキュリティは、個々の管理者のセキュリティ・スキルに大きく依存することになる。とりわけセキュリティ機能が豊富な HP-UX では、上述した各対策を実施するためには、それらのセキュリティ機能の役割や使い方をひととおり熟知しておくことが要求される。こうした「属人性」の高さが、セキュリティ対策の難しい点だ。

とはいえ、上述したセキュリティ対策のうち、前者の「OS レベルのセキュリティ強化」については、あらゆるプロジェクトに共通する普遍的な対策ポイントとなる。よって、「OS レベルのセキュリティ対策」をテンプレート化できれば、管理者のスキルに依存せずに、少なくとも OS レベルでは一定したセキュリティ・レベルを確保可能になる。

HP の Protected Systems Web Server（PS-WS）は、そうした観点で作成された「セキュアな Web サーバーのリファレンス・アーキテクチャ」だ。金融機関などで豊富な実績のある厳格なセキュリティ強化のベストプラクティスを、手軽に利用可能なテンプレートとして提供する。具体的には、HP-UX の多彩なセキュリティ機能を使いこなし、OS レベルで高度なセキュリティ対策を施すためのスクリプト群や設定ツール、ドキュメントから構成される。HP-UX 11i v2 0609 以降で利用が可能で、HP-UX 11i v3 でもサポートされる。

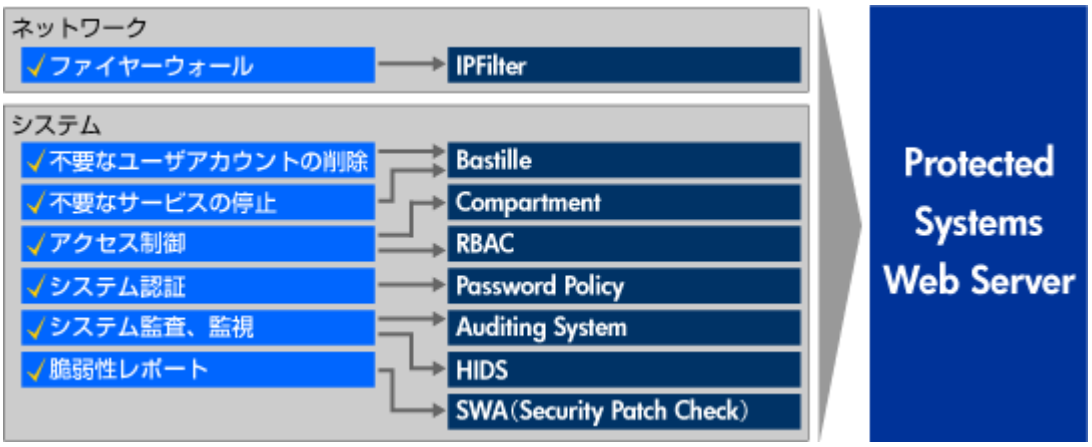

図 1：PS-WS を構成する各種のセキュリティ機能やツール

PS-WS の適用範囲

PS-WS を用いることで、HP-UX に備わる以下の各種セキュリティ機能が自動・半自動で設定され、利用可能になる。

- IPFilter――ファイアウォール機能
- Bastille――不要なサービスやアカウントのロックダウン、OS 各所のパーミッション設定強化など
- Security Containment――コンパートメント化による Web サーバー、アプリケーション・サーバー等の隔離性確保と集約
- RBAC――root 権限を細分化し、一般ユーザへ委譲。root 権限の乱用によるセキュリティ低下を防ぐ
- システム監査――システムコール・レベルで監査を実施し、独立したコンパートメント内で監査ログを記録
- HIDS――侵入検知システム

もっとも、PS-WS をインストールしさえすれば、これらすべてが自動的に設定されるわけではない。例えば HIDS やシステム監査、Apache などの設定は、管理者が個別に実施する必要がある。またもちろん、Web アプリケーション・レベルのセキュリティ脆弱性は、PS-WS だけでは防ぐことができない。PS-WS は、あくまでも支援ツールなのである。

では続いて、これらのセキュリティ機能についてもう少し詳しく見ていきたい。

Security Containment による隔離性の確保

PS-WS のもっとも大きな特長は、HP-UX の"セキュア OS 版"である Virtual Vault で長年培われてきた「Security Containment」によるコンパートメント化を採用している点だ。これはいわば、「OS の内部を頑丈な区画（コンパートメント）で仕切る」ための機能である。

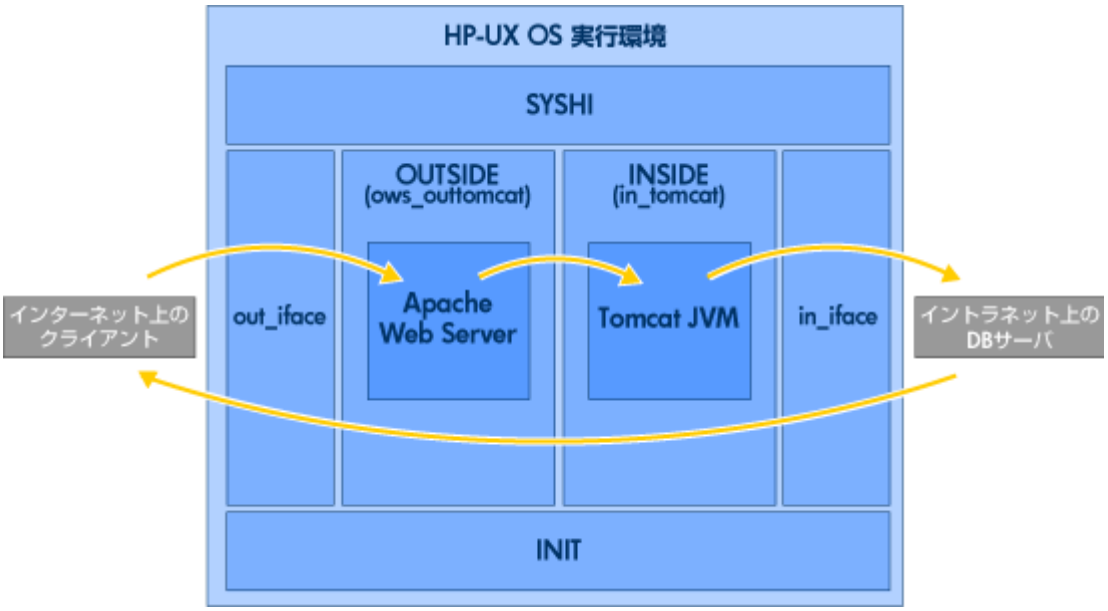

図 2：Security Containment によるコンパートメント化

ここでは、Apache サーバーと Tomcat サーバーを組み合わせた典型的な Web アプリケーション・サーバーの構成を例にとって説明しよう。PS-WS をインストールすると、上図のコンパートメントが自動的に作成される。これらのコンパートメントは、それぞれ以下のような役割を担う。

コンパートメント	稼働するプロセス
out_iface	インターネット側からの HTTP アクセスを送受信する I/O プロセス
ows_out_tomcat	Apache サーバー・プロセス
in_tomcat	Tomcat プロセス
in_iface	イントラネット側（DB やアプリなど）へのアクセスを送受信する I/O プロセス
INIT	OS 起動時の各種基本プロセス
SYSHI	管理用プロセス

表 1：PS-WS が作成するコンパートメント

例えば、一般的な Web サーバーでは、こうしたコンパートメントという概念は一切利用されていない。そのため、もし Apache サーバーの脆弱性が攻撃され不正アクセスが行われると、そこを起点に Tomcat サーバーも攻撃され、Tomcat がアクセスする DB サーバーへの攻撃も可能となる。

これに対し、コンパートメントが導入された Web サーバーでは、上記の表のとおり、I/O プロセス、Apache、Tomcat のそれぞれが個別のコンパートメントに分割される。そして個々のコンパートメントは、物理的に独立したサーバーと同等の隔離性を備えている。つまり、たとえ Apache 用コンパートメントで Web サーバー実行権限を獲得しても、そのほかのコンパートメント内部のプロセスやファイルを操作したりアクセスしたりすることはできないため、さらなる不正アクセスの伝播を防ぐことが可能だ。よって、あたかも「Apache と Tomcat を個別のサーバー・マシンで運用する」ような、徹底したセキュリティ強化が可能になる仕組みだ。

これを逆にとらえると、これまでセキュリティ上個別に運用されてきた Apache や Tomcat、もしくはそのほかの CGI プロセスも、すべて 1 台のサーバーに安全に集約可能になる。セキュリティを強化しつつ TCO も削減できる点が、Security Containment のユニークな点だ。

また、Apache や Tomcat 以外にも、HIDS による侵入検知システムやシステム監査プロセスなども、独立したコンパートメントで運用される。よって、万が一クラッカーに侵入された場合でも、侵入検知の妨害やログの改ざんといったクラッカーによる“証拠隠滅”を防ぐことができる。

RBAC による root 権限の細分化

また PS-WS では、HP-UX の Role based Access Control（RBAC）による root 権限の細分化が実施される。これはつまり、これまで root アカウントに集中していた数多くのシステム管理権限を用途ごとに分割し、一般ユーザ・アカウントに委譲する仕組みだ。

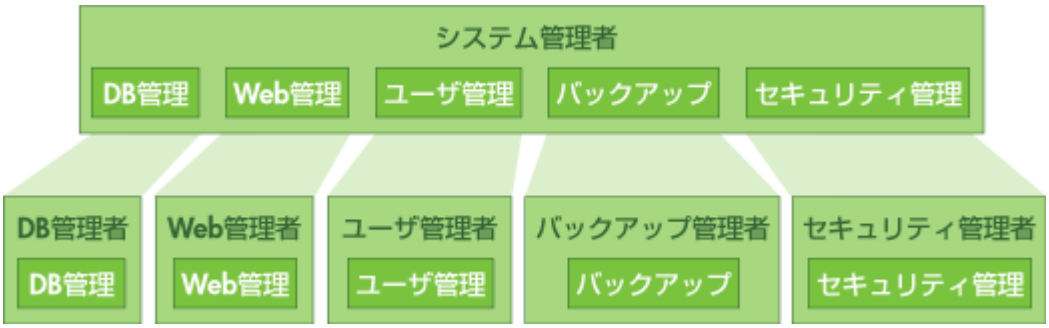

図 3：RBAC による root 権限の細分化

これにより、Apache の管理、DB の管理、バックアップ作業、ユーザ・アカウント管理といったさまざまな管理作業のすべてに root アカウントを用いる必要がなくなり、本来権限を持たない管理者が必要以上の管理作業ができてしまうという状況をなくすことができる。つまり不正アクセスの機会を与えず、統制の効いたシステム運用を実施できる。 以上、今回は PS-WS によるセキュアな Web サーバー構築の手法を紹介した。ここで説明したのは、PS-WS によって利用可能になる多彩なセキュリティ機能のごく一部ではあるが、管理者のスキルに依存せずに高度なセキュリティ強化が実現可能な点はご理解いただけたはずだ。

第 4 回

Bastille によるシステムアセスメント

2008 年 4 月　テクニカルライター 吉川和巳

Bastille は、HP-UX システムのセキュリティ設定を強化するツールである。GUI や TUI 上でのインタラクティブな操作を通じて、システムのロックダウン（不要なサービスの停止やシステム設定の変更）を簡単に実施できるのが、最大の特徴だ。ただ従来の Bastille には「実施しているロックダウンの詳細内容がレポートとして残らない」という不便な点があった。そこで Bastille の最新バージョンである 3.0 では、新たにシステムアセスメント結果をレポート出力する機能が追加されている。

Bastille が備える特徴とは

Bastille は、UNIX システムに対してロックダウン（不要なサービスの停止やシステム設定の変更）を実施し、セキュリティを強化するツールである。もともとはオープンソース・ソフトウェアとして開発されたもので、HP ではこの Bastille を HP-UX 向けに移植し、機能拡張を施したものを Web サイト上で無償提供している。Bastille は、おもに以下のような特徴を備える。

システムのロックダウン

- 各種デーモン設定やシステム設定のセキュリティ強化
- 不要なサービスの停止
- Jail による Web サーバーや DNS などのインターネット・サービスのセキュリティ強化
- Software Assistant や Security Patch Check の自動実行設定
- IPFilter ベースのファイアウォール設定

アセスメントレポート作成

- HTML やテキスト形式、設定ファイル形式でアセスメントレポートを生成
- システムのベースライン設定を記録し、その後の変化をチェックする

SIM との連携

- SIM メニュー上からのロックダウン設定やレポート表示が可能
- SIM サーバーのロックダウンをサポート

その他

- セキュリティ設定方法を解説したヘルプテキストを豊富に提供
- Bastille 実行前の設定を簡単に復元可能

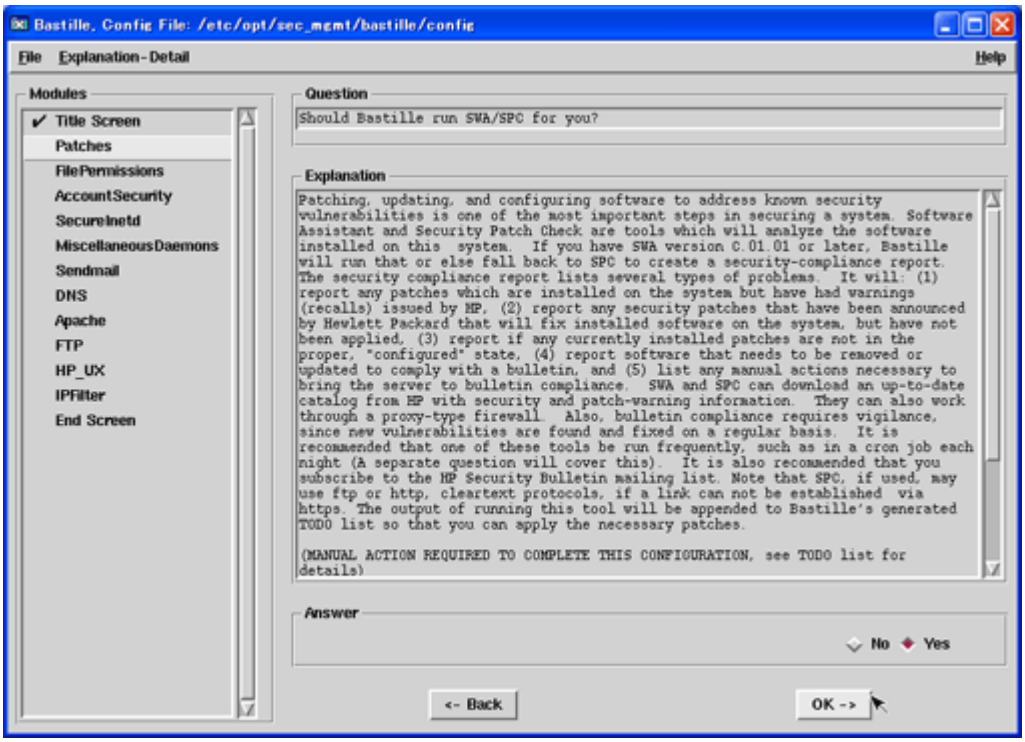

図 1：Bastille の GUI 画面

Bastille の最大の特徴は、図 1 のような GUI や TUI 上でのインタラクティブな操作を通じて、システムのロックダウンを簡単に実施できる点だ。この画面の左側には「モジュール」と呼ばれる項目が並んでおり、それぞれの項目についてウィザード形式の質問事項が表示される。個々の質問事項では、その設定を実施することで得られるセキュリティ強化のメリットと、それにともなうデメリットが詳細に解説されている。管理者は、それぞれの項目について Yes か No かを選択したり設定値を入力したりすることで、用途に応じて適切なロックダウンを実施できる。すべての質問に答えると、Bastille が自動的にロックダウンを開始する。もっとも、すべての設定が自動化されているわけではなく、管理者による手動設定が必要な部分については「To Do」リストを作成してくれる。

では、Bastille によるロックダウンの内容をもう少し詳しく見ていこう。

ファイル・パーミッションやパッチ管理の強化

Bastille のモジュール「File Permission」では、HP-UX のファイル・システムに対して world-writable なディレクトリ（すべてのユーザから読み書き可能なディレクトリ）のスキャンを実施するかどうかを指定できる。このスキャンを実施すると、world-writable なディレクトリのパーミッション設定を個別に編集するためのスクリプトを自動作成できる。
また、モジュール「Account Security」では、ユーザ・アカウントのアクセス権管理の詳細なチューニングが可能だ。例えば、すべてのユーザとシェルにおけるデフォルトの umask 値（作成されるファイルのパーミッション値）の設定をはじめ、ログイン・ポリシー（ログイン拒否やログイン許可数、ルートログイン禁止）およびパスワード・ポリシー（パスワード変更間隔や期限切れ警告、シャドウ・パスワードの利用）の設定を容易に行える。さらには、HP-UX の高信頼性モードを利用したセキュリティ監査（すべてのシステムコールのトレース）も指定することができる。

一方、モジュール「Patches」では、HP が提供するパッチ管理ツール Software Assistant（SWA）や Security Patch Check（SPC）のセットアップが可能だ。簡単な質問に答えるだけで、SWA の起動や crontab への登録、起動時間の設定などを行え

る。これにより、毎日一定の時間に SWA が起動し、ホストに適用されているセキュリティパッチの一覧と HP が公開するパッチ一覧の比較が行われる。

ネットワーク・サービスのロックダウン

一方、モジュール「Secure Inetd」および「Miscellaneous Daemons」では、HP-UX に備わる多数のネットワーク・サービスのロックダウン設定が可能である。実際の設定では、inetd デーモンにより起動される大半のサービス（telnet や ftp/tftp、login/shell/exec、finger/ident、bootp、uucp、ntalk など）や、NFS、NIS、snmpd などはすべてデフォルトで無効とすることが可能だ。よって、本当に利用したいサービスだけを個別に起動を指定すればよい。
また、Web やメール、DNS などの標準的なインターネット・サービスのセキュリティ保護のためのモジュールとして、Apache および sendmail、DNS、FTP が用意されている。これらのモジュールでは、Apache httpd や sendmail のロックダウンを行えるほか、chroot を利用して BIND や Apache のプロセスがアクセス可能なディレクトリを制限することも可能だ。

では、こうしたネットワーク・サービスのロックダウンの効果を実際に確認してみよう。まずは、ロックダウン実施前の HP-UX に対して、ネットワーク・スキャン・ツール「Nessus®」を実行し、ポートのオープン状態を確認する。図 2 は、その結果である。

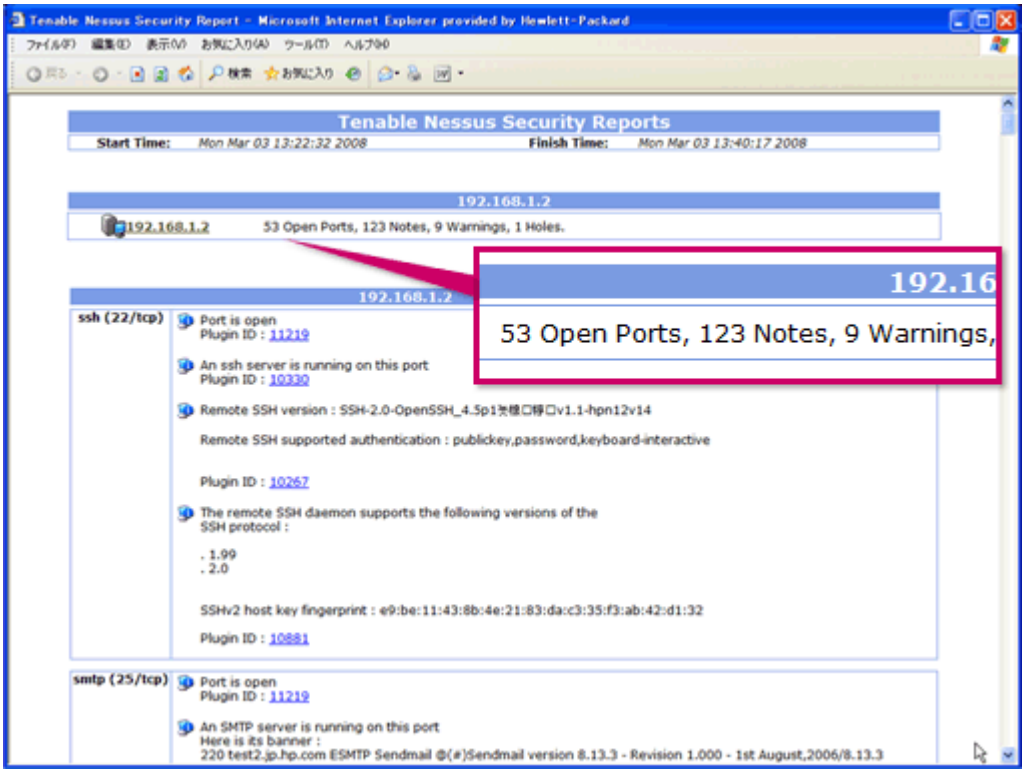

図 2：ロックダウン前のポートのオープン状態
※このレポートは Tenable Network Security 社の Nessus®を使用して生成されたものです。

このように、HP-UX のデフォルト設定では ssh や SMTP などのポートがオープン状態であり、外部から接続可能であることがわかる。つづいて、Bastille による「DMZ ロックダウン」（DMZ ホスト向けのロックダウン）を実施し、その後に Nessus を再度実行すると、図 3 のようなレポートが得られる。

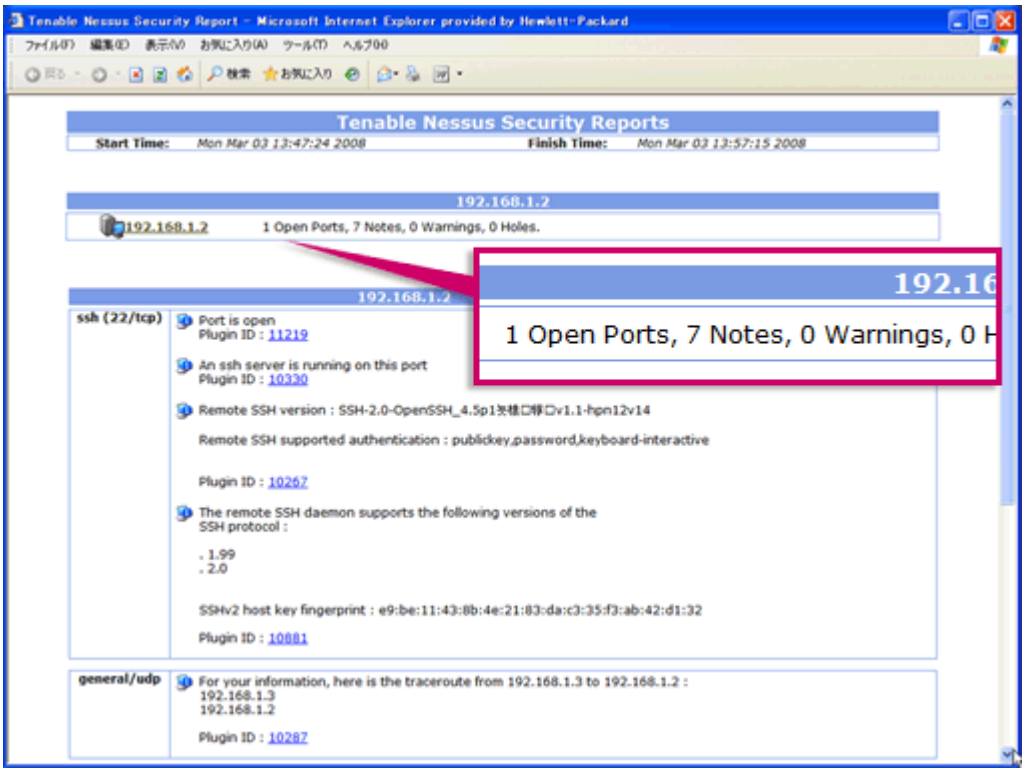

図 3：ロックダウン後のポートのオープン状態
※このレポートは Tenable Network Security 社の Nessus®を使用して生成されたものです。

このように、ssh は接続可能であるものの、SMTP はストップしており、ssh のような必要最小限のサービス以外はロックダウンされていることがわかる。

新たに追加されたシステムアセスメント機能

ここまで見てきたとおり、Bastille を用いることで HP-UX のロックダウンが容易に実施でき、経験の浅い管理者でも高度なセキュリティ強化を施したシステムを構築できることがわかる。ただ、これまでの Bastille には「実施しているロックダウンの詳細内容がレポートとして残らない」という不便な点があった。つまり、どのサービスを停止し、どの設定をどのように変更したか、ひと目で理解できるようなドキュメントが得られないのである。とりわけ近年の IT 統制の観点からすると、セキュリティ強化ポイントの文書化はぜひ提供してほしい機能だ。

そこで Bastille の最新バージョンである 3.0 では、新たにシステムアセスメント機能が追加された。同機能では、Bastille によって実施したロックダウンの内容を HTML 形式やテキスト形式のアセスメントレポートとして文書化できる。

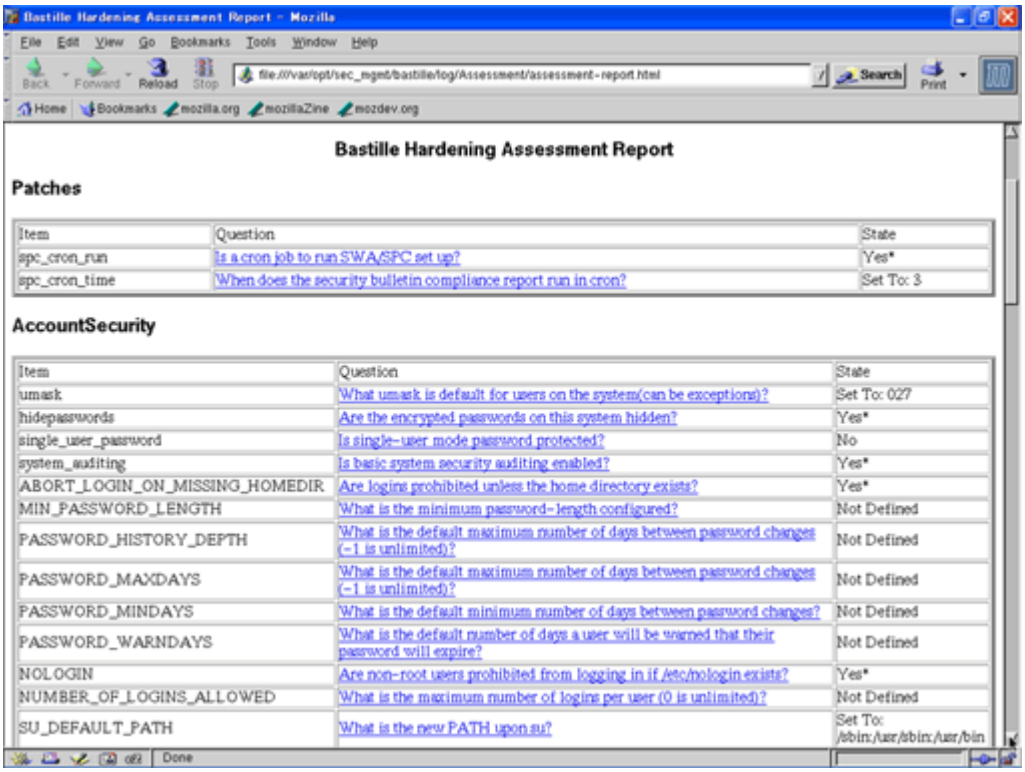

図 4：アセスメントレポートの例

図 4 にあるように、例えば umask 設定や hidepasswords といったセキュリティ設定項目について、Bastille 実行後のそれぞれの設定内容を簡単に確認できる。この HTML ページをそのままシステムアセスメントレポートとして IT 統制に活用できる仕組みだ。
また新バージョンでは、Bastille 実行後のシステム設定状態を「ベースライン」として保存する機能が新たに追加され、IT 統制で重要になる変更管理を強力に支援してくれる。このベースラインを起点として、例えばシステムの運用開始後に定期的に設定状態を再チェックし、ベースラインとの比較を行う。

```
# bastill_drift --from_baseline test-baseline
# more /var/opt/sec_mgmt/bastille/log/Assessment/Drift.txt
 1,2c1,2
 < AccountSecurity.ABORT_LOGIN_ON_MISSING_HOMEDIR="N"
 < AccountSecurity.NOLOGIN="N"
 ---
 > AccountSecurity.ABORT_LOGIN_ON_MISSING_HOMEDIR="Y"
 > AccountSecurity.NOLOGIN="Y"
 5,7c5,8
 < AccountSecurity.SU_DEFAULT_PATHyn="N"
 < AccountSecurity.create_securetty="N"
 < AccountSecurity.hidepasswords="N"
 ---
 > AccountSecurity.SU_DEFAULT_PATH="/sbin:/usr/sbin:/usr/bin"
 > AccountSecurity.SU_DEFAULT_PATHyn="Y"
 > AccountSecurity.create_securetty="Y"
```

```
> AccountSecurity.hidepasswords="Y"
10,11c11,13
< AccountSecurity.system_auditing="N"
< AccountSecurity.umaskyn="N"
---
> AccountSecurity.system_auditing="Y"
> AccountSecurity.umask="027"
> AccountSecurity.umaskyn="Y"
＜以下略＞
```

図 5：ベースラインからの変更点の表示例

図 5 を見ても分かるとおり、ベースラインから変化した設定項目を洗い出すことができる。これにより、運用開始後のシステムに後から生じる新たなセキュリティ脆弱性のリスクをとらえ、プロアクティブなセキュリティ対策を講じることが可能だ。

以上、今回は Bastille による HP-UX のロックダウン機能について概観した。とりわけ新機能であるシステムアセスメント機能を活用すれば、上司やクライアントからの「セキュリティ対策状況のレポートが欲しい」といったリクエストにも即対応でき、管理者にとっては便利なツールとなるはずだ。

第 5 回

audsys と HIDS による監査と侵入検知

2008 年 5 月　テクニカルライター 吉川和巳

例えば、機密情報の流出や、Web サイトの書き換えといったセキュリティ・インシデントが発生したとき。そうしたケースにおいて HP-UX の管理者が頼れるツールとなるのが、「audsys」と「HIDS」だ。HP-UX 11i v3 に標準で備わる監査機能 audsys は、「システムコールレベルで監査ログを記録する」機能。シェル上で実行されたコマンドの履歴を単に記録するのではなく、HP-UX 上で実行されたすべてのプロセスのシステムコールをリアルタイムに記録できる。また HP が提供する侵入検知システム HIDS では、HP-UX システムに対する侵入の兆候をつねに監視し、何らかの疑わしき動きを検知するとリアルタイムに管理者に通知する。

HP-UX における監査の方法

例えば、機密情報の流出や、Web サイトの書き換えといったセキュリティ・インシデントが発生したとき、もしくはそうした事態が発生していないことを明示する必要があるとき、システムの監査記録が残されていなければ、システム管理者はなすすべがない。経済産業省が公開している「情報セキュリティ管理基準」においても、以下のように規定されている。

- **7.7.1 例外事項、その他のセキュリティに関連した事象を記録した監査記録を作成して、将来の調査及びアクセス制御の監視を補うために、合意された期間保存すること。**

このように、セキュリティ情報の監査記録は、IT 統制の基盤をなす要素だ。監査記録が残されていることで、いざ不正侵入などが発生した場合でも、その経路や情報流出の実態などをすばやく把握できる。例えば、システムログが改ざんされているかどうかを確認したり、誰が（UID）、どこから（IP アドレス）、いつ（実行日時）、何をしたか（実行コマンド）を調査したりといったことが可能となる。
では、HP-UX ベースのシステムでは、具体的にどのような監査記録を用いることができるだろうか。もっとも基本的な手段としては、ログイン履歴を表示する last コマンドや、コマンド実行履歴を表示する lastcomm コマンド、.sh_history ファイル、script コマンドなどがある。しかし、これらのコマンドやファイルは監査機能として用いるには機能不足な点が多い。例えば、履歴の改ざんが容易であったり、スクリプト内部で実行されたコマンドまでは履歴が残らなかったりといった問題がある。

HP-UX の標準監査機能、audsys

そこで HP-UX 11i v3 以降では、監査機能「audsys」が標準で利用可能となった。この監査機能は実は従来から提供されていたものだが、利用するには HP-UX を高信頼性モードで運用する必要があり、敷居が高いという難点があった。11i v2 以降では、高信頼性モードでなくても audsys を利用可能となっている。（11i v2 では Standard Mode Security Extensions のインストールが追加で必要）

audsys は「システムコールレベルで監査ログを記録する」機能である。上述した各種コマンドのようにシェル上で実行されたコマンドの履歴を記録するのではなく、HP-UX 上で実行されたすべてのプロセスのシステムコールをリアルタイムに記録できる。そのため、例えばスクリプト内部で実行された処理は記録できない、といった"取りこぼし"は発生しない。また、監査ログはバイナリ形式で記録されるため、改ざんは困難である。

システムコールをログに記録するとなると、パフォーマンス上のオーバーヘッドを気にする向きもあるだろう。しかし audsys は HP-UX のカーネルレベルに組み込まれた標準機能であり、複数の書き込みスレッドで監査記録を実行することで、オーバーヘッドはきわめて低く抑えられている。また、管理者が必要とする項目だけをログ記録するフィルタ機能を備えるほか、監査ログの自動ローテーション機能も備える。

さらに HP-UX 11i v3 以降に搭載されている audsys では、監査対象とするシステムコールのリストを「プロファイル」としてまとめて管理できる。例えば、あらかじめ用意されている「basic プロファイル」では、ログインやプロセス起動といった基本的な操作のみを記録する。この basic プロファイルを指定して audsys を実行することで、図 1 のようなイベント実行履歴が記録される。

```
Event:                  execve
Time:                   Wed Nov 21 17:22:43 07 JST
PID:                    8025
PPID:                   8006
User/Grp:                 109/20(test/users)
Groups:                  20(users)
Effective privileges:          "BASIC"
Permitted privileges:           "BASIC"
Retained privileges:            "BASIC"
```

```
Compartment id:                 2
Audit tag:                  0:      -1:test:200711210821
TTY:                        unknown
Return1:                    0
Arg 1 (file info):
given path = "/usr/sbin/getrules"
inode = 12671
device = 64, 0x3
mode = 0100555
owner uid/gid = 2/2
type = regular file
Arg 2 (argument list):
arg #1 = "/usr/sbin/getrules"
arg #2 = "init"
Other (file info):
inode = -1
```

図 1：audsys による exec イベントの実行履歴

こうした audsys による監査ログの表示や設定作業は、Web ベースの管理ツールである SMH（Systems Management Homepage）にも統合されており、Web ブラウザ上から監査機能の設定やログの表示を行うこともできる。

図 2：SMH での audsys 設定の例

このように、audsys の特徴は、実運用環境でもシステムや管理者に大きな負担を掛けずに利用できる点だ。HP-UX 11i v3 ベースのシステムさえあれば、SMH にて簡単な設定を行うだけで、十分な機能性と情報量を備えたセキュリティ監査機能が直ちに手に入るのである。

ホストベースの侵入検知システム、HP-UX HIDS

ここまで見てきたとおり、HP-UX における監査記録の手段としては、audsys が十分な機能を提供する。しかし、もし Web サーバーやメールサーバー、DNS サーバーなどのように、不正アクセスの脅威に日常的にさらされるサーバーを運用する場合は、監査記録だけでは万全とは言い難い面もある。例えば、実際にセキュリティ・インシデントの発生が明らかになり、迅速な対処が必要とされる状況では、大量の監査ログを眺めて原因を特定するだけの時間的余裕はない。また、システムコールレベルのログの意味を理解し、それが不正アクセスかどうかを判断するには、熟練した管理者のスキルが要求される。

HP-UX HIDS は、こうした状況でもすばやい対応を可能とする侵入検知システムである。HIDS では、HP-UX システムに対する侵入の兆候をつねに監視し、何らかの疑わしき動きを検知するとリアルタイムに管理者に通知する。監査ログを調査して侵入の形跡を見つけるまでに要する時間を省けるため、迅速な対処が実現できる。さらには、侵入検知時には"対抗策"のスクリプトを自動的に実行でき、管理者が対応を始める前の段階で侵入を防ぐことも可能だ。通常このような侵入検知システムは有償であることが多いが、HP-UX では標準で OE にバンドルされ無償で利用できる。

具体的には、HIDS では以下のような不審な操作や攻撃を検出できる。

- ログ・ファイルの改ざん
- seduid ファイルの作成
- world-writable ファイルの作成
- ほかのユーザが所有するファイルの変更
- 特定のファイルやディレクトリの変更
- ログインやログアウト
- ログイン失敗の繰り返し
- su コマンド失敗の繰り返し
- 対話型セッションの開始
- バッファ・オーバーフロー攻撃
- Race Condition 攻撃

一般に侵入検知システムには、大きく分けて「ネットワーク型」と「ホスト型」の 2 種類がある。前者はネットワーク機器の形態をとり、ネットワーク上に流れるパケットを常時監視して不正アクセスの兆候を見つけ出す仕組みである。一方、HIDS をはじめとするホスト型の侵入検知システムでは、攻撃対象となりうる個々のホストで監視用ソフトウェア（エージェント）を動作させ、ホスト内部で不正アクセスを検出するメカニズムとなっている。図 3 に、HIDS のシステム構成図を示す。

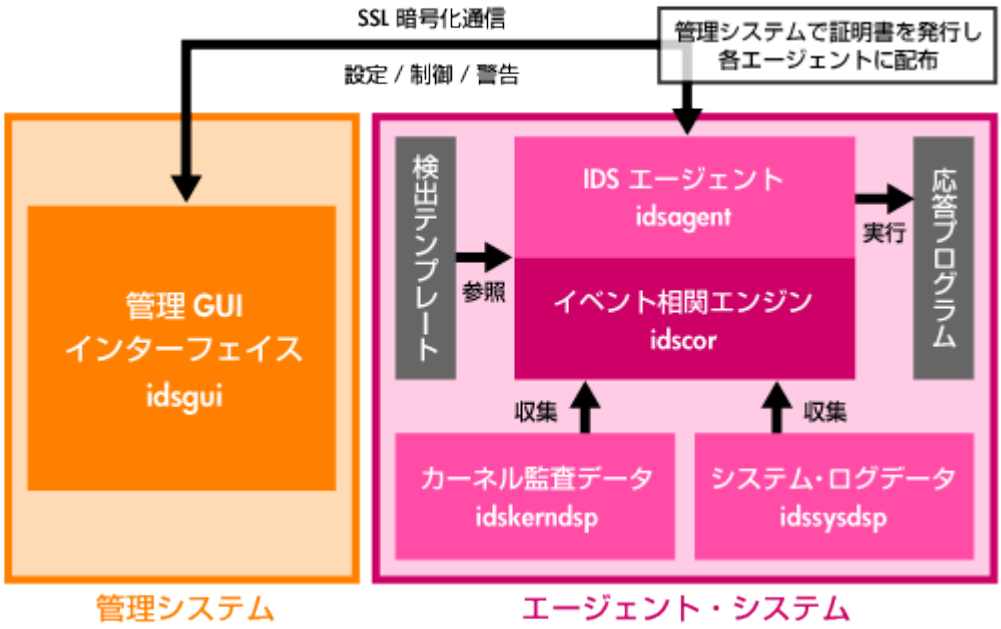

図 3：HP-UX HIDS システム構成

HIDS では、個々の HP-UX システムにインストールしたエージェントに対し、管理用 GUI を通じて設定や管理を実施する。例えば図 4 の画面では、監視対象ホストにおいてユーザのログアウトを検出したことを示している。

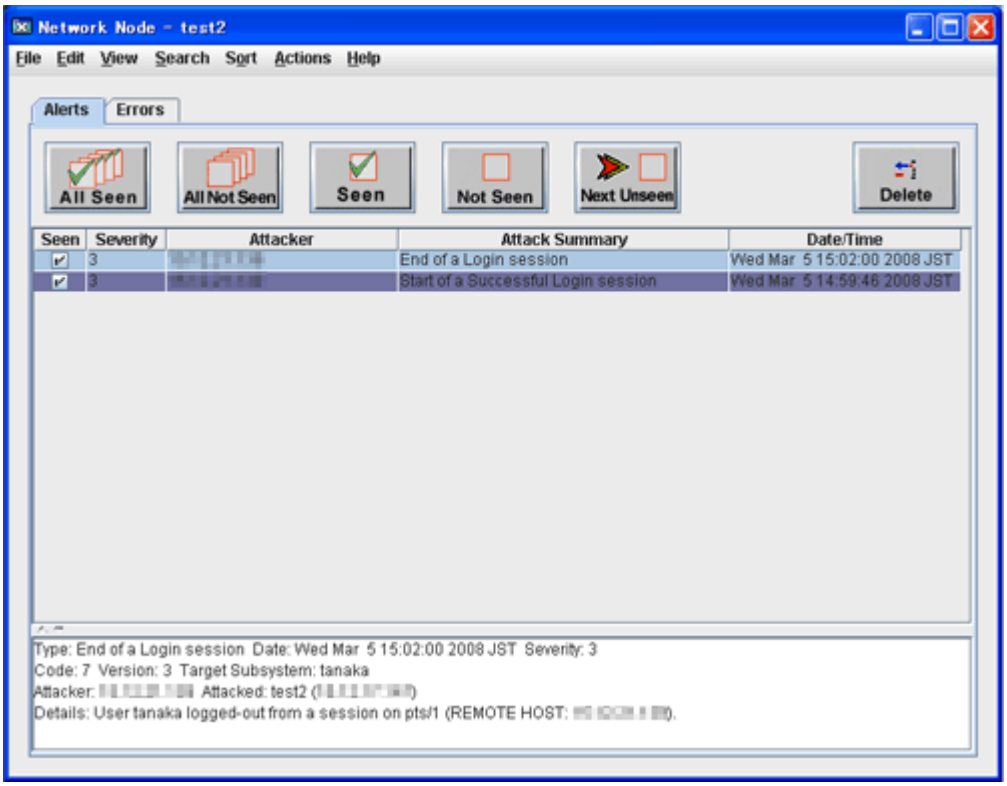

図 4：HIDS によるノード監視の例

HIDS はこう使う

では、HIDS は実際のシステム運用においてどのようなかたちで活用できるだろうか。実際の導入事例としては、情報システム事業者で PCI DSS（PCI Data Security Standard）対応に用いられたケースがある。PCI DSS とは、クレジットカード分野におけるデータ・セキュリティ基準として 2005 年に米国で策定された標準仕様だ。同仕様では、例えば以下のような基準が定められている。

- **既存のログ・データが改ざんされたときに必ず警報が発せられるように、ログに対してファイル完全性監視／変更検出ソフトウェアを使用する**

そこで上述の情報システム事業者では、この要件を満たすために HIDS を導入した。HIDS は、syslog や btmp、wtmp、sulog、mail.log といった主要なシステム・ログファイルを監視対象としてテンプレートを用意している。これらのファイルが書き換えられた場合には管理者に対してリアルタイムに通知する設定を容易に行える。実際の導入時にはテンプレートが要件に応じてカスタマイズされた。こうしたファイル監視は、図 5 のような GUI 上で簡単に設定できる。

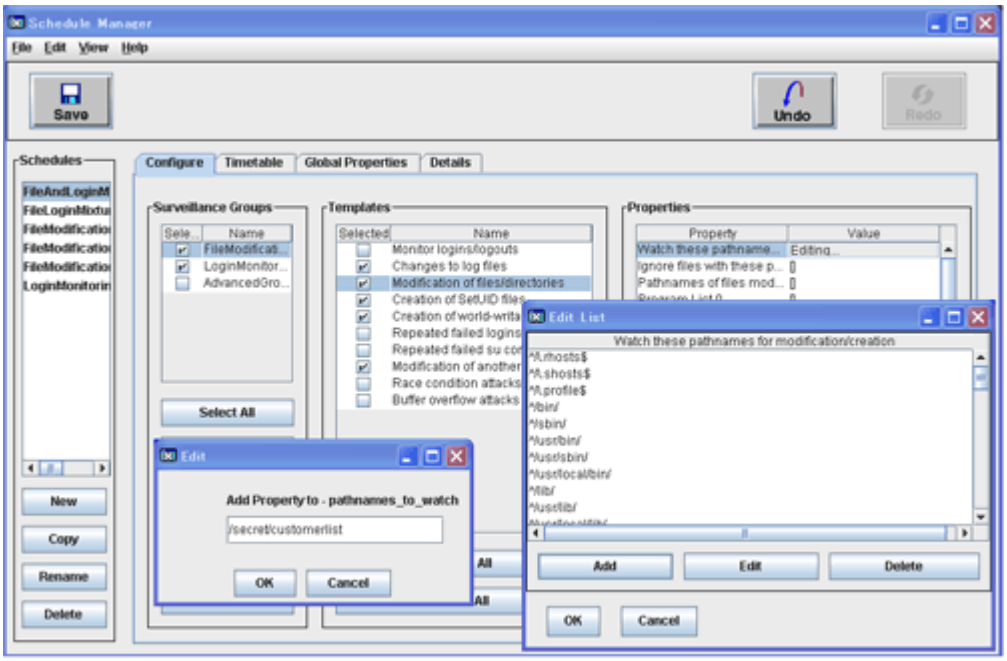

図 5：HIDS でのファイル監視設定

このほかにも、HIDS では、たとえば/etc/default/security や/etc/rbac 以下にあるセキュリティ設定用ファイルなどの書き換え検知を設定する方法も可能だ。これらのファイルは運用時に編集されることは比較的少なく、事前に変更の予定がない時に書き換えられたとすれば管理者によるミスオペや不正アクセスが発生した可能性が高いと考えられる。図 6 の例では、実際にファイルの改ざんを検知した場合の GUI 表示を示している。

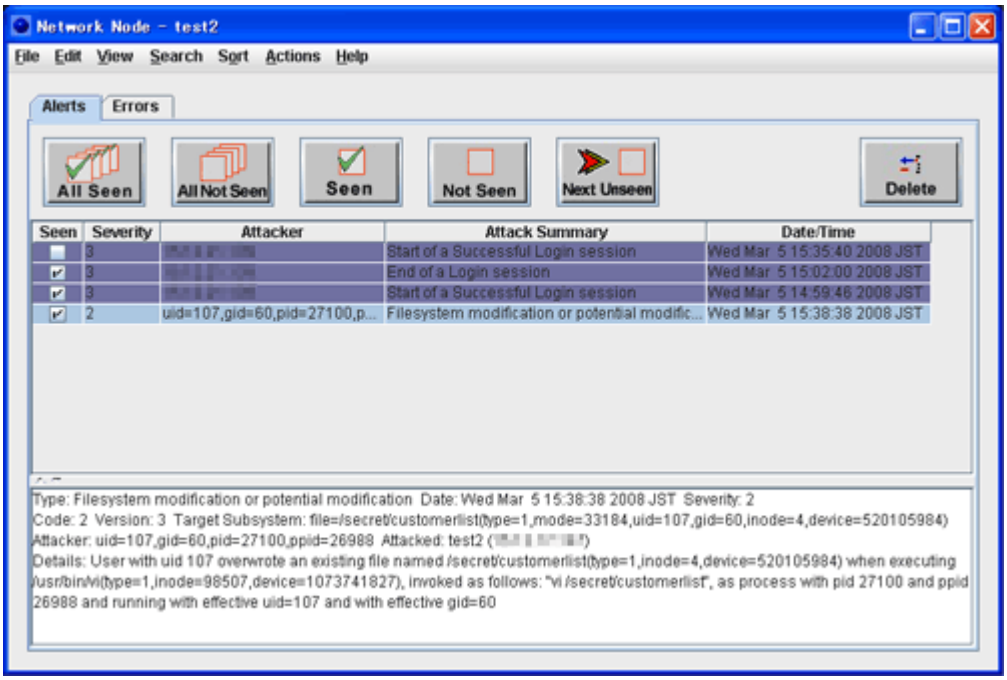

図 6：HIDS でのファイル改ざん検知例

以上、今回は HP-UX に備わる監査機能 audsys と、侵入検知システム HIDS の概要を説明した。セキュリティ・インシデントへの対応は、いざそれが発生してから実施したのでは後手に回り、社内やクライアントへの報告も苦しいものとなってしまう。しかしここで紹介したように、監査や侵入検知というと物々しい印象を受けるが、audsys や HIDS の導入は決して難しくはない。むしろ、“情報セキュリティに強い管理者”となるためには最適なツールと言えるだろう。

第 6 回

RBAC による権限分掌

2008 年 6 月　テクニカルライター 吉川和巳

Linux や UNIX の「root アカウント」、そして Windows の「Administrator アカウント」を複数の管理スタッフで共有する例は少なくない。また、本来はアクセス権限を制限すべきコンテンツ管理スタッフや開発者にも root アカウントのパスワードを教えているケースもある。こうした慣習から脱却し、「権限分掌」による IT 統制や J-SOX 対応を実現する手段として、HP-UX 11i v3 では「Role-based Access Control（RBAC）」を提供する。RBAC により、root が持つすべての権限のうち個々の作業に必要な権限だけをユーザに付与することで、root アカウントの乱用によるセキュリティ・リスクの増加を抑えられる。

IT 統制の敵は「root の使い回し」

Linux や UNIX の「root アカウント」、そして Windows の「Administrator アカウント」を複数の管理スタッフで共有する例は少なくない。また、本来はアクセス権限を制限すべきコンテンツ管理スタッフや開発者にも root アカウントのパスワードを

教えているケースもある。もちろん、こうした慣習は IT 統制や J-SOX 対応の実現を阻むものとなる。経済産業省が公開している「情報セキュリティ管理基準」においても、以下のように規定されている。

- **7.6.1 ソフトウェア及び情報への論理アクセスは、認可されている利用者に制限すること**

root アカウントを不用意に使い回すと、本来認可されていないソフトウェアや情報にアクセス可能な状況を生みだし、情報漏洩や改ざんのリスクが高まる。例えば root のパスワードを持つスタッフが、別部門のサービスのファイルも開けることに気づき、好奇心でつい機密情報に目を通してしまう……といったことも起こりうる。企業で発生するセキュリティ・インシデントの大半が「外部からの攻撃」ではなく、このような「内部からの攻撃」であることを考えれば、例えばファイアウォール構築にかけるのと同じ程度のコストと労力を、root の使い回しを防ぐ「権限分掌」の徹底に費やしてもいいくらいだ。

HP-UX 11i v3 では、こうした権限分掌のための基盤として「Role-based Access Control（RBAC）」を提供する。RBAC では、root が持つすべての権限のうち、個々の作業に必要な権限だけをユーザに付与する。これにより、インストール作業やファイル操作といった簡単な作業のために root でログインする必要がなくなる。ただし、すべてのユーザについてこと細かにアクセス権を設定するのは面倒な作業だ。そこで RBAC では、「ネットワーク管理者」や「バックアップ担当者」といった「ロール（役割）」に対して詳細な権限を割り当て、各ユーザにはロールを設定するという方式を採用している。

RBAC の導入手順は、以下のようになる。

1. **ロールを設計する**――管理作業や管理者情報の整理を行い、役割と管理作業を明確化する
2. **ユーザにロールを設定する**――例えばネットワーク管理者やバックアップ担当といったロールを定義し、個々のユーザ・アカウントに割り当てる
3. **ロールに権限を設定する**――個々のロールがどのような「認可権限」を持つかを設定する
4. **認可権限とコマンドのマッピングを設定する**――個々の認可権限のもとで実行可能なコマンドを定義する
5. **運用手順の変更を周知する**――RBAC による運用手順導入に伴う変更を関係者に周知する

このうち、3 および 4 の作業については、RBAC に備わるデフォルトの認可権限設定を大半のケースで利用できる。例えば「hpux.security.audit（セキュリティ監査機能）」や「hpux.user.add（ユーザ登録）」といった一般的な認可権限がデフォルトで定義されており、それらをロールに割り当てて使用可能だ（表 1）。

表 1： RBAC によるロールベースのアクセス制御

	User Admin	Network Admin	Backup Operator	User	Admin
hpux.user.add	●				●
hpux.user.delete	●				●
hpux.user.modify	●				●
hpux.user.password.modify				●	●
hpux.network.nfs.start		●			●

hpux.network.nfs.stop	●		●
hpux.network.nfs.config	●		●
hpux.fs.backup		●	●
hpux.fs.restore		●	●

RBAC 管理の実際

以下の図は、RBAC で用いる管理コマンドの位置づけを示したものだ。

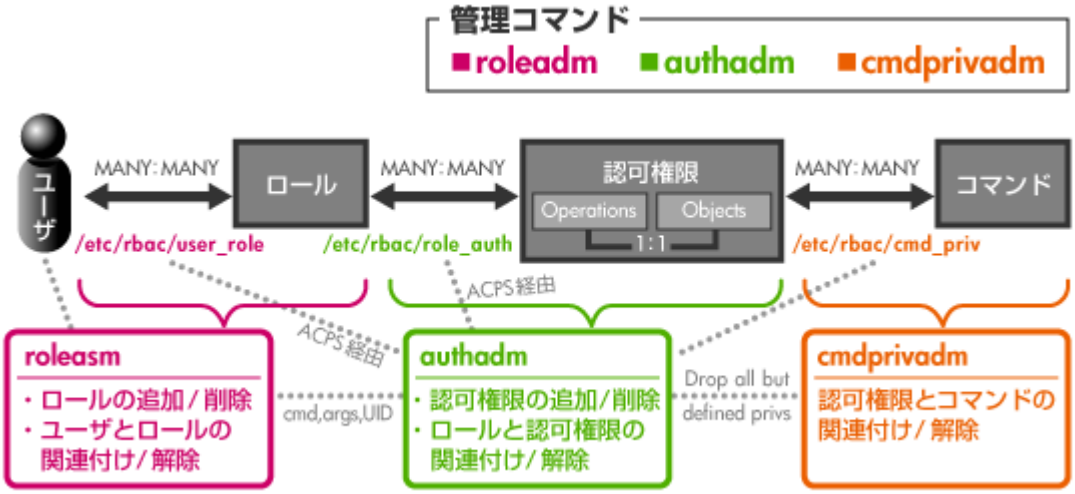

図 1：RBAC の管理コマンド

ここで示す 3 つのコマンドは、それぞれ以下の役割を担う。

- **roleadm**——ロールの追加・削除、ユーザとロールの関連付け、解除
- **authadm**——認可権限の追加・削除、ロールと認可権限の関連付け、解除
- **cmdprivadm**——認可権限とコマンドの関連付け、解除

一方、これらのコマンドと同様の RBAC 管理作業は、HP-UX に備わる管理ツール「System Management Homepage（SMH）」でも実施できる。※SMH からの管理や privsh を使用するには AccessControl が必要です。
ここでは例として、先月紹介した「audsys」コマンドの利用権限を持つ「Auditor（監査管理者）」ロールを定義する例を紹介しよう。まずは SMH の「Tools」ページにある「Role Based Access Control」という項目にて「Configure Roles」を選択する。

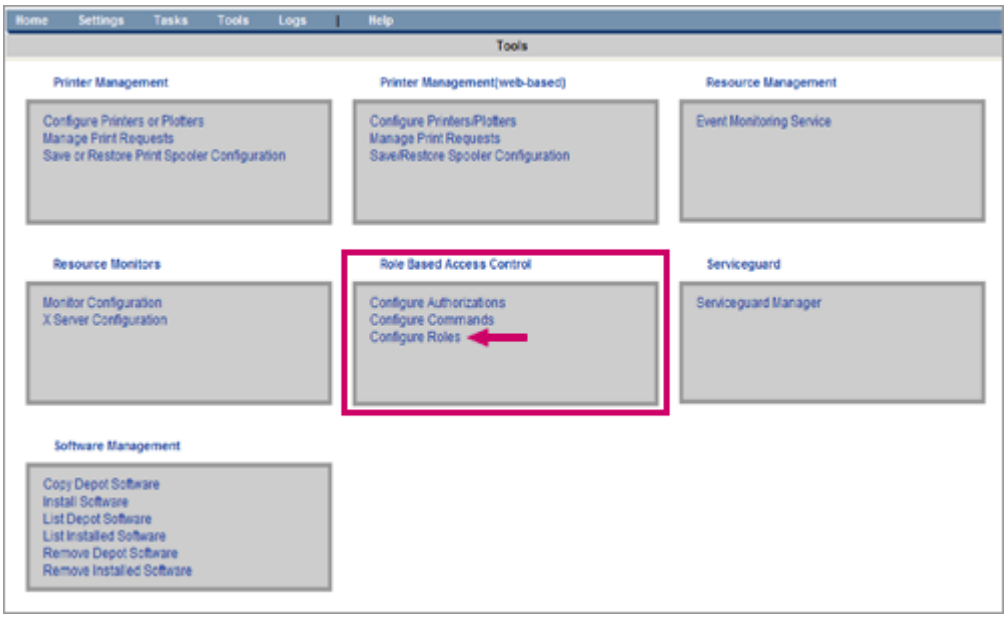

図 2：SMH による RBAC 設定

ここで表示される「RBAC Configuration」画面の「Roles」タブにて、「Add New Role」を選択し、「Role Name」として追加するロールの名称「Auditor」を入力したのちに「Add」ボタンをクリックする。こうして追加した Auditor ロールを選択し、「Assign Authorizations」をクリックして認可権限を割り当てる。ここでは、「audit」というキーワードで認可権限を検索し、audsys の一連のコマンドを利用するために必要な 3 種類の認可権限を Auditor ロールに割り当てる。

図 3：ロールに対する権限の設定

一方、こうして設定した Auditor ロールを個々のユーザに割り当てるには、「Assign Users」をクリックし、対象となるユーザ・アカウントを選択する。

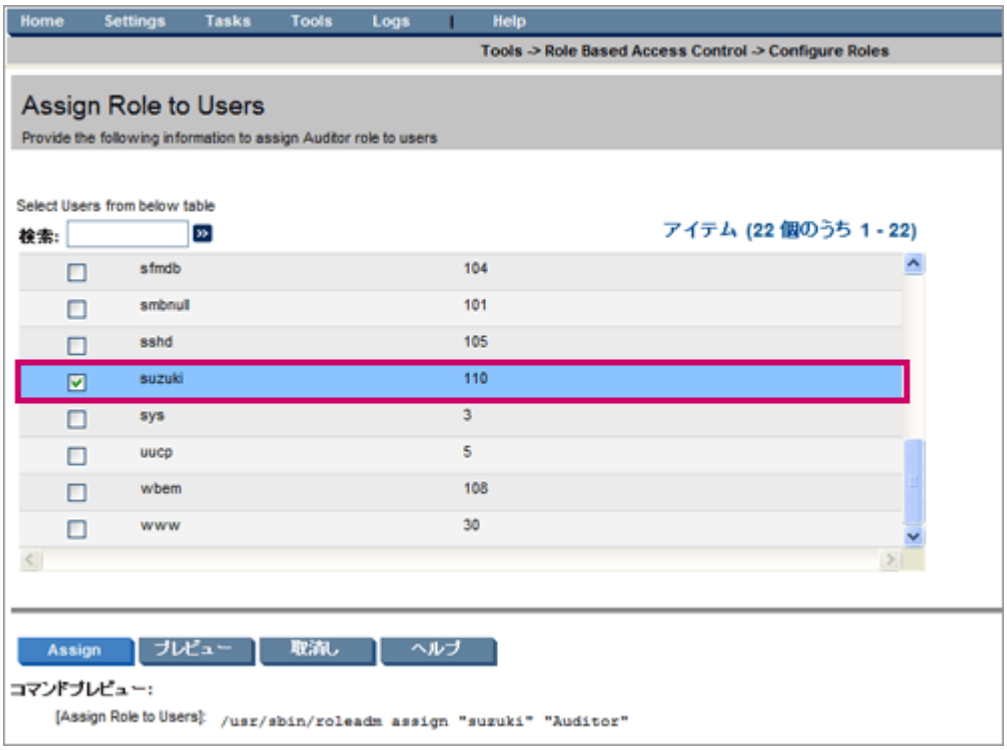

図 4：ユーザに対するロールの設定

こうして一般ユーザに対して audsys 管理のための認可権限を与えることで、root 以外のアカウントでも audsys の各種コマンドを実行可能となる。ただし、すぐに audsys コマンドを直接利用できるわけではない。実際の例を見てみよう。

```
$ id
uid=110(suzuki) gid=20(users)
$ audsys
you do not have access to the auditing system
```

このアカウント suzuki には audsys 実行の認可権限が与えられているが、上記例のようにそのままでは直接起動できない。「privrun」コマンドをラッパーとして利用して audsys を起動する必要がある。

```
$ privrun audsys
warning: /etc/audit/audnames does not exist
auditing system is currently off
current trail: ** unknown **
next    trail: ** unknown **
statistics-     afs Kb  used Kb  avail %    fs Kb  used Kb  avail %
current trail:       ** no data available **
next    trail:       ** no data available **

auditing system, when enabled, will write to 1 file(s)
```

また、アカウントのログインシェルを「privsh」に変更することで、privrun コマンドなしでも直接対象の管理コマンドを起動できるようになる。

```
$ chsh suzuki /usr/bin/privsh
```

この privsh を使えば、root アカウントと同じように各種の管理コマンドを直接起動できるため、これまで root アカウントを使っていた場合と変わらない使い勝手を実現できる。
また RBAC では、管理者が一時的に離席した場合に他人がシェルを不正利用することを防ぐために、「再認証機能」を備えている。これは、認可権限を持つコマンドであっても、実行時に逐一パスワードを要求する仕組みだ。

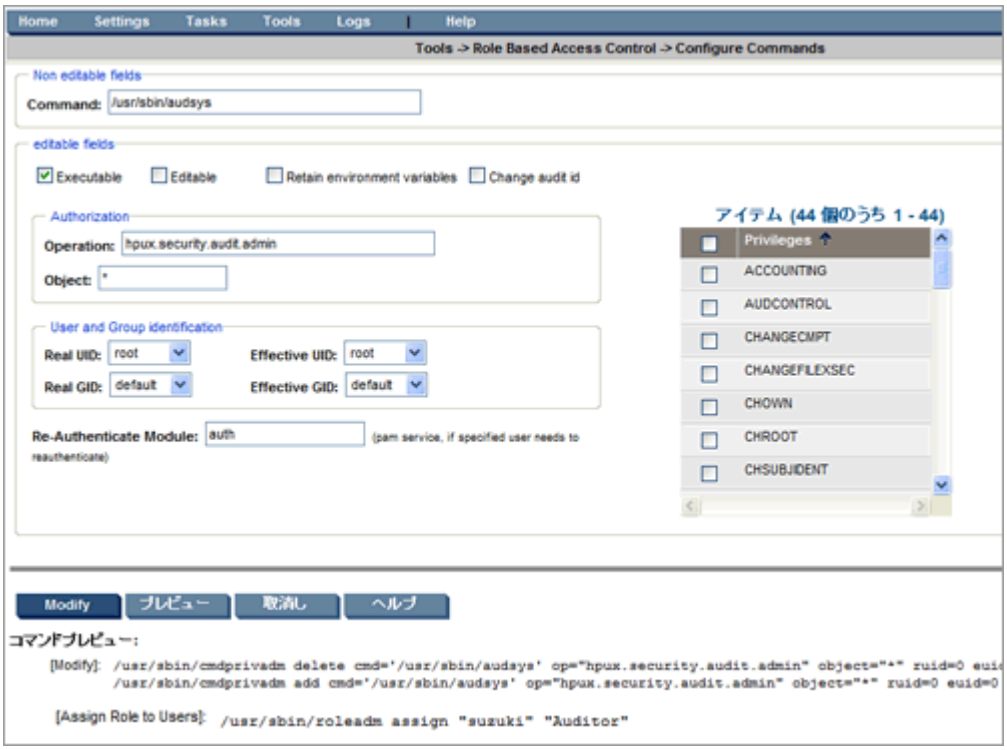

図 5：再認証設定

上図のように、SMH 上で再認証設定を実施しておけば、例えば suzuki アカウントにて audsys コマンドを実行する場合にも毎回パスワード入力が必要となる。

```
$ audsys
Password:   ←パスワードを要求
auditing system is currently on
current trail: /tmp/audout
next    trail: none
statistics-     afs Kb  used Kb  avail %    fs Kb  used Kb  avail %
current trail:     2048      692       66 19251200  8563472       56
next    trail: none

auditing system is actively writing to 1 file(s)
```

以上、今回は RBAC による権限分掌の実際を紹介した。ここで見たとおり、RBAC は SMH の GUI を通じて簡単に設定できるうえ、privsh を使えば従来どおりの使い勝手を再現できる。こうした簡単な設定作業だけで、冒頭で述べたようなセキュリティ・リスクを低減できることがおわかりいただけたはずだ。

最終回

LDAP によるアカウント統合化

2008 年 7 月　テクニカルライター 吉川和巳

HP-UX を搭載したサーバーの台数が増えてくると、アカウントの管理が懸案となる。例えば数台、数 10 台といったサーバーのそれぞれにユーザ・アカウント登録を行い、整合性を保持するのは大変面倒なうえ、古いアカウントの放置による脆弱性のリスクも生じる。これまで UNIX 環境では NIS や NIS+による一元管理が一般的であったが、セキュリティ脆弱性の問題などから最近では LDAP（Lightweight Directory Access Protocol）ベースのディレクトリ・サービスがアカウント統合に利用されつつある。そこで本連載の最終回となる今回は、HP-UX に備わる LDAP サーバー機能「Red Hat Directory Server」と LDAP クライアント機能「LDAP-UX」を利用したアカウント一元管理の実際を紹介する。

LDAP によるアカウント管理が主流に

HP-UX を搭載したサーバーの台数が増えてくると、アカウントの管理が懸案となる。例えば数台、数 10 台といったサーバーのそれぞれにユーザ・アカウント登録を行い、整合性を保持するのは大変面倒な作業だ。とりわけ、最近では Integrity VM を始めとする仮想化技術の普及にともない、管理対象となる HP-UX イメージの数も増加し、管理者にとってはアカウントの一元管理がやっかいな問題となりつつある。手間が掛かるという点もさることながら、例えば古いアカウントが削除されずに放置されるといった脆弱性を生み出す原因ともなる。
もちろん、こうしたアカウント管理は HP-UX に限ったことではない。例えば Windows 環境では、Active Directory による一元管理が一般的だ。また HP-UX をはじめとする UNIX 環境では、「NIS」や「NIS+」による一元管理が伝統的に広く利用されてきた。しかし、NIS/NIS+はセキュリティ脆弱性の問題などから、最近ではあまり使われなくなっている。

そこで近年では、NIS/NIS+に代わるアカウント管理の手段として、LDAP（Lightweight Directory Access Protocol）ベースのディレクトリ・サービスが利用されるようになった。ディレクトリ（directory）とは、住所録や電話帳を意味する。つまりディレクトリ・サービスとは、「ネットワーク上の電話帳サービス」と言えばわかりやすいだろう。具体的には、以下のような情報の保管や検索に広く用いられている。

- ID 情報（ユーザ名とパスワード、アクセス権など）
- 社員名簿（名前、メールアドレス、電話番号、組織名など）
- PKI（公開鍵基盤）におけるデジタル証明書
- ネットワーク機器やアプリケーションの設定情報
- セキュリティ・パッチ情報

LDAP 対応のディレクトリ・サーバーは、「LDAP サーバー」とも呼ばれる。ディレクトリ・サービスを利用するクライアント、すなわち OS やアプリケーション、ネットワーク機器などは、この LDAP プロトコルを通じてディレクトリ・サーバーにアクセスし、図 1 に示すようなツリー構造に基づいて情報の検索・更新・認証を実施する。

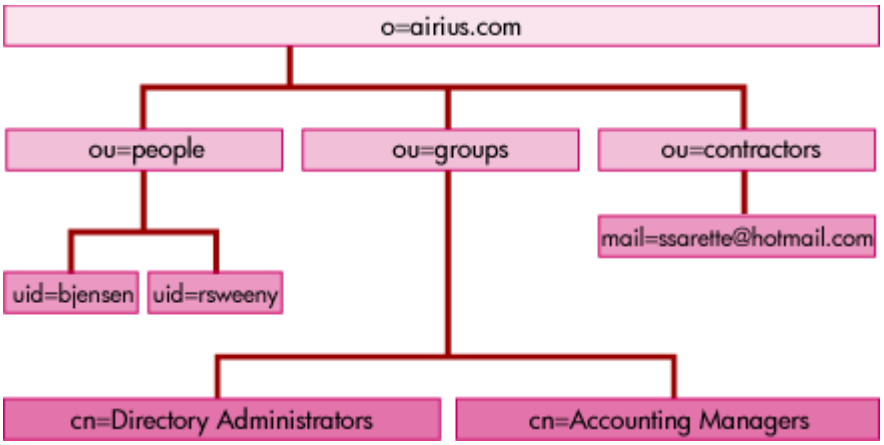

図 1：LDAP プロトコルのツリー構造

HP-UX の LDAP サポート

HP-UX は、この LDAP に対応した「サーバー」と「クライアント」の両方を標準でサポートしている。まず、LDAP サーバーとしては業界標準の商用ディレクトリ・サーバーである「Red Hat Directory Server（以下、RHDS）」を搭載する。RHDS は社内利用に限りユーザ数無制限で利用可能なほか、SSL 暗号化への対応、レプリケーション機能などに対応しており、LDAP サーバーとして豊富な機能を備えつつも無償で使うことができる優れものだ。
一方、LDAP クライアントとしては、「LDAP-UX」を備える。LDAP-UX は、LDAP サーバーに接続する LDAP クライアント機能を中心として、各種の移行ツールや管理ツール、HP-UX の認証機能（PAM や NSS）のサポート、SMH との統合、そして NIS/LDAP ゲートウェイ機能などを提供する。

この両者を活用すれば、例えば HP-UX 上の LDAP サーバー（RHDS）にてアカウント情報を一元管理し、UNIX 環境と Windows 環境のアカウント統合も実現可能だ。そこで後半では、LDAP-UX による LDAP クライアント機能の実際を紹介したい。

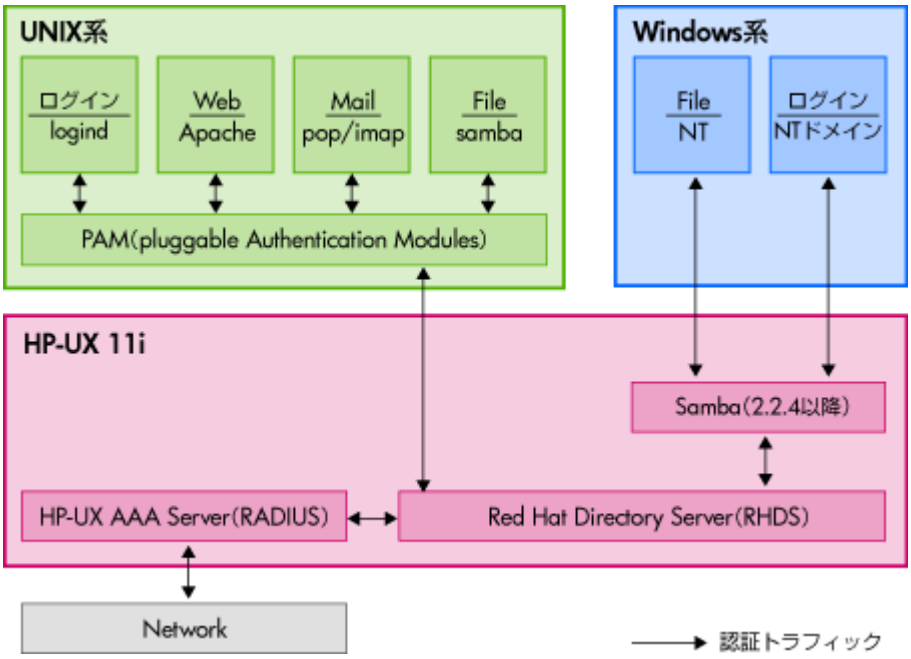

図 2：RHDS による認証統合化の例

LDAP-UX クライアント・サービスの構成

まずは、LDAP-UX のクライアント・サービスを概観しよう。図 3 に示すように、LDAP-UX では、ldapclientd デーモンが LDAP クライアントとして動作し、LDAP サーバーに認証情報などの問い合わせを実行する。そこで得られた認証情報が HP-UX の組み込み可能な認証モジュールである PAM(Pluggable Authentication Module)や名前解決のサービスを切り替える NSS（Name Service Switch）に渡され、通常の HP-UX のアカウント情報と同じように扱われる仕組みだ。こうして登録された LDAP アカウントは、ネットワーク上の複数の HP-UX サーバーから共有可能であり、個々の HP-UX に同じアカウント情報を繰り返し登録する手間を省くことができる。

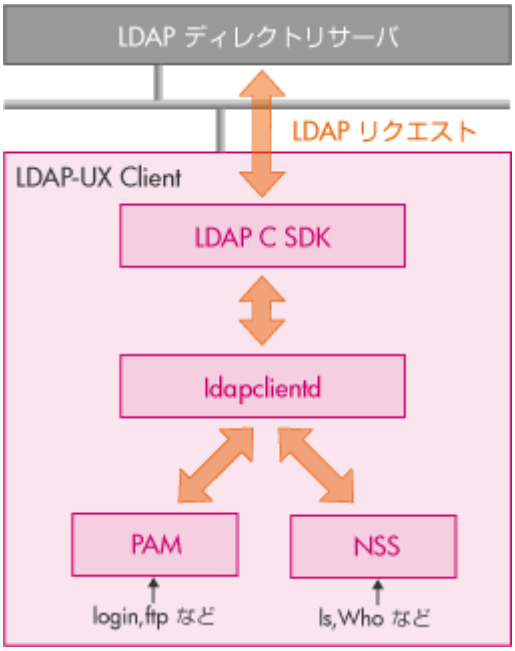

図 3：LDAP-UX クライアント・サービスの構成

こうした LDAP-UX によるアカウント統合を実際に導入するには、まずは LDAP サーバーとなる RHDS の SSL 設定を実施する必要がある。RHDS では GUI ベースの設定ツールを備えており、SSL 証明書の登録等を比較的簡単に実施できる。

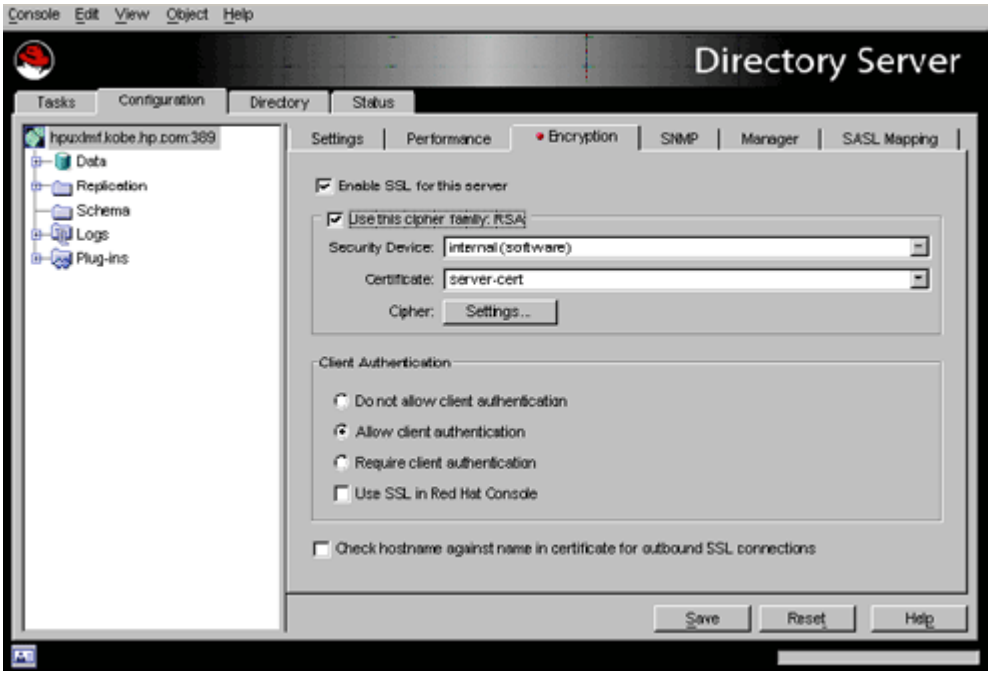

図 4：RHDS による SSL 設定の例

一方、LDAP クライアントとなる LDAP-UX では、証明書管理用のコマンド certutil を用いて、証明書データベースの初期化や証明書の登録を実施する。

証明書データベースの初期化

```
/opt/ldapux/contrib/bin/certutil -N -d /etc/opt/ldapux
```

LDAP サーバーの証明書の登録

```
/opt/ldapux/contrib/bin/certutil -A -n my-server-cert \
-t "P,," -d /etc/opt/ldapux -a -i /tmp/mynew.cert
```

セットアップ時に SSL のポートを指定（デフォルト 636）

```
/opt/ldapux/config/setup
```

NSS および PAM の設定（LDAP 環境向けテンプレートのコピー）

```
cp /etc/pam.ldap /etc/pam.conf
cp /etc/nsswitch.ldap /etc/nsswitch.conf
```

以上の設定により、HP-UX の通常のローカル・アカウント以外に、LDAP サーバー上に保存される「LDAP アカウント」が登録可能となる。HP-UX 11i v3 の 2007 年 9 月版以降では、この LDAP アカウントを SMH 上で管理できるようになった。図 5 は、SMH による LDAP アカウントの追加の例である。

図 5：SMH による LDAP アカウントの追加

ちなみに、UNIX 環境や Linux 環境では一般的に、PAM と LDAP の連携に pam_ldap が用いられるが、LDAP-UX では HP 独自に pam_authz を提供している。この pam_authz は、LDAP サーバーに登録されたさまざまな属性情報に基づいてシステムにログイン可能なユーザを制御できる機能を備える。例えば、ある LDAP アカウントがログイン可能なホスト名の一覧を LDAP サーバーに登録したり、会社組織上の属性情報に応じてログインの許可／不許可を制御したりすることが可能だ。

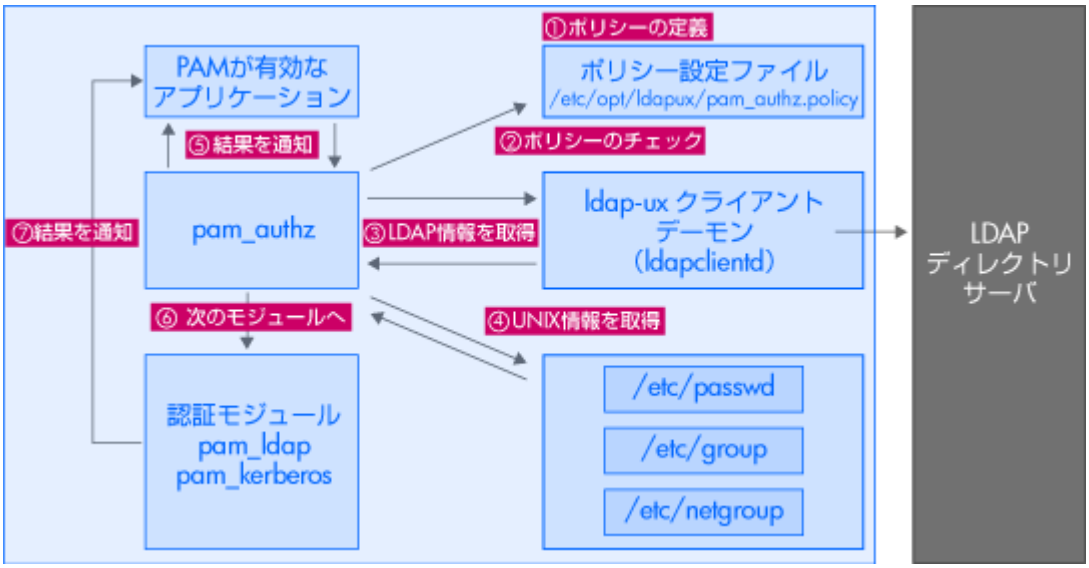

図 6：pam_authz による PAM と LDAP の連携

以上、今回は LDAP-UX を利用したアカウント統合を紹介した。さて、HP-UX に備わる最新のセキュリティ機能にスポットをあてる本連載は、今回で最終回となる。さまざまなセキュリティ上の課題に対して、「HP-UX ならばこう解決できる」と答えるためのヒントを本連載で提供できたならば幸いである。

コラム：HP-UX 11iv3 が Common Criteria 認定を取得

2008 年 3 月、HP-UX 11i v3 は IT セキュリティの国際共通基準である「Common Criteria （以下、CC）/ISO15408」認定を取得した。CC は、ISO15408（正式名：ISO/IEC 15408）としても知られるセキュリティ基準の国際標準規格。情報技術をセキュリティの観点から、製品及びシステムが適切に設計され、その設計が正しく実装されているかを評価する国際標準規格だ。HP-UX11i v3 は、同規格の「ALC_FLR.3 付き評価保証レベル 4（EAL4+）」認定を取得した。

今回、HP-UX 11i v3 が CC/ISO15408 認定を取得したことにより、2008 年 3 月以降に資産登録を行う HP-UX 11i v3 および、それを搭載するサーバーなどについて、税制上の優遇措置（減税）に対する申請条件のひとつを満たすことになる。

HP-UX

www.hpe.com/jp/hpux

Hewlett Packard Enterprise