Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

仮想マシンモニタ BitVisor のための ロールベースアクセス制御機構の設計と実装

幾世知範[†] 平野学[†] 品川高廣[††] 奥田剛[†††] 河合栄治[††††]
加藤和彦[††] 山口英[†††]

企業や官公庁などの組織におけるユーザ端末のセキュリティを仮想マシンモニタを導入して高める取り組みが行われはじめている．セキュリティ用途の仮想マシンモニタである BitVisor を組織内のユーザ端末に導入することで，組織内に分散して存在するユーザ端末上でのユーザの行動をユーザ ID に基づいて制御することができる．本研究ではセキュリティ機能の組み込まれた仮想マシンモニタである BitVisor の実運用に向け，ロールベースアクセス制御機構の設計と実装を行った．ロールベースアクセス制御機構では，ロールに対応するコンフィグレーションファイルの自動更新，ロールに基づくアクセス制御，権限の無いアクセスに対する画面を介した警告通知を行う機能を提供する．本稿では提案システムでゲスト OS の起動と USB メモリの利用のアクセス制御を行った結果を報告する．

Design and Implementation of a Role-based Access Control Mechanism for Virtual Machine Monitor “BitVisor”

Tomonori Ikuse[†], Manabu Hirano[†], Takahiro Shinagawa[††], Takeshi Okuda[†††], Eiji Kawai[††††], Kazuhiko Kato[††] and Suguru Yamaguchi[†††]

A virtual machine monitor can be used to improve security in governmental and commercial organizations. BitVisor is a thin hypervisor specialized for security purpose. BitVisor has security functions that enable administrators to control end-users’ operations using the user ID. In this paper, we show a design and an implementation of a role-based access control extension for BitVisor. The role-based access control mechanism for BitVisor consists of an automatic update function of configuration files, an access control function based on roles, and a warning function for unauthorized accesses through a display. In this paper, we show the result of prototype implementation that includes role-based access control for booting a guest OS and use of USB thumb drives.

1. はじめに

近年，コンピュータを用いた業務が一般的となり，企業や官公庁などの組織では管理しなければならないコンピュータが増加している．このように増加したコンピュータからの情報漏洩などを防ぐため，これまでにセキュア OS の開発が行われてきた[1][2]. 最近は仮想マシンモニタ(VMM)を用いてセキュリティを高める取り組みが盛んに行われている．VMM のサーバ側での利用が進められる一方，VMM をクライアント側のユーザ端末で動作させて情報漏洩などを防ぐ研究も進められている[3][4].

2009 年，セキュリティ用途の小型で高速に動作する VMM である BitVisor[3]のバージョン 1.0 が公開された. BitVisor は VMM 層での IC カードを用いたユーザ認証, USB メモリのような記憶装置の暗号化，VPN による暗号化通信などのセキュリティ機能を備えている．BitVisor ではユーザ認証で得たユーザ ID に基づいてセキュリティ機能を提供する考え方が導入されているため，BitVisor を組織内のユーザ端末に導入することで，組織内に分散して存在するユーザ端末をユーザ ID に基づいて管理することができる．しかしながら，現在のところ，実際の組織での運用に必要となるユーザのロール（組織における役職など）に基づく強制アクセス制御の機能やコンフィグレーションファイルの自動更新機能が不足している．また，コンピュータ上でのユーザの行動を管理するセキュリティシステムとして運用する場合に要求される，ユーザが権限のない操作を行った場合に警告する仕組みも実現されていない．

そこで，本研究ではロールベースアクセス制御ができるように VMM を用いた強制アクセス制御機構を拡張するとともに，コンフィグレーションファイルなどの自動更新機能を新たに実装する．さらに，ユーザが権限の無い操作を行った場合に，VMM からユーザに対してディスプレイを通じて警告を行う機能を実装する．BitVisor の特徴と組織での利用方法を考えた上で実装における要件をまとめ，BitVisor に実装するロールベースアクセス制御機構の設計と実装を行う．動作検証ではゲスト OS の起動と USB メモリの利用をロールに対応するアクセス制御ポリシファイル（以下，ポリシと呼ぶ）に基づいて制御できることを確認する．

本稿では，2 節で多くのユーザを抱える組織における分散されたコンピュータの管理の課題について述べ，続く 3 節で仮想マシンモニタをセキュリティシステムに応用

[†] 豊田工業高等専門学校 専攻科
Toyota National College of Technology
[††] 筑波大学
University of Tsukuba
[†††] 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
[††††] 情報通信研究機構
National Institute of Information and Communications Technology

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

する従来研究を示す．そして，4 節で BitVisor の特徴と課題を説明する．5 節では実装における要件を述べ，6 節でロールベースアクセス制御機構の設計を行う．7 節で BitVisor への実装を示し，8 節で動作検証した結果を報告する．9 節では BitVisor に実装したロールベースアクセス制御機構の有用性と拡張性およびセキュリティについて考察する．最後に 10 節で今後の課題を示し，まとめる．

2. 組織内に分散されたコンピュータの管理の課題

一般的に企業や官公庁などの組織はユーザ端末やサーバなどの多くのコンピュータを抱えている．特にユーザ端末は組織内に分散して存在しており，そのような状態にあるコンピュータを適切に管理することは困難である．このため，企業や官公庁では管理を適切に行うことができず，多くの情報漏洩を発生させてしまっている[5].

情報漏洩を防ぐため，組織内の管理者は OS やアプリケーションのレベルでセキュリティシステムを導入している．しかし，OS の脆弱性を攻撃することでセキュリティ機能が回避されてしまう場合も考えられる．

また，別の方法としてシンクライアントを導入することによる情報漏洩対策も行われている．シンクライアントでは，ユーザ端末は最小限の機能のみを持ち，サーバ側がアプリケーションやファイルの管理を行う．ユーザ端末とサーバの間でやりとりされる情報はユーザからの入力とユーザ側で表示する画面情報だけである．ユーザ端末側に重要な情報が存在しないため，シンクライアントは情報漏洩の対策として効果的である．しかし，パフォーマンスや可用性がネットワークの信頼性に依存してしまう欠点がある．

近年は，新たな手法として VMM を利用してセキュリティを高める取り組みが行われはじめている．VMM はシンクライアント環境を構築するためにサーバで利用される場合が多いが，VMM が OS と隔離された環境で動作することに注目して VMM にセキュリティ機能を組み込む研究が行われている．本研究では組織内に分散されたユーザ端末を管理するため，セキュリティ機能が組み込まれた VMM をユーザ端末に導入する場合を扱う．

3. 仮想マシンモニタを用いたセキュリティシステム

VMM はコンピュータ資源を管理し，ゲスト OS に割り当てを行う．そのため，VMM によって資源管理が適切に行われている場合，ゲスト OS は同一コンピュータ内で動作している他のゲスト OS に対して干渉することはできない．これにより，複数のアプリケーションを隔離されたゲスト OS 上で別々に動作させることでセキュリティを確保することができる．この特徴を活かしてセキュリティシステムを構築する研究が行われている[4][6][7]．また，実ハードウェアのリソースを VMM が管理しているという特徴を利用したセキュリティ機能の研究も行われている．sHype[8]はサーバ統合を行った際のセキュリティを高めるため，VMM に組み込まれた強制アクセス制御機構でゲスト OS 同士の干渉を防いでいる．一方で，セキュリティ機能を組み込んだ VMM を利用することで，一般的な OS を導入しているコンピュータにセキュリティ機能を付加する取り組みも行われている．Overshadow[9]ではメモリを仮想化し，暗号化することでアプリケーションを OS カーネルから保護する機能を実現している．また，SecVisor[10]では OS カーネルを保護し，カーネルの完全性を保つ機能を実現している．

このように，VMM はセキュリティを実現するために利用されているが，VMM の特権が奪われると，システム全体が危険にさらされてしまう．このため，セキュリティ機能を提供する VMM は安全である必要がある．VMM の信頼性を高めるためには，コード数を少なくすることが有効である[11]．そのため，セキュリティ機能を提供する VMM はその用途に応じてできるだけコード数を少なくし，小さく実装する必要がある．しかし，現在主に利用されている VMware や Xen はさまざまな機能を持っているため，コード数が多くなってしまっている．

本研究で設計と実装の対象とする BitVisor はクライアントのユーザ端末での利用を目的としたセキュリティ用途の VMM である．BitVisor は VMM として動作するための必要最低限の VMM の機能と，デバイスドライバから構成されている．そのため，一般的に利用されている VMM に比べて小型であり，高速に動作する．

4. BitVisor におけるユーザ ID に基づいた強制アクセス制御と課題

4.1 BitVisor の概要

BitVisor はユーザ端末のセキュリティを高めることを目的とした高速に動作する小型の VMM である．BitVisor は VMM 層での IC カードを用いたユーザ認証，USB メモリやハードディスクなどの記憶装置の強制暗号化，VPN で通信を暗号化するセキュリティ機能などを持っている．これらの暗号化処理はユーザに意識させることなく透過的に行われる．

BitVisor はパラパススルー型と呼ばれるアーキテクチャを取っている（図 1）．パラパススルー型とは，セキュリティ機能の実現に必要な機能のみを仮想化し，可能な限りゲスト OS からハードウェアに直接アクセスさせる方式である．また，パラパススルー型ではパラパススルードライバを利用して I/O の捕捉を行っている．パラパススルードライバは，デバイスの初期化やエラー処理などをゲスト OS のドライバに任せ，ゲスト OS と特定のハードウェアとの間で送受信されるデータの監視と制御（暗号化）のみを行う．このため，セキュリティ機能を実現しても小さく実装することが可能となっている．このように，ゲスト OS が直接ハードウェアにアクセスしているため，動作させることのできるゲスト OS は 1 つに限られている．また，Xen のドメイン 0

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

のような管理 OS は存在しない．

図 2 は従来研究[12]における VMM を用いた強制アクセス制御機構のモデルである．IC カードを利用したユーザ認証によって得たユーザ ID とポリシに基づいて，ハードウェアとゲスト OS の間でやり取りされるデータと命令をアクセス制御することが提案されている．BitVisor でのユーザ ID に基づいたアクセス制御には，この考え方が導入されている．

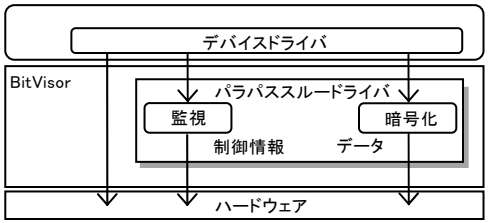

図 1　パラパススルー型アーキテクチャ

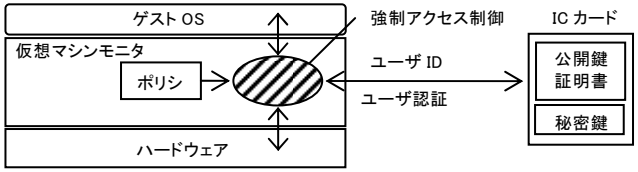

図 2　強制アクセス制御機構のモデル

4.2 ユーザ認証からゲスト OS 起動までの仕組み

BitVisor では，BitVisor の起動後に IC カードを用いたユーザ認証が行われる．ユーザ認証は IC カード内の X.509 公開鍵証明書[13]を用いたチャレンジ･レスポンス方式で行われる．ユーザ認証の際，ユーザが行うのは IC カードの PIN コードの入力だけである．

図 3 は BitVisor の起動から BitVisor 上で保護したい目的のゲスト OS（Windows XP, Vista など）の起動までの流れを表している．BitVisor の起動後， MiniOS という Initial RAM Disk 上で動作する小さな Linux カーネルが最初のゲスト OS として動作し，ユーザ認証などの BitVisor の初期処理を行う．ユーザ認証が完了すると予め組み込まれているコンフィグレーションファイルが読み込まれ，BitVisor にメモリを介して設定情報が渡される．その後，MiniOS は停止し，続いて動作させたい目的のゲスト OS が起動する．この時点でセキュリティ機能の設定情報(USB メモリを暗号化するかなど)が BitVisor に渡されており，ゲスト OS 上でのユーザの行動はこれに基づいて制限される．

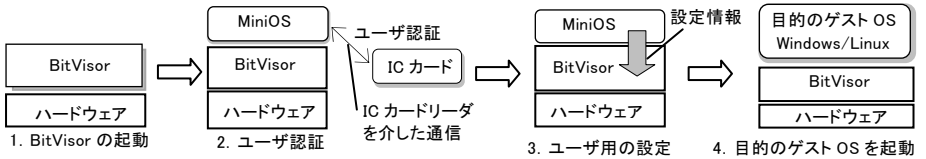

図 3　ゲスト OS 起動までの流れ

4.3 課題

ユーザの ID に応じたアクセス制御はユーザ端末を制御するために有効である．しかし，ユーザ ID ごとにアクセス制御対象に対する権限を設定すると，ユーザが組織を異動や退職した場合に，関連する全ての権限を変更もしくは消去する必要がある．また，ユーザとアクセス制御対象の数の増加に比例して設定しなければならない権限の数も増加する．そのため，多くのユーザを抱える組織においてユーザ ID ごとに多くの細かい権限を管理することは困難である．

さらに，バージョン 1.0 の BitVisor の実装ではコンフィグレーションファイルや証明書失効リスト(CRL)などがあらかじめ組み込まれているため，これらの更新を行う場合，現状では BitVisor をユーザ端末に入れ直す必要がある．多くのコンピュータが分散して存在している組織でこのような作業を頻繁に行うことは困難である．

管理者がユーザの不正行為を防ぐためにセキュリティシステムをユーザ端末に導入する場合，不正行為を抑制するため，権限の無い操作を行ったユーザに対して警告を行うことが有効であると考えられる．バージョン 1.0 の BitVisor は透過的にセキュリティを実現することを意識した実装になっているため，ゲスト OS 上でユーザが権限の無い操作を行った場合に警告をする仕組みは実現されていない．ユーザへの警告機能があれば不正行為の抑制につながると考えられる．

5. 提案システムの要件

本節では提案システムの要件について述べる．

5.1 機能要件

4.3 節で示した課題を解決するため，本研究では BitVisor にロールベースアクセス制御を導入する．Sandhu.R らによって提案された NIST モデル[14]のうち最も基本的なロールベースアクセス制御モデルである Flat RBAC を採用し，ロールごとに権限を割り当てることによって効率的な権限管理を行う．ただし，今回は BitVisor 側の機能のみを扱い，管理者がユーザ ID とロールを設定するための管理機能は扱わない．

それに加えてコンフィグレーションファイルであるポリシ及び CRL の自動更新機能を実装する．これにより，権限管理にかかるコストの削減を狙う．また，権限のな

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

い行動を行ったことを警告し，ユーザに監視下にあることを通知する仕組みについても検討し，実装する．以下に，本研究で提案するシステムが持つ機能をまとめる．

機能 1：ユーザ認証をする．
機能 2：ユーザのロールを自動取得する．
機能 3：ロールに対応するポリシを自動取得する．
機能 4：ポリシに基づく認可判定をする．
機能 5：認可判定に応じてアクセス制御する．
（今回は試験的にゲスト OS の起動と USB メモリの利用の制御を行う．）
機能 6：ユーザが権限の無い操作を行った場合に警告する．

機能 1 については BitVisor1.0 に実装されている IC カードを利用したユーザ認証機能を利用し，機能 2 から機能 6 までの機能を実装する．

5.2 実装における制約条件

4.1 節で示したように，BitVisor は自身を小型にするために VMM として動作するための最小限の機能とセキュリティの機能のみで構成されている．そのため，管理 OS が存在しておらず，実行中の BitVisor ではファイルという概念を扱うことができない．また，高速に動作するという BitVisor の特徴を損なうことなく，アクセス制御の仕組みを実装しなければならない．このような条件の下，ロールベースアクセス制御機構を VMM に実装することが本研究の狙いである．

5.3 想定する環境

多くのユーザを抱える組織で提案システムを運用する際の一例を図 4 に示す．

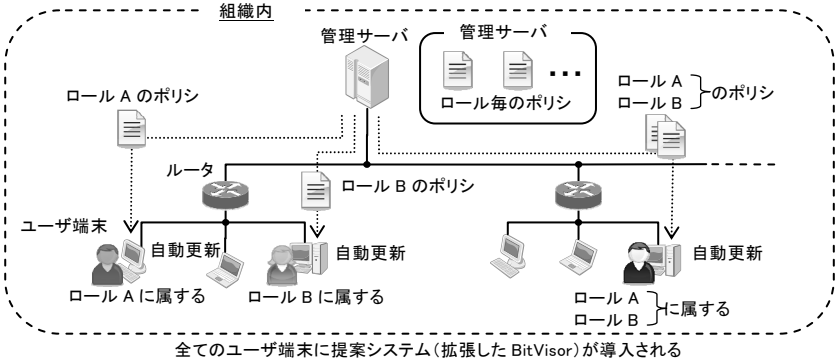

図 4　組織での提案システムの運用例

この例での組織は，組織内に多くのユーザ端末を抱えており，全てのユーザ端末に提案システム(拡張した BitVisor)が導入されていると仮定する．また，各端末にはユーザが作業を行うためのゲスト OS が 1 つだけインストールされている．ユーザは 1 つ以上のロールに属しており，ユーザ端末上でのユーザの行動は，管理者がユーザのロール毎に設定したポリシに基づいて制限される．管理者が設定したポリシのデータは管理サーバで管理される．管理者は各ユーザ端末にポリシを自動更新させ，ポリシの適用を強制する．本稿では，ロールに対応したポリシを実装においてロール毎のコンフィグレーションファイルを使って実現する．

6. ロールベースアクセス制御機構の設計

本節ではロールベースアクセス制御機構の設計について述べる．

6.1 全体構成

ロールベースアクロールベースアクセス制御ができるように拡張した BitVisor の設計を図 5 に示す．ユーザの ID とロールを管理するために，公開鍵証明書と属性証明書[13]をそれぞれ利用する．認証局(CA)の公開鍵証明書，属性認証機関(AA)の公開鍵証明書は予め管理者が Initial RAM disk に格納しておく．証明書失効リスト(CRL)，属性証明書およびコンフィグレーションファイルであるポリシは BitVisor の起動時に管理サーバから自動的にダウンロードする．ポリシにはロールに対応する権限の設定情報を記す．

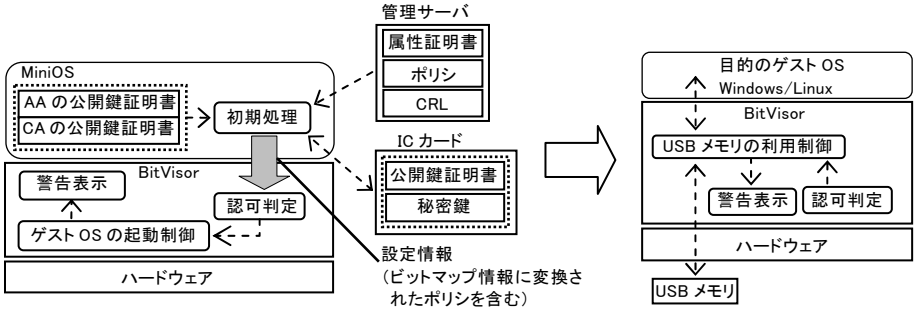

図 5　提案システムの設計

BitVisor が起動すると，MiniOS で初期処理が以下の手順で行われる．ユーザから見ると，IC カード認証で PIN を入力するだけであり，その後は自動的にユーザ用の設定で BitVisor が動作を開始する．

1. 管理サーバから CRL を取得し，CRL を更新する．
2. チャレンジ・レスポンス方式によるユーザ認証（BitVisor 1.0 に実装されている既存の機能）をし，認証に成功したらユーザ ID を取得する．
3. ユーザ ID に対応する属性証明書を管理サーバから取得する．属性証明書の正当性を検証し，公開鍵証明書との関連付けを確認する．その後，属性証明書から

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

ユーザのロールを得る．

4．管理サーバからロールに対応するポリシを取得する．ポリシの電子署名を確認し，正規のポリシであることを確認する．

5．ポリシをビットマップ情報に変換する．

本稿で拡張した BitVisor では得られたポリシのビットマップ情報を基にゲスト OS の起動と USB メモリの利用を制御する．ビットマップ情報とはアクセス制御対象ごとにユーザが権限を持っていることを'1', 持っていないことを'0'で表した権限情報のビット列である．MiniOS から BitVisor へ渡されるビットマップ情報にはどの位置の情報が何の権限を表すのかを示す情報は含まれない．どの位置の情報が何の情報を表すかは MiniOS と BitVisor で共通の構造体を用いて予め定めておく．構造体にはアクセス制御対象に対応するメンバが存在し，各メンバに権限の有無が記される．アクセス制御の際, ユーザに権限が無い場合はディスプレイを介して警告メッセージを表示する．

以下に詳しい設計を仮想マシンモニタでのロールベースアクセス制御，コンフィグレーションファイルなどの自動取得機能，警告表示機能に分けて説明する．

6.2 仮想マシンモニタでのロールベースアクセス制御

本研究ではロールベースアクセス制御で必要となるユーザのロールを属性証明書で管理する．属性証明書はユーザの属性（権限）を管理するための証明書である．提案システムでは属性証明書の属性としてユーザのロールを与える．公開鍵証明書でユーザの ID，属性証明書でユーザのロールをそれぞれ別々に管理することで，ユーザの ID を変更することなくユーザのロールのみを変更することが可能になる．

図 6 にユーザ ID，ロールと権限の関係を示す．公開鍵証明書から取得したユーザ ID と属性証明書から取得したユーザのロールは, 公開鍵証明書の所有者情報と属性証明書の保有者情報が一致することを確かめることで関連付けを行う．ユーザのロールと権限については，ユーザのロールに対応するポリシを管理サーバから取得し，ポリシに記されたロールが属性証明書から取得したロールと一致することを確かめることで関連付けを行う．また，ユーザとロール，ロールと権限は多対多に対応する．

ポリシは各ロールに対して 1 つ用意する．図 7 の場合，ロールが Employee のユーザはゲスト OS を起動して作業を行うことはできるが，USB メモリの利用は許可されないことを表す．ユーザが複数のロールに属する場合は，それぞれのロールに対応するポリシに指定されている権限を合わせた権限をユーザの権限とする．

ポリシには署名を付け，公開鍵証明書と属性証明書，CRL，ポリシは利用する前に改ざんされていないことを検証する．検証には管理者によって Initial RAM disk に保存された CA の公開鍵証明書と AA の公開鍵証明書を用いる．これらの公開鍵証明書は信頼できるものとして利用する．

本研究の提案の利点は BitVisor で高速にアクセス制御を実行させるため，ポリシをあらかじめビットマップ情報に変換して利用することである．アクセス制御の度に VMM がポリシに対して処理を行う場合，アクセス制御の対象の数に比例して処理速度が低下してしまう．ビットマップ情報にあらかじめ変換しておくことで，ポリシに対する処理にかかる時間を省くことができる．BitVisor ではビットマップ情報が格納されている構造体のメンバを参照するだけで権限の有無を知ることができるため，BitVisor の高速に動作するという特徴を損なうことなくアクセス制御を行うことが可能となる．

今回はゲスト OS の起動と USB メモリの利用をロールベースアクセス制御するプロトタイプシステムを開発する．ゲスト OS を起動させる権限がある場合はゲスト OS の起動処理を行い, 権限の無い場合は VMM を停止させる. USB メモリの利用制御は, USB メモリとゲスト OS のドライバ間でやり取りされるデータをパラパススルードライバで制御して行う．USB メモリを利用する権限がある場合はデータをそのまま通し, 権限がない場合はデータを破棄する．どちらのアクセス制御も認可判定はビットマップ情報に変換されたポリシに基づいて行う．

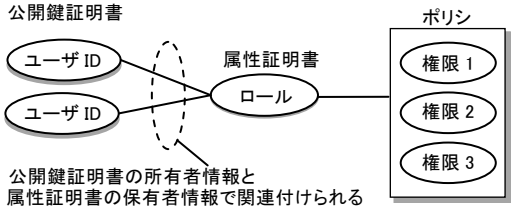

図 6 ユーザ ID とロールと権限の関係

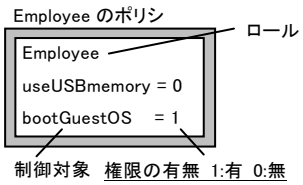

図 7 ポリシの例

6.3 コンフィグレーションファイルと CRL の自動更新機能

管理サーバではコンフィグレーションファイルであるポリシ，CRL 及び属性証明書を管理する．アカウント情報およびディレクトリの管理には LDAP[15]を利用する．LDAP では図 8 のようにユーザ ID およびユーザのロールを階層的に管理する．LDAP で管理しているユーザ ID は公開鍵証明書から取得できるユーザ ID と対応し，LDAP で管理するロールが属性証明書から取得されるユーザのロールと対応する．ユーザのホームディレクトリに属性証明書, ロールのホームディレクトリにポリシを保管する．CRL は公開用フォルダに保管する．組織の管理者が管理サーバを運用し，必要に応じてこれらのファイルの更新を行うものとする．管理者がどのようにファイルの更新および設定を行うかといった運用上の問題については本研究では扱わない．

本稿で拡張した BitVisor が導入されたユーザ端末は初期処理で管理サーバからポリシと CRL をダウンロードし，自動更新して利用する．また，属性証明書についても管理サーバから取得して利用する．管理サーバから取得した各ファイルは利用する前に署名の検証を行う．BitVisor の MiniOS にはあらかじめ管理サーバの IP アドレスが記録されており，管理サーバの各アカウントのホームディレクトリにアクセスして目的

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

のファイルを暗号通信でダウンロードする．

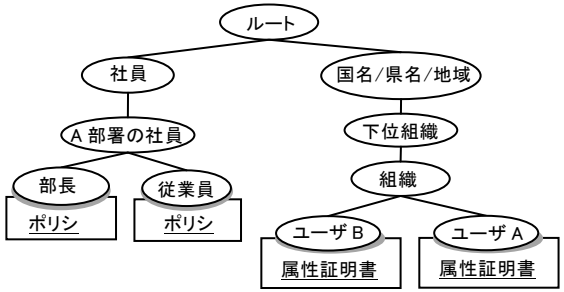

図 8　LDAP で管理するユーザ ID とロールの階層構造

6.4 警告表示機能

ユーザに権限が無く，アクセス制御によってユーザの操作が制限された場合，権限が無い操作を行おうとしたことを警告する．警告にはあらかじめ管理者が定めた任意の文章を表示する．例えば，USB メモリを利用することができないユーザが USB メモリを利用しようとした場合，「あなたは USB メモリの利用権限がありません」といった文字列をディスプレイ上に表示する．

画面表示は図 9 のようにゲスト OS が利用しているメモリ領域に BitVisor がアクセスすることで行う．ビデオチップの制御はゲスト OS 上のデバイスドライバが行うため，BitVisor は PCI コンフィグレーション情報と，ゲスト OS のデバイスドライバが設定しているビデオチップのレジスタ情報から，ゲスト OS 側が画面表示を行うために利用しているメモリ領域を特定することができる．BitVisor はメモリ上に表示させたい内容を画面データ領域に強制的に上書きすることで画面表示を行う．

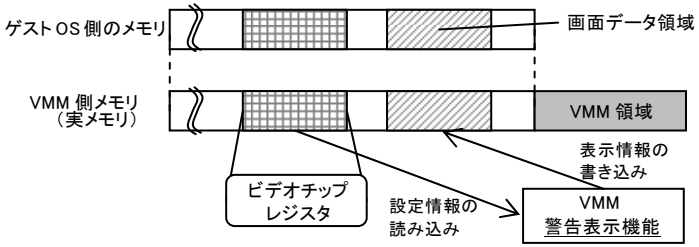

図 9　画面に描画する仕組み

PCI コンフィグレーションおよびレジスタ情報の参照は表示処理毎に行う．警告表示機能によるオーバヘッドを減らすため，レジスタの常時監視は行わない．ゲスト OS が設定している情報を表示の度に確認することで，BitVisor はゲスト OS 側の設定に合わせて表示を行うことができる．BitVisor が PCI コンフィグレーション情報とレジスタ情報からメモリの位置を特定するためには，どのレジスタが何を意味するのかを知っている必要がある．例えば，レジスタから読み取れる画面表示用メモリのオフセット値や現在の解像度を保持するレジスタは画面に警告を表示するために重要である．レジスタの位置と意味はビデオチップごとに特有なものであるため，ビデオチップ毎に対応する BitVisor 用のデバイスドライバを用意する必要がある．

7. 実装

6.1 節で示した CRL の更新からポリシのビットマップ情報への変換までの処理は BitVisor のユーザ認証処理を拡張して実装した．MiniOS は DHCP を利用して IP アドレスを取得し，ネットワークの確立を行った後，CRL の更新をする．管理サーバからのファイルの取得には OpenSSH で提供される scp を Initial RAM disk で動作する Linux システム上でユーザからの直接入力を必要とせず，あらかじめ設定したユーザ ID で動作するように拡張して利用した．IC カード認証が終わると，バックグラウンドで自動的に，LDAP で管理されているユーザもしくはロールで管理サーバにログインし，各ホームディレクトリからコンフィグレーションファイルであるポリシと属性証明書をダウンロードする．ファイルの署名には RSA 暗号アルゴリズムと SHA-1 ハッシュアルゴリズムを利用した．各証明書および，ポリシは利用する前に署名の検証を行う．

ゲスト OS の起動制御はポリシがビットマップ情報に変換され，BitVisor に渡された直後に行い，USB メモリの利用制御は BitVisor で採用されているパラパススルードライバでデータを制御することで実現した．

警告表示機能はチップセットに統合されている Intel GMA3000 を対象に実装を行った．ビデオチップのレジスタ情報を参照することで，画面の解像度が変更された場合でも表示することができるよう実装した．表 1 に実装に用いたソフトウェア，表 2 に実装に用いたハードウェアを示す．

表 1　実装に用いたソフトウェア

ゲスト OS	Windows XP Professional SP2
仮想マシンモニタ	BitVisor 1.0
ソフトウェア	OpenSSH 5.3
管理サーバ用 OS	CentOS 5.3
ソフトウェア	OpenLDAP

表 2　実装に用いたハードウェア

クライアント PC	Intel Core 2 Duo 1.86GHz 2GB RAM チップセット　Intel Q965 ビデオチップ　Intel GMA 3000
IC カード	NTT コミュニケーションズ社 eLWISE TYPE-B
カードリーダ	Axalto Reflex USB v3
サーバ用 PC	Intel Pentium M 1.73GHz 512MB RAM

8. 動作検証

IC カードを設定し，一台のユーザ端末に拡張した BitVisor(提案システム)と Windows

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

XP を導入して処理が適切に行われることを確認した．管理サーバには CentOS を導入して OpenLDAP でアカウント管理を行った．各ホームディレクトリにはコンフィグレーションファイルであるポリシと属性証明書，CRL を格納している．ユーザ端末と管理サーバは同一セグメント内で動作させた．

ゲスト OS の起動制御では，権限が無い場合に VMM の動作を停止し，権限がある場合には目的のゲスト OS（Windows XP）が起動されることを確認できた．USB メモリの利用制御については，権限がある場合には USB メモリを読み書きすることができ，権限が無い場合は USB メモリの読み書きができないことが確認できた．パラパススルードライバ内で制御情報以外のデータのみを破棄するため，ゲスト OS ではフォーマットが認識できないと表示され，USB メモリを利用できない．

アクセス制御で USB メモリの利用が制限された場合の警告メッセージを図 10 に示す．USB メモリをゲスト OS に認識させ，利用しようした場合このように警告が行われることが確認できた．ゲスト OS の起動制御では，テキストモードの画面に警告メッセージが表示されることが確認できた．ゲスト OS から画面データ領域への書き込みの監視を行っていないため，警告メッセージの表示後，ゲスト OS 側が警告メッセージと同じ領域を再描画した場合には警告メッセージが上書きされる．これについては考察で対策を検討する．

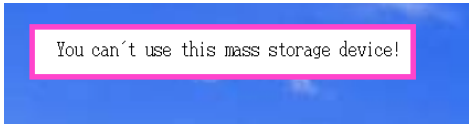

図 10　警告表示

それぞれの処理にかかる時間はタイムスタンプカウンタ[16]を利用して計測した．各処理にかかる時間は表 3 に示したとおりである．ポリシの取得以降の処理はユーザが 1 つのロールに属している場合の計測結果である．複数のロールに属している場合にはロールの数に比例した時間がかかると考えられる．

表 3　処理にかかる時間（単位[秒]）

初期処理	ネットワークの確立	8.18
	CRL の取得	0.495
	ユーザ認証に関わる処理 (PIN コードの入力にかかる時間は除く)	3.30
	属性証明書の取得	0.500
	属性証明書の検証	0.00630
	ポリシ取得	0.452
	ポリシ検証	0.00722
	ポリシをビットマップ情報に変換	0.0236
警告表示	図 10 に示した表示	0.00630

9. 考察

9.1 提案システムの有用性

本研究ではポリシをビットマップ情報に変換して利用した．予めビットマップ情報に変換して利用することで，ファイルという概念を扱えない小型の VMM でロールベースアクセス制御を行うことを可能としている．通常のポリシのようにファイルを扱う場合，アクセス制御のたびにポリシの構造および意味を理解する処理をしなければならない．今回の実装ではその処理が不要であるため，高速な処理と VMM の小型化が可能である．

また，ロールベースアクセス制御を行うことには次のようなメリットがある．ユーザが組織から脱退する，もしくは異動して権限が変更される場合，ユーザ ID ベースポリシを用いると全てのポリシ の該当行に対して処理（更新，削除）をする必要がある．それに対し，ロールベースのポリシを用いた時はユーザとロールの関係のみを処理すればよく，ポリシを変更する必要がない．また，ユーザ ID ベースのポリシに比べてロールベースのポリシは行数を少なくできる．このため，ロールベースのポリシを用いたアクセス制御は VMM を用いたセキュリティシステムの運用において，権限管理コストを削減するために有用だと考えられる．

9.2 拡張性

本研究ではユーザに警告を行う機能として VMM による画面表示機能を実装した．これにより，ユーザに対して管理者は任意の警告メッセージを表示することができる．この機能を利用して権限の無い操作を行ったユーザに対して警告を行うことは，不正行為の抑制につながると考えられる．一方で，VMM の動作の異常を伝えるデバッグ機能として利用することも考えられる．また，キーボードの I/O 捕捉と組み合わせることで，ユーザから VMM に何らかの情報を入力するためのコンソールを実現するなどの応用も考えられる．このような応用を行うためには，オーバヘッドについて考慮しつつ，ゲスト OS から画面データ領域への書き込みの監視を行い，警告メッセージを再描画することで警告メッセージの消去を防ぐなどの改良を行っていく必要がある．VMM による確実な警告表示機能は，ユーザとコンピュータの間の信頼されたパス(Trusted Path)を構築する上で重要な機能であると考えられる．

9.3 セキュリティ

セキュリティ設定情報を管理サーバからダウンロードして利用するシステムへの攻撃として，管理サーバになりすまし，偽装した設定情報によって不正な権限を得ることが考えられる．提案システムでは管理サーバから取得したポリシや属性証明書を利用する前に電子署名による正当性の検証を行っているため，ファイルの偽造を検知して攻撃を防ぐことが可能である．

また，提案システムへの攻撃として，BitVisor，MiniOS（Initial RAM disk と Linux

Vol.2010-DPS-142 No.15
Vol.2010-CSEC-48 No.15
2010/3/4

カーネル）の書き換えが考えられる．書き換えが行われた場合，アクセス制御の仕組みを回避することが可能となってしまう．そのため，実際に組織で運用する場合には，これらのファイルを ROM に書き込んで運用することで書き換えられないよう対策を行うか，TPM[17]などのハードウェア耐タンパモジュールで起動前にファイルや起動シーケンスの完全性の確認を行うなどの対策が必要である．

本稿ではコンフィグレーションファイルや CRL をサーバで管理し，VMM の起動時に自動で取得する仕組みをとった．こうすることで，管理者が設定した内容は VMM の起動時に即座に反映されるため，管理者は CRL などの更新のたびに VMM をユーザ端末に入れ直す必要が無くなった．これにより，BitVisor を利用したセキュリティシステムを運用する際のコストを削減することができる．しかし，VMM の起動時にだけファイルを取得して処理を行うため，VMM の動作中は管理者が行った変更が反映されない．ユーザの失効や権限の変更など，早急に更新を行いたい場合の仕組みについて検討していく必要がある．

VMM のカーネルの脆弱性を突くことなどにより VMM の特権が取得される可能性がある．そのため，今回の実装によって新たな脆弱性が発生しないことを今後確認する必要があると考えられる．

10. まとめ

本研究はセキュリティ機構を組み込んだ VMM である BitVisor の実運用に向け，セキュリティ管理のための運用コストを削減するとともに，情報漏洩対策として効果的なセキュリティ機能を実現することを目的とした．そこで，本研究ではロールに対応するセキュリティコンフィグレーションの自動更新機能，ロールベースアクセス制御の機能，権限のない操作を行ったユーザへの警告表示機能を含むロールベースアクセス制御機構を設計し，BitVisor に実装して動作検証を行った．これにより，セキュリティ管理のための運用コストの削減とユーザの不正行為の抑制が可能になると考えられる．今後は警告表示機能の改良および BitVisor の起動後に権限の変更を行う仕組みの実現に取り組んでいく．

謝辞 この研究の一部は平成 18 年度，科学技術振興調整費課題解決型研究「高セキュリティ機能を実現する次世代 OS 環境の開発」にて支援を受けて実施したものである．BitVisor の開発に携わった多くの方々に感謝を申し上げます．

参考文献

1) Toshihiro, H., Hideki, E. and Kouichi, T.: Task Oriented Management Obviates Your Onus on Linux, in Proc. of the 2004 Linux Conference, (2004).
2) Loscocco, P. and Smalley, S.: Integrating Flexible Support for Security Policies into the Linux Operating System, In Proc. of the FREENIX Track: 2001 USENIX Annual Technical Conference, pp.29-42(2001).
3) Shinagawa, T., Eiraku, H., Tanimoto, K., Omote, K., Hasegawa, S., Horie, T., Hirano, M., Kourai, K., Ohyama, Y., Kawai, Eiji., Kono, K., Chiba, S., Shinjo, Y., Kato, K.: BitVisor: A Thin Hypervisor for Enforcing I/O Device Security, In Proc. of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp.121-135(2009).
4) Meushaw, R. and Simard, D.: NetTop: Commercial Technology in High Assurance Applications, National Security Agency Tech Trend Notes, Vol.9, (2000).
5) 日本ネットワークセキュリティ協会 : 2008 年度情報セキュリティインシデントに関する調査報告書，(2009).
6) Garfinke, T., Pfaff, B. and Chow, J.: Terra: A virtual machine-based platform for trusted computing, In Symposium on Operating Systems Principles(SOSP), Vol.37, pp.193-206(2003).
7) Cox, S.R., Gribble, D.S., Levy, M.H. and Hansen, G.J.: A safety-oriented platform for web applications, In Proc. of the 2006 IEEE Symposium Security and Privacy, pp.350-364(2006).
8) Sailer, R., Jaeger, T., Valdez, E., Caceres, R., Perez, R., Berqer, S., Griffin, L.J. and Doorn, V.L.: Building a MAC-Based Security Architecture for the Xen Open-Source Hypervisor, In Proc. of the Annual Computer Security Application Conference, pp.276–285(2005).
9) Chen, X., Garfinkel, T., Lewis, C.E., Subrahmanyam, P., Waldspurger, A.C., Boneh, D., Dwoskin, J. and Ports, K.R.D.: Overshadow: A virtualization- based approach to retrofitting protection in commodity operating systems, In Proc. of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems, pp.2–13(2008).
10) Seshadri, A., Luk, M., Qu, N. and Perrig, A.: SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSes, In Proc. of the 21st ACM Symposium on Operating Systems Principles, pp.335–350(2007).
11) Murray, G.D., Milos, G. and Hand, S.: Improving Xen Secutiry through Disaggregation, In Proc. of the fourth ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, pp.151-160(2008).
12) Hirano, M., Okuda, T., Kawai, E. and Yamaguchi, S.: Design and Implementation of a Portable ID Management Framework for a Secure Virtual Machine Monitor, Journal of Information Assurance and Security (JIAS), Vol.2, pp.211–216(2007).
13) ITU-T: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks, (2005).
14) Sandhu, R., Ferraiolo, D., Kuhn, R.: The NIST Model for Role-based Access Control:Towards a Unified Standard, In Proc. of the fifth ACM workshop on Role-based Access Control, pp.47-63(2000).
15) Sermersheim, J.: RFC:4511 Lightweight Directory Access Protocol(LDAP) : The Protocol, (2006).
16) Intel Corporation: Intel 64 and IA-32 Architectures Software Developer’s Manual Volume 3A: System Programming Guide, Order Number:253668-033US, (2009).
17) Trusted Computing Group: TPM Specification, Version 1.2.