推薦論文

マルチ OS 環境を利用したアクセス制御システムの実装と性能評価

荒井正人[†] 佐々木慎一[††]
梅都利和[†] 永井康彦[†]

インターネットを利用したサービスが増加する一方で，不正アクセスが深刻な問題となっている．報告者らは，インターネットにおける数多くのセキュリティ脅威の中でも，特にインターネットサーバへの侵入により引き起こされる問題に着目し，たとえ侵入が発生した場合にも，情報資産の保護を可能とするアクセス制御システム（IRACS：Intrusion Resistant Access Control System）を考案した．本報告では，IRACS 自身の保護までも考慮した実装方式と，ファイルアクセスに関する性能テストの結果について示し，WWW をはじめとする各種インターネットサーバに適用可能であることを述べる．

Implementation and Performance Analysis of the Access Control System Using Multi-OS

Masato Arai,[†] Shinichi Sasaki,[††] Toshikazu Umedu[†] and Yasuhiko Nagai[†]

While the number of services on the Internet has increased, illegal accesses are serious issue. We focused on problems that are caused by intrusion into the Internet Servers, and proposed the access control system (IRACS: Intrusion Resistant Access Control System) that can protect informational assets even if the intrusion occurred. In this paper, we describe the implementation of IRACS with consideration to the protection for itself. Moreover, we describe the performance analysis result, and state IRACS is applicable for the Internet Servers such as WWW servers.

1. はじめに

インターネットを利用したサービスが拡大する一方で，不正アクセスが深刻な問題となっている．特に WWW サーバに対するページの改ざん攻撃や，サーバを踏み台にした機密情報の盗聴攻撃が数多く発生している．報告者らは先に，一般的なセキュリティ対策では防ぎきれない不正アクセスから情報を保護するのに有効なアクセス制御システムを提案してきた[1)]．本論文では，本アクセス制御システムを IRACS（Intrusion-Resistant Access Control System）と呼び，IRACS 自身の保護も可能な，マルチ OS 環境を利用した実装方式について述べる．また，WWW サーバへの本システムの適用例と，WWW サーバのレスポンス性能に与える影響について評価した結果を示す．

2. 一般的なセキュリティ対策

インターネットに公開するサーバは，つねに不特定多数のクライアントからの攻撃にさらされていることから，サービスの提供にあたり，十分なセキュリティ対策を施す必要がある．一般的なセキュリティ対策としては，たとえば下記のようなものがあげられる．

(1) **暗号とディジタル署名**
ネットワーク回線を流れるデータの機密性と完全性を保護する．

(2) **サービス利用の制限**
ファイアウォールによりサービスの利用とデータフローの制御を行うとともに，サーバホストにおいても不要なサービスや通信ポートの無効

† 株式会社日立製作所
Hitachi, Ltd.
†† 株式会社日立情報システムズ
Hitachi Information Systems, Ltd.

本論文の内容は 2001 年 7 月のコンピュータセキュリティ研究会にて報告され，CSEC 研究会主査により情報処理学会論文誌への掲載が推薦された論文である．

化を実施する．

(3) **セキュリティ監査・監視**
各種ログを採取し，これらを収集・解析することで不正アクセスを追跡する．

(4) **セキュリティ問題の修正**
たとえば，CERT®/CC [2]等が発信しているウイルス情報やセキュリティホールに関する情報をつねにウォッチしながら，早期対策を実施する．

多くの場合，これらの対策が十分に施されていれば不正アクセスを防止できる．しかし，実際には設定ミスや，セキュリティ問題を放置してしまうこともあり，不正アクセスが発生しやすいのが現状である．特に，サーバプログラムのセキュリティホールを狙われると，サーバホストに侵入されて制御を奪われるケースもある．これらセキュリティホールを修正するためのパッチプログラムが公開されているが，それらを利用しても，当然ながら未公開のセキュリティホールは残されたままとなる．つまり，侵入を未然に防ぐには限界があり，サーバへの侵入を前提としたセキュリティ対策も必要となる．

3. IRACS の不正アクセス対策方針

通常のサーバプログラムは，アクセス制御機能を有する OS 上で動作している．汎用的な OS が採用している任意アクセス制御とは，ファイルの所有者となるユーザの裁量によりファイルの共有範囲を設定し，その範囲内でファイルが利用されるようにするものである．これには，ユーザの悪意や誤操作によって関係者以外へ情報が漏洩するという問題がある．また，トロイの木馬やウイルスに感染したプログラムが，たとえばユーザ A の権限で動作すれば，ユーザ A が利用可能なすべてのファイルが危険にさらされる．特に，OS の管理者権限が侵入者に乗っ取られた場合は，そのサーバは無防備な状態となる．そのような状態は，プログラムのセキュリティホール等が原因となって実際に起こりうる．したがって，任意アクセス制御では侵入を前提としたセキュリティ対策にはならない．

一方，国防システムのように高いセキュリティが要求される分野では，任意アクセス制御と強制アクセス制御の両機能を備えた OS（Trusted OS）が使われることがある．強制アクセス制御とは，各ファイルの機密レベルと各ユーザが扱えるレベルの上限を組織（管理者）があらかじめ決めておき，取扱い資格のない人物へ情報が漏洩しないように，レベルに基づいて情報の流れを制御するものである．また，各プログラムがどの機密レベルで動作しているかも OS が管理しており，原則的には同一レベルのファイルのみアクセス可能のため，トロイの木馬等による被害範囲も限定される．

このように，Trusted OS を用いてサーバを構築すれば，侵入を前提としたセキュリティ対策となる．しかし，Trusted OS ではすべてのファイルに機密レベルを割り当て，すべてのユーザに取扱い資格，つまり機密レベルの上限を割り当てる必要がある．また，プログラムについても実際にはレベルを超えたファイルアクセスを許可しなければ正常に動作しないものが多く，プログラムへの適切な特権の割当て作業も発生することから，運用管理面の負荷が大きいといえる．

そこで IRACS では，次のような不正アクセス対策方針を満たすものとした．

(1) アクセス主体が所有するユーザアカウントだけでなく，そのアクセス手段となるプログラムまで限定するといったアクセス制御ポリシーとし，重要なファイルに対して前記ポリシーを設定する．つまり，アクセス許可されたユーザであっても，正しいアクセス手段でなければすべて禁止する．これにより，不正なプログラムだけでなく，OS の管理者権限を悪用された場合でもファイルアクセスに制限を与えられるようにする．上記ポリシーは，IRACS の管理者が重要なファイルに対して設定するだけでよい．また，上記ポリシーに違反したアクセスの内容とその理由をアクセスログとして記録し，セキュリティ監査・監視にも活用可能とする．

(2) サーバホストに侵入した攻撃者によって，IRACS のセキュリティ機能が容易に無効化されることがないようにする．具体的には，OS の管理者権限を悪用されても，上記アクセス制御ポリシーの改ざんや，IRACS の機能無効化ができないようにする．

4. IRACS の概要と実装方式

本章では，前記 2 つの不正アクセス対策方針を満たす IRACS の機能概要と実装方式について述べる．

4.1 IRACS の概要

IRACS の概要を**図 1** に示す．IRACS は，サーバホスト内部において，実行中の各プログラムとファイルの間に介在するセキュリティ機構である．また，IRACS はポリシー情報とログ情報を管理しており，そのポリシーに基づいた厳密なファイルアクセス制御機能と，アクセスログを記録する機能と，さらには IRACS 自身が提供する機能自体を不正なアクセスから保護す

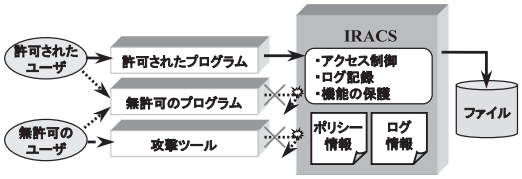

図 1 IRACS の概要
Fig. 1 The overview of IRACS.

ファイル名	タイプ	プログラム名称	ユーザID	特徴値	時間帯
C:¥HTTP-DOCS¥	read	C:¥Serv¥WWWServ.exe	SYSTEM	0x89E3	-
C:¥HTTP-DOCS¥	write	C:¥Prog¥WWWEdit.exe	WWW_Admin	0x3D20	08:00-17:30

図 2 ポリシーファイルの例
Fig. 2 An example of the policy file.

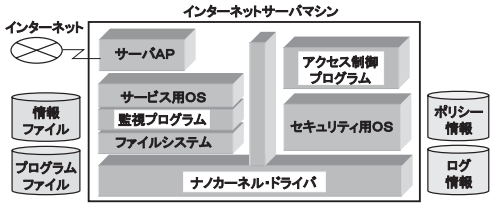

図 3 IRACS の構成
Fig. 3 Composition of IRACS.

る機能を持つ．ここでは，前記不正アクセス対策方針 (1) を満たすポリシー情報とログ情報の内容について説明する．また，4.2 節では前記不正アクセス対策方針 (2) を満たす IRACS の実装方式について説明する．

(1) **ポリシー情報**

各ファイルに対するアクセス制御ポリシーを記述したものである．ポリシー情報にはファイルまたはディレクトリへのアクセス権限を有するアクセス主体を，

- ユーザ名
- プログラム名称
- プログラムファイルが有する特徴値

の組合せで，ファイルの読み出し，書き込み，削除，実行，名称変更といったアクセスタイプごとに指定できるようにした．上記ユーザ名とは，ユーザ ID やグループ ID のことであり，プログラム名称とは，たとえばプログラムファイルのパス名である．特徴値とは，たとえばプログラムファイルのサイズやハッシュ値のことであり，プログラムの真正性検査のために用いる．さらにオプションとして，アクセス可能な時間帯を設定することもできる．このようなポリシー情報により，特定のユーザが，改ざんされていない特定のプログラムを用いた場合に限り，ファイルアクセスを許可するといった厳密なファイルアクセス制御が可能となる．ポリシー情報の記述例を**図 2** に示す．

(2) **ログ情報**

IRACS はファイルアクセスに関する次のような情報をアクセスログとして記録する．

- アクセス日時
- アクセス主体に関する情報
- アクセス対象に関する情報
- ポリシー違反の理由

4.2 IRACS の実装方式

IRACS は，侵入という行為を防ぐものではなく，侵入時の被害，特に情報の機密性と完全性の喪失を極小化することを目的としている．したがって，3 章の不正アクセス対策方針 (2) で述べたように，仮にサーバホストが侵入された場合にも IRACS が確実に機能すること，つまり下記要件を満たすことが要求される．

要件 1：機能のバイパスを防止できること．
要件 2：機能の改ざんを防止できること．
要件 3：機能の非活性化を防止できること．

これらの要件は，セキュリティ評価基準である ISO/IEC15408 [8] のセキュリティ要件としても規定されており，セキュリティ機能自体を保護するうえで重要なものである．これら要件を満たすために，次の 3 つの方針に基づいて IRACS の実装方式を検討した．

- **方針 1**：機能のバイパス防止
  アクセス制御機能を回避したファイルアクセスが発生しないように，各プログラムによるすべてのファイルアクセスを検出可能なレイヤで監視する．
- **方針 2**：機能の改ざん防止
  アクセス制御機能を不当な干渉から保護するために，攻撃対象となりやすい領域から分離された安全な領域でアクセス制御機能を実装する．
- **方針 3**：機能の非活性化防止
  アクセス制御機能が常時有効となるように，サーバホストの稼働中は，IRACS の機能停止を禁止する．

その結果，IRACS を**図 3** に示すような構成とした．IRACS の各構成要素について以下述べる．

(1) **ナノカーネルド ライバ**

ナノカーネル方式 [3),4)] を用いて 1 台のサーバ上に 2 つの OS を共存させ，かつ OS ごとにデバイスとメモリ空間を分割・専有可能にするためのソフトウェアである．IRACS では，一方をサービス用 OS として利用し，もう一方をセキュリティ用 OS として利用する．ここで，サービス用 OS 側は，インターネットに対してサービスを提供する役割を持つことから，つねに攻撃を受けやすい領域であるといえる．IRACS で

は，サービス用 OS から独立したセキュリティ用 OS 側に，アクセス制御プログラムやポリシー情報，ログ情報を配置することで，機能の改ざん防止を実現している．また，ナノカーネルドライバは，監視プログラムとアクセス制御プログラムの間で利用する OS 間メッセージ通信機能を提供する．本機能をサービス用 OS 側から利用できるのは，監視プログラムのみとすることで，機能の改ざん防止をより確実なものとしている．さらにナノカーネルドライバは，サーバホストにおけるどのプログラムよりも先に起動され，サービス用 OS の稼働中には終了できないように実装することで，機能の非活性化防止も実現している．

(2) **監視プログラム**

監視プログラムは，ファイルオープン要求を検出すると，その要求元，つまりアクセス主体となるプログラムの情報（プログラム名称，特徴値，ユーザ ID）を取得する．取得したプログラム情報と，アクセス対象のファイル名，およびアクセスタイプをアクセス制御プログラムに通知し，認められたアクセスであるか否かの判定結果を受信する．判定結果に応じて，ポリシーで認められたアクセスであればファイルシステムにアクセス続行を要求し，ポリシー違反であればアクセス要求元のプログラムにエラーを返す．

監視プログラムは，すべてのファイルアクセスを検出できるよう，サービス用 OS 側のファイルシステムの近傍（カーネルレベル）で監視する．具体的には，ファイルシステムと同様に，サービス用 OS とともに起動・終了するドライバソフトとして実装する．これにより，監視プログラムはどのアプリケーションプログラムよりも先に起動され，サービス用 OS の稼働中には終了されないものとなる．いい換えれば，アプリケーションの実行中はつねに監視状態となり，機能のバイパス防止を実現できる．さらに，監視プログラムを構成するファイルも，不当に改ざん・削除されないようポリシー情報を設定し，IRACS 自身により保護しておく．

(3) **アクセス制御プログラム**

監視プログラムからの通知を受け，ポリシー情報の記述内容に基づきファイルアクセスの正当性を検査し，その結果を監視プログラムに返信する．ポリシー違反のアクセス要求であると判定した場合には，ログ情報への記録もあわせて行う．アクセス制御プログラムは，ポリシー情報やログ情報とともにセキュリティ用 OS 側に配置しているため，サービス用 OS 側から不正な干渉を受けることはない．

4.3 高度な攻撃への対策

IRACS のアクセス制御ポリシーによれば，仮に OS の管理者権限を悪用されても，アクセス手段がポリシーに違反すれば不正アクセスとして禁止できる．また，プログラムファイルの改ざんも検知できるため，トロイの木馬や，ウイルスに感染したプログラムからもファイルを保護できる．ただし，プログラムファイルを書き換えることなく不正な機能を追加するような攻撃があるとすれば，IRACS では防止できない．そのような機能拡張が容易なプログラムは，攻撃者から悪用される危険性が高いため，重要なファイルへのアクセス手段として用いないことを推奨する．あるいは，そのようなプログラムを通常は CD-ROM 等の可搬記憶媒体に格納しておき，必要なときにマウントして利用すれば，より高いセキュリティが実現できると考える．

また，4.2 節で述べた実装方式により，アプリケーションレベルの不正なプログラムを用いたアクセス制御機能のバイパス，改ざん，非活性化を防止できるが，それを上回る非常に高度な攻撃として，以下に示すものが考えられる．

(1) **機能のバイパス**：監視プログラムが検出できない特殊なファイルアクセス経路を確立し，不正にファイルをアクセスする．

(2) **機能の改ざん**：OS 間メッセージ通信の妨害や，偽のメッセージへのすり替えにより，不正なファイルアクセスが許可されるようにする．

(3) **機能の非活性化**：IRACS を構成するプログラム（ドライバソフト）を削除して，次回のシステム起動時に IRACS が起動しないようにする．

これらはすべてドライバソフトの不正な導入または削除によるものであることから，対策としてはドライバソフトの新規導入と削除を監視または制限する手段を別途設けることが有効といえる．

5. IRACS のログイン方式

IRACS では，セキュリティ用 OS 側に格納されるポリシー情報の設定や，ログ情報の参照といった管理作業を，OS の管理者権限ではなく，IRACS の管理者権限を用いて行う．IRACS の管理者権限取得には，セキュリティ用 OS 側のエージェントプログラムによ

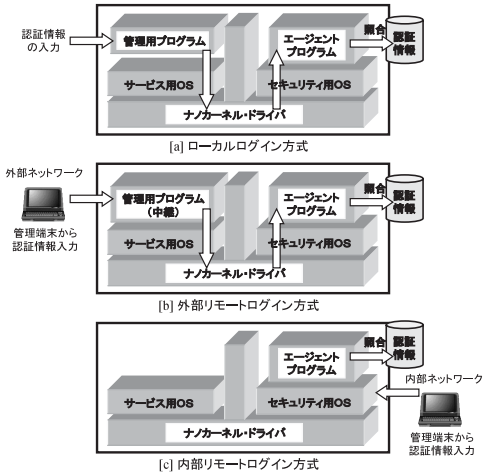

図 4 IRACS の管理者ログイン方式
Fig. 4 The administrator log in method of IRACS.

る認証が必要である．これにより，OS の管理者権限が悪用されても，ポリシー情報やログ情報の削除を防ぐことができる．ここでは，エージェントプログラムによる 3 種類のログイン方式を提案する．

(1) **ローカルログイン方式**（**図 4** の [a]）
サービス用 OS 側の管理用プログラムから OS 間通信機能を利用してログインする．セキュリティ用 OS 側には，IRACS の管理者認証情報（たとえばパスワード）を登録しておき，管理者が入力した認証情報と照合して認証する．

(2) **外部リモートログイン方式**（図 4 の [b]）
リモートの管理端末から，サービス用 OS 側が利用する外部ネットワークと，前記管理用プログラムと，OS 間通信機能を通じてログインする．エージェントプログラムによる認証処理は前記ローカルログイン方式と同様である．

(3) **内部リモートログイン方式**（図 4 の [c]）
セキュリティ用 OS 側で占有可能なネットワークインタフェースを装備し，リモートの管理端末から内部ネットワークを通じてログインする．ただし，セキュリティ用 OS が利用する内部ネットワークは，サービス用 OS が利用する外部ネットワークと物理的または論理的に切り離しておく．

前記**ローカルログイン方式**では，管理対象のサーバマシンを直接操作することになるが，管理対象が少数であれば問題ない．また，サービス用 OS 側から IRACS の管理者ログインが可能なため，内部リモートログイン方式に比べてセキュリティ面で劣る．したがって，推測し難いパスワードの設定や，ID カードとパスワードの組合せ，あるいは生体情報を利用した認証技術等の採用によりセキュリティを高める必要がある．

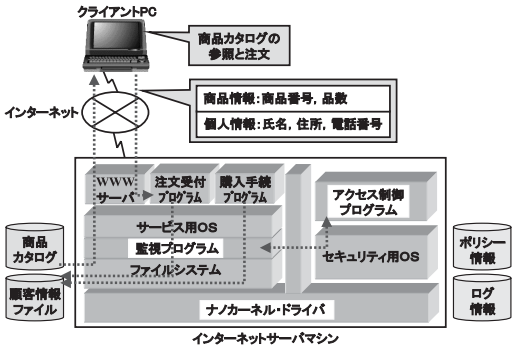

図 5 通信販売システムへの適用例
Fig. 5 An application example for online shopping system.

外部リモートログイン方式は，既存のネットワーク環境を用いて集中管理が可能であるという利点があるが，セキュリティ面では 3 つの方式の中で最も劣るため，たとえば IP アドレスによる管理端末の認証や，通信データの暗号化等，ほかのセキュリティ対策と組み合わせる必要がある．

内部リモートログイン方式は，サービス用 OS を介さずにログイン可能であるためセキュリティ面で優れている．また，管理端末から複数のサーバを集中管理可能であるといった利点もあるが，セキュリティ用 OS 側にもネットワークインタフェースを装備し，管理用のネットワークまで整備する必要があり，導入コストは比較的大きい．以上述べた各方式の特徴を考慮したうえで，システムの運用形態に適したログイン方式を採用することが望ましい．なお，今回実装したシステムでは，パスワードを用いたローカルログイン方式を採用した．

6. WWW への適用例

通信販売システムを構成する WWW サーバの要塞化を例にとり，アクセス制御ポリシー設定の考え方について説明する．なお，要塞化とは弱点となる穴を塞いで侵入に備えることであり，たとえば不要なサービスの停止や，セキュリティホールの修正も要塞化につながる．

6.1 通信販売システムの概要

通信販売システムの概要を**図 5** に示す．通信販売システムの利用者は，自宅の PC 等を用いてインターネット経由で商品の購入手続きができる．インターネットサーバマシンにおいて，サービス用 OS 側のハード

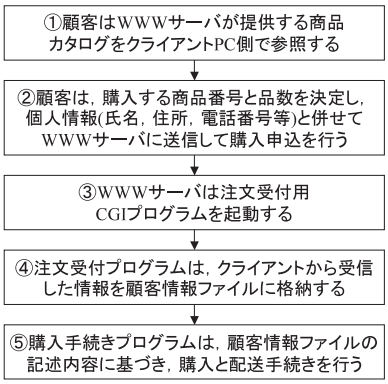

図 6 商品購入フロー
Fig. 6 The flow chart of commodities purchase.

ディスクには，保護対象である商品カタログ情報や，顧客情報が格納されている．WWW サーバは，クライアント PC からの要求に応じて，商品カタログを開示したり，所定の CGI プログラムを起動したりする役割を持つ．注文受付プログラムは，WWW サーバにより起動される CGI プログラムであり，顧客からの注文情報を顧客情報ファイルに格納する役割を持つ．購入手続きプログラムは，正式な商品購入や配送に必要な手続きをするために，顧客情報ファイルを参照しながら所定の処理を実行する役割を持つ．

このような通信販売システムを利用した商品購入の大まかな流れを**図 6** に示す．

6.2 IRACS によるファイル保護

次に，IRACS による各ファイルの完全性・機密性確保について述べる．まず，商品カタログ等のファイルについては，改ざんされないよう保護する必要がある．また，顧客情報ファイルについては，改ざん防止だけでなく，機密性の確保も必要である．

安全な通信販売サービスを提供するためには，上記ファイルに対して必要最小限の正常アクセスのみを許可するようにアクセス制御ポリシーを設定すればよい．通信販売システムの WWW サーバにおける正常なファイルアクセスとは次のとおりである．

(1) 商品カタログファイルへの正常アクセス
- 商品カタログの公開：WWW サーバによる商品カタログファイルの読み出しができればよい．
- 商品カタログの編集：WWW サイトの管理者が，所定の編集用ツールを就業時間内に利用した場合に読み出しと書き込みができればよい．

(2) 顧客情報ファイルへの正常アクセス
- 顧客情報の登録：受信した注文情報を，所定の注文受付プログラムのみが追記できればよい．
- 顧客情報の参照：WWW サイトの管理者が所定の購入手続きプログラムを就業時間内に利用した場合に参照（読み出し）できればよい．

表 1 WWW ページのダウンロード時間
Table 1 Download time of WWW pages.

	高速 5 サイト	低速 5 サイト	40 サイト
ページを構成するファイル数の平均 [個]	13	39	20
ページダウンロードの平均時間 [msec]	1,130	5,790	2,640

このように，IRACS を WWW サーバに適用すれば，攻撃者が WWW サーバに対して攻撃を行った場合でも，IRACS がファイルアクセスの正当性をポリシー情報に基づいて厳密に検査することから，商品カタログの改ざんや，顧客情報の漏洩・改ざんといった脅威の発生可能性を抑えることができる．さらに，顧客情報のような重要ファイルへのアクセスを特別に許可されたプログラムについては，たとえば可搬記憶媒体に格納しておき，必要なときにのみマウントして利用するといった運用上の工夫と組み合わせることでもプログラムの不正利用を防止でき，より高いセキュリティを実現できる．

7. IRACS の性能評価

IRACS は，インターネットサーバにおける不正アクセス対策を目的として開発したものである．インターネットサーバは短時間に集中的なアクセスが行われることがあるため，IRACS のアクセス制御機能のオーバヘッドがインターネットサーバの性能に大きな影響を与えないことが重要である．ここでは，WWW サーバへの適用を想定した性能評価結果を示す．

7.1 IRACS の性能目標

まず，WWW サーバの一般的な性能を把握するために，Mills ら[5]が報告した，著名な 40 サイトの WWW ページダウンロード時間を**表 1** に示す．これによると，WWW サイトからページをダウンロードするのに要する時間は，高速 5 サイト平均で約 1.1 秒，低速 5 サイト平均で約 5.8 秒，40 サイト全体の平均で約 2.6 秒である．

米国の調査会社 Zona Research のレポート[6]によれば，14.4 kbps モデムで接続しているユーザがページのダウンロードに我慢できる時間は 8 秒で，それを超えると大部分のユーザは商品の購入をあきらめる傾向があると報告されており，これがいわゆる「8 秒ルール」という言葉の根拠となっている．つまり，EC サイトの運営にとっては，8 秒ルールに違反しないレス

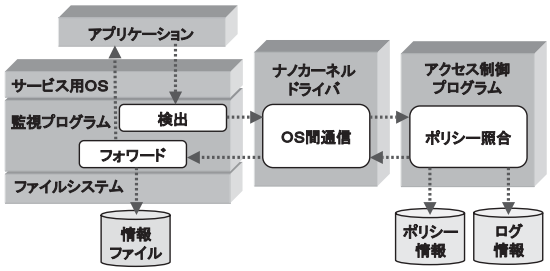

図 7 IRACS におけるアクセス制御処理のフロー
Fig. 7 The flow of the access control process on IRACS.

ポンス性能を達成することが重要となる．最近では，ADSL や CATV インターネットによる高速回線が普及してきているため，今後は許容される待ち時間が縮小することも予想できる．一般的な情報処理システムについていえば，比較的簡単な情報検索のような業務の場合，許容されるレスポンス時間は 2 秒以下であるといわれている[7]．これはネットワーク遅延等を含まないスタンドアロンの環境を想定した時間であるが，レスポンス性能が最重要視されるようなサービスであれば，たとえページのダウンロードであっても，いずれはこの「2 秒ルール」が要求されるものと考える．

ここで再び表 1 を見ると，低速 5 サイト平均であっても上記「8 秒ルール」に違反していないことが分かる．また，高速 5 サイト平均となると，より厳しい「2 秒ルール」を満たしていることが分かる．

そこで，IRACS をそれらサイトの WWW サーバに搭載した場合にも，低速 5 サイト平均のレスポンス時間は 8 秒以下，高速 5 サイト平均においては 2 秒以下になることを，IRACS の性能目標とする．

7.2 IRACS の性能測定

次に，IRACS の性能について検証する．IRACS によるアクセス制御処理は，**図 7** に示すように監視プログラムによる「検出処理」,「フォワード処理」のほかは，ナノカーネルドライバによる「OS 間通信処理」とアクセス制御プログラムによる「ポリシー照合処理」からなる．前記「OS 間通信処理」と「ポリシー照合処理」それぞれの所要時間を計測するために，下記 3 つのタイプのサーバを用意した．なお,「検出処理」と「フォワード処理」は，便宜上「ポリシー照合処理」の一部と見なす．

(1) **標準サーバ**
IRACS を搭載せずに，通常のファイルアクセス処理に要する時間を計測する．測定用のサーバは，Pentium®Ⅲ☆ 600 MHz，256 MB RAM，20 GB HDD のマシンに Microsoft® Windows® 2000☆☆ Service Pack 2 をインストールしたものを使用する．

(2) **マルチ OS 版 IRACS 搭載サーバ**
本報告の IRACS をサーバに搭載し，IRACS のアクセス制御処理によるオーバヘッドを計測する．測定用のサーバは，上記と同等のマシンに，サービス用 OS として Microsoft® Windows® 2000 Service Pack 2 をインストールし，さらにセキュリティ用 OS として Linux version 2.2.17☆☆☆を搭載したナノカーネルドライバをインストールしたものを使用する．また，監視プログラムをサービス用 OS のドライバソフトとして実装し，アクセス制御プログラムをセキュリティ用 OS のアプリケーションプログラムとして実装した．

(3) **シングル OS 版 IRACS 搭載サーバ**
シングル OS 版 IRACS は，OS 間通信処理時間を計測することを目的に用意したものであり，ナノカーネルドライバを使わずに，前記標準サーバと同等のマシンとソフトウェア構成に対して，監視プログラムとアクセス制御プログラムをともに Microsoft® Windows® 2000 のドライバソフトとして実装している．

表 2 ファイル読み出し平均時間
Table 2 The average time of the file reading.

	WWW サーバの最大同時接続数			
	1	10	20	30
標準サーバ	0.154	1.731	3.560	6.157
シングル OS 版	0.430	4.472	9.339	14.313
マルチ OS 版	2.953	29.591	56.546	84.729

ファイル読み出し実測時間：単位 [msec]

上記 (1)〜(3) のそれぞれのタイプについて，サーバ負荷を変更しながらファイル読み出しに要する時間を実測した結果を**表 2** に示す．本結果は，WWW サーバプログラムによるファイル読み出し，つまりファイルのオープンからリード，クローズまで一連の処理を複数回行い，その所用時間の平均をとったものである．なお，上記ファイルのサイズは，報告者らが過去ダウンロードした WWW コンテンツの平均ファ

☆ Pentium は，米国およびその他の国における，Intel Corporation またはその子会社の商標または登録商標です．
☆☆ Microsoft および Windows 2000 は，米国 Microsoft Corporation の，米国およびその他の国における登録商標または商標です．
☆☆☆ Linux は，Linus Torvalds 氏の米国およびその他の国における登録商標あるいは商標です．

表 3 WWW ページダウンロード時間の予測値
Table 3 The expected time for WWW pages download.

	WWW サーバの同時接続数			
	1	10	20	30
高速 5 サイト	1,166 （103）	1,492 （132）	1,819 （161）	2,151 （190）
低速 5 サイト	5,899 （101）	6,877 （119）	7,856 （136）	8,854 （153）
40 サイト平均	2,695 （102）	3,197 （121）	3,700 （140）	4,211 （160）

単位 [msec] 括弧内の数値は低下率%

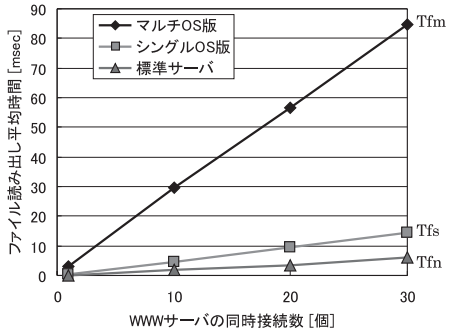

図 8 ファイル読み出しの平均時間
Fig. 8 The average time of the file reading.

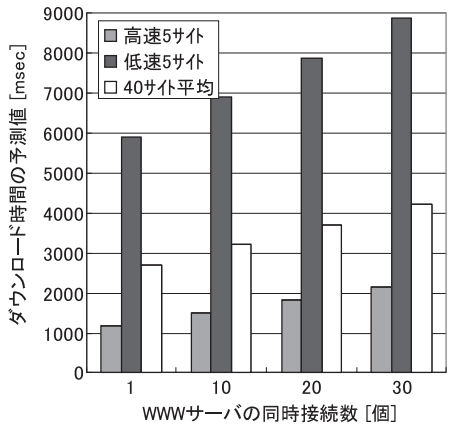

図 9 WWW ページダウンロード時間の予測値
Fig. 9 The expected time for WWW page download.

イルサイズから 7 k バイトとした．また，上記サーバ負荷は WWW サーバの最大同時接続数の設定により変更し，設定した数だけクライアントと接続した状態でサーバ側でのファイル読み出し時間を測定した．

ここで，マルチ OS 版 IRACS 適用時のファイル読み出し時間を Tfm，シングル OS 版 IRACS 適用時のファイル読み出し時間を Tfs，標準サーバ使用時のファイル読み出し時間を Tfn とした場合,「OS 間通信処理」の時間 Tcom と,「ポリシー照合処理」に要する時間 Tpc は，それぞれ式 (1) と (2) により求まる．

$$Tcom = Tfm - Tfs \quad (1)$$

$$Tpc = Tfs - Tfn \quad (2)$$

図 8 は，上記 Tcom と Tpc が，サーバ負荷によりどのように変化するかを分かりやすくするために，表 2 に示した測定値をグラフ化したものである．このグラフから，Tcom および Tpc ともサーバ負荷に比例して大きくなり，特に Tcom の増加が大きいことが分かる．

次に，上記 (2) マルチ OS 版 IRACS 搭載サーバを WWW サーバへ適用したことを想定して，WWW ページダウンロード時間の予測値を求め，どの程度のサーバ負荷までならば，前記 2 秒ルールや 8 秒ルールを満たせるかを示す．ここで，標準サーバ使用時の WWW ページダウンロード時間を Tdn，WWW ページを構成するファイルの数を Cn とした場合，マルチ OS 版 IRACS 適用時の WWW ページダウンロード予測値（時間）Tdm は，式 (3) により求まる．

$$Tdm = Tdn + (Tfm - Tfn)Cn \quad (3)$$

上記 Tdn と Cn に，それぞれ表 1 に示したダウンロード時間とファイル数を用いて算出した値とグラフをそれぞれ**表 3** と**図 9** に示す．なお，表 3 における括弧内の数字は，標準サーバ使用時を 100%とした場合のレスポンスの低下率を表す．

表 3 に示す予測値によると，マルチ OS 版 IRACS を WWW サイトに適用したときのダウンロード予測時間は，低速 5 サイト，高速 5 サイトともにサーバ負荷，つまり WWW サーバの同時接続数が 20 以下であれば，それぞれ 8 秒以下と 2 秒以下という目標値をクリアしている．

8. ま と め

不正なプログラムからのアクセスのみでなく，OS の管理者権限を悪用したアクセスからもファイルを保護可能なアクセス制御システム IRACS を提案し，その実装方式と IRACS の管理者ログイン方式，さらには WWW サーバへの適用例と性能評価結果について述べた．IRACS は，汎用的な OS が採用する任意アクセス制御では防止できない不正なファイルアクセスを，強制アクセス制御よりも簡易な設定により防止す

るものである．また，IRACS はファイルアクセス制御機能を提供するだけでなく，その機能自体をバイパス・改ざん・非活性化から保護することもできる．したがって，IRACS を搭載したインターネットサーバであれば，侵入された場合でも IRACS が確実にアクセス制御機能を提供することができ，ファイルの完全性と機密性の確保が期待できる．

一方，性能評価結果によると，IRACS 適用による性能の劣化は，クライアントとの同時接続数が 20 程度のサーバ負荷までならば，WWW サイトのレスポンス性能にとって問題ない程度であることが分かった．しかし，非常に高いレスポンス性能が求められる WWW サイトへ適用するのであれば，OS 間通信処理を重点に性能向上策を検討する必要がある．

参 考 文 献

1) 荒井ほか：マルチ OS 環境を利用したアクセス制御システム，第 61 回情報処理学会全国大会論文集（1），pp.45–46 (2000).
2) CERT® Coordination Center. http://www.cert.org/
3) 新井ほか：ナノカーネル方式による異種 OS 共存技術「DARMA」の提案，第 59 回情報処理学会全国大会論文集（1），pp.139–140 (1999).
4) 佐藤ほか：ナノカーネル方式による異種 OS 共存技術「DARMA」の実装，第 59 回情報処理学会全国大会論文集（1），pp.141–142 (1999).
5) Mills, P. and Loosley, C.: A Performance Analysis of 40 e-Business Web site, *CMG Journal of Computer Resource Management*, Issue 102 (2001).
6) Zona Research: The Need for Speed, *Industry Update Report*, Issue 32 (June 1999).
7) ヒューマンインタフェース調査研究会（編）：ヒューマンインターフェース，pp.48–49, 財団法人 日本データ通信協会 (1990).
8) ISO/IEC 15408-2: Information Technology — Security Techniques — Evaluation Criteria for IT security, Part 2: Security functional requirements.

(平成 14 年 6 月 18 日受付)
(平成 15 年 2 月 4 日採録)

推 薦 文

ナノカーネル方式をセキュリティ確保に応用し，WWW サーバへの適用を想定して実装および実測を行い，従来方式と比較して定量的に評価した点が推薦に値する． （CSEC 研究会主査 岡本 栄司）

荒井 正人（正会員）

1990 年日本大学理工学部電子工学科卒業．1992 年同大学院理工学研究科修士課程修了．同年（株）日立製作所入社．マイクロエレクトロニクス機器開発研究所を経て，システム開発研究所横浜ラボラトリに勤務．現在，同研究所セキュリティシステム研究部にて情報システムのアクセス制御技術，セキュリティ評価技術の研究開発に従事．

佐々木慎一

1997 年東京理科大学理工学部情報科学科卒業．同年（株）日立情報ネットワーク（現日立情報システムズ）入社.（株）日立製作所システム開発研究所でのアクセス制御技術に関する研究開発を経て，現在（株）日立情報システムズセキュリティソリューション部に勤務．情報システムのセキュリティ診断に関する業務に従事．

梅都 利和

1982 年東京理科大学工学部経営工学科卒業．同年（株）日立製作所入社．情報機器事業部にてソフトウェア開発に従事．現在，同事業部ソフトウェア開発センターの主任技師として，主に金融機関システムの端末装置に関する基本ソフトウェア，ミドルソフトウェアの開発およびプロジェクト管理を担当．最近では，同事業部 ISO/IEC15408 認証取得プロジェクトの責任者として，情報セキュリティに関するソフトウェア製品の認証を取得．

永井 康彦（正会員）

1983 年日本大学理工学部航空宇宙工学科卒業．1985 年同大学院理工学研究科修士課程修了．同年（株）日立製作所入社．システム開発研究所横浜ラボラトリ勤務．情報セキュリティ，ネットワーク管理システム，グループウェア等の研究開発に従事．現在同研究所セキュリティシステム研究部主任研究員．電気学会，電子情報通信学会，日本航空宇宙学会各会員．