平成16年度内閣官房情報セキュリティ
対策推進室委託調査

電子政府におけるセキュリティに配慮したOSを活用した情報システム等に関する調査研究

報告書

みずほ情報総研株式会社

目　次

はじめに

近年、我が国においてはインターネットの普及が進み、一般利用者の裾野が急拡大するとともに、いわゆるブロードバンド型の常時接続回線の利用などその利用形態の面でも高度化が進展しつつある。また、エネルギー供給、交通、政府･行政サービス等の国民の社会経済活動に不可欠なサービス提供や公共の安全確保等において、情報システムがますます重要な役割を果たすようになってきているが、このことは同時に、これら社会基盤の多くが情報システムへの依存性を一層高めつつあることをも意味している。

このような状況の中、コンピュータウイルスやサイバー攻撃等の情報セキュリティ関係事案の発生、過失や内部犯行による情報漏洩やデータの消失、各種システム内の不具合・事故などにより、電子政府や電子自治体、重要インフラ等の社会基盤を支える情報通信ネットワークや情報システムの機能不全や信用低下などが引き起こされ、我が国社会に混乱を生じさせる危険性がますます高まってきている。

これら危険性を低下させるべく、政府は e-Japan 重点計画の策定を行い、情報セキュリティ関連施策の実行を図るなど、様々な取り組みの推進により着実に情報セキュリティ対策向上に努めてきたが、今後とも、継続的かつ着実に実施していくことが重要であり、電子政府や電子自治体、重要インフラ等の公共的分野における体制整備や情報システムの評価・検証と改善、運用管理の適切な実施、広く一般への普及啓発、研究開発や人材育成の推進などを引き続き図っていくことが必要である。

このように情報システムの安全性を確保するためには、様々な観点から対策に取り組む必要があるが、ひとつの方策として、情報システムの要となるコンピュータシステムに関して、その中核となるオペレーティングシステム（OS）のセキュリティ機能について検討をすることは有益であると考えられる。

今般、内閣官房情報セキュリティ対策推進室は、みずほ情報総研株式会社に「電子政府におけるセキュリティに配慮した OS を活用した情報システム等に関する調査研究」の実施を依頼した。これを受け、平成 16 年 11 月よりみずほ情報総研株式会社において専門家と政府職員から構成される検討会を開催するなどして、セキュア OS 等について精力的な検討を行ってきた。

検討会においては、セキュア OS の特徴について整理を行い、いくつかのシステムにおけるセキュア OS の導入イメージについてまとめ、今後の活用にあたっての課題について委員の方々より活発なご議論、ご意見をいただいた。本報告書は、その検討結果の集大成である。

本報告書の構成は次の様である。第 1 章において読者にセキュア OS の大まかなイメージを持っていただけるよう歴史的な経緯を述べつつ、いわゆるセキュア OS と呼ばれるものについて概説し、第 2 章においてその導入の一般的な利点と問題点について解説した。さらに、第 3 章にお

いて、いくつかの情報システムに導入した場合の適用可能性について検討するとともに、第４章において複数のセキュア OS 導入モデルについて、そのメリットや課題を検討した。これらの検討結果等を踏まえ第５章において、今後の課題・展望について記述した。

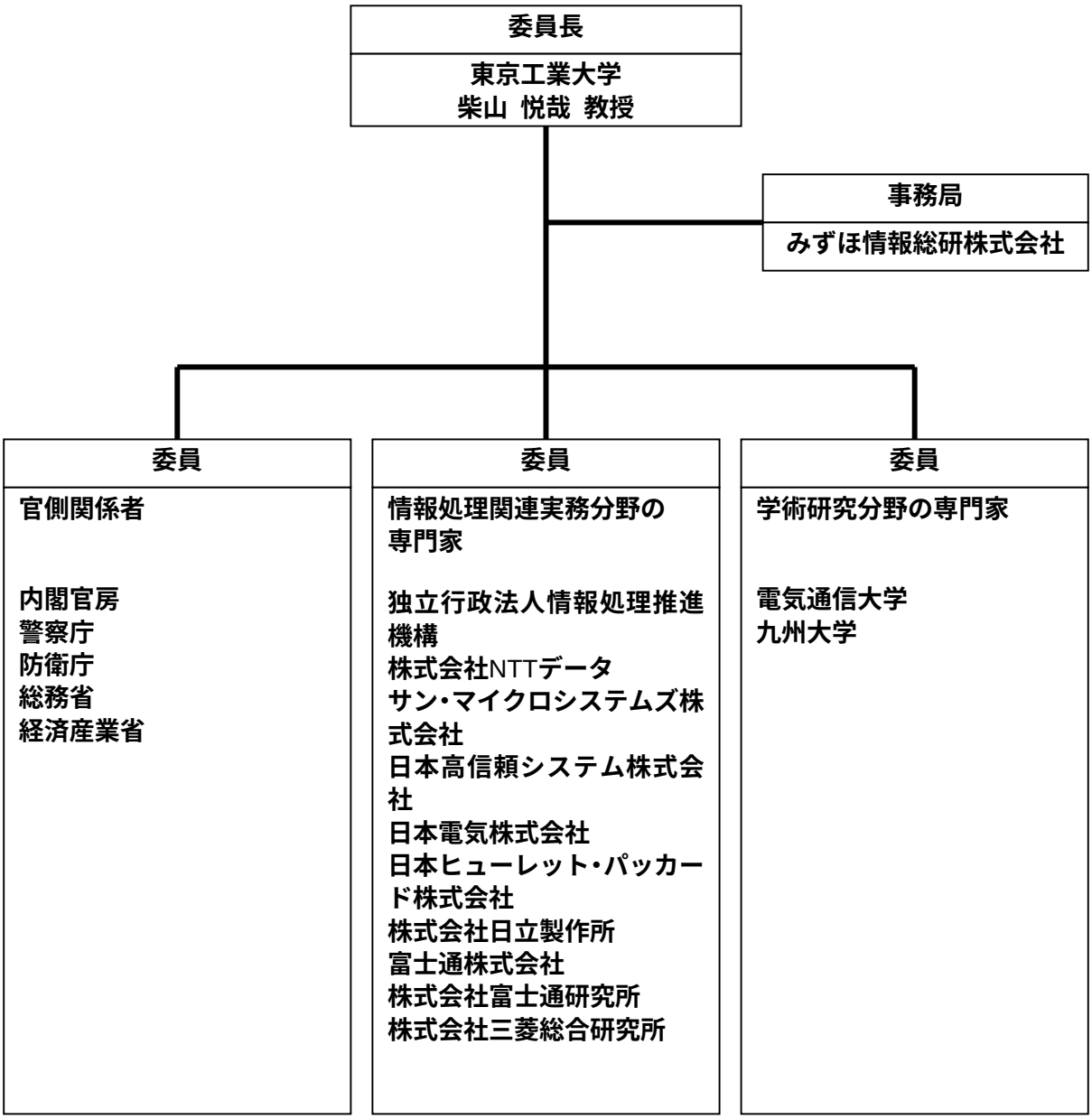
また、報告書の作成にあたっては、必ずしもこの分野において専門家でない方々にも、検討結果をご理解いただけるよう平易な用語を用い、読み手に分かりやすくすることに心がけたつもりである。従って、技術的には必ずしも正確でない表現があることをご理解いただきたい。

本調査研究報告書が、今後、電子政府等の情報セキュリティレベルの一層の向上への一助となれば幸いである。

検討委員会委員長　柴山 悦哉

実施体制

本調査研究は下記に示す各専門家から構成される検討委員会を編成の上実施し、その検討結果をもとに報告書を取り纏めたものである。



検討委員会委員等名簿

○民間委員（敬称略）

	氏 名	所 属 機 関 等
委員長	柴山 悦哉	東京工業大学 大学院 情報理工学研究科 教授
委員	浅原 健	株式会社三菱総合研究所 科学技術研究本部 戦略技術研究部 研究部長
委員	女部田 武史	情報処理推進機構 セキュリティセンター 情報セキュリティ技術ラボラトリー
委員	蒲田 順	株式会社富士通研究所 ITコア研究所 セキュアコンピューティング研究部
委員	木下 俊之	株式会社日立製作所 システム開発研究所 主管研究員
委員	河野 健二	電気通信大学 情報工学科 講師
委員	櫻庭 健年	株式会社日立製作所 システム開発研究所 主任研究員
委員	佐藤 慶浩	日本ヒューレット・パッカード株式会社 個人情報保護対策室室長
委員	澤田 栄浩	日本高信頼システム株式会社 代表取締役社長
委員	田端 利宏	九州大学 大学院システム情報科学研究院 情報工学部門 助手
委員	寺澤 慎祐	サン・マイクロシステムズ株式会社 営業統括本部 e-Japan 営業開発本部 専任部長
委員	原田 季栄	株式会社 NTT データ オープンソース開発センタ 技術開発担当 シニアスペシャリスト
委員	宮川 寧夫	情報処理推進機構 セキュリティセンター 情報セキュリティ技術ラボラトリー 主任研究員
委員	宮田 義文	富士通株式会社 プロダクトビジネス企画本部 事業開発室
委員	依田 透	日本電気株式会社 e-ガバメントソリューション推進本部 システムマネージャー

○官側委員（敬称略）

	氏 名	所 属 機 関 等
委員	立石 譲二	内閣官房 情報セキュリティ対策推進室 内閣参事官
委員	青木 信義	内閣官房 情報セキュリティ対策推進室 内閣参事官
委員	山田 浩一	内閣官房 情報セキュリティ対策推進室 内閣事務官
委員	高村 信	内閣官房 情報セキュリティ対策推進室 内閣事務官
委員	田辺 雄史	内閣官房 情報セキュリティ対策推進室 内閣事務官
委員	今井 俊博	内閣官房 情報セキュリティ対策推進室 内閣事務官
委員	和田 敏一	警察庁 情報通信局 情報管理課 課長補佐
委員	金剛 章	警察庁 情報通信局 情報管理課 専門官
委員	亀田 健一	防衛庁 技術研究本部 技術部 技術情報管理課 情報ネットワーク管理室 情報ネットワーク管理係
委員	小川 浩史	防衛庁 長官官房 情報通信課 部員
委員	工藤 篤	総務省 情報通信政策局 情報セキュリティ対策室 推進係長
委員	風間 博之	経済産業省 商務情報政策局 情報政策ユニット 情報処理振興課 係長
委員	上原 智	経済産業省 商務情報政策局 情報政策ユニット 情報処理振興課 係長
委員	川口 修二	経済産業省 商務情報政策局 情報政策ユニット 情報セキュリティ政策室 課長補佐

○検討委員会事務局

	氏 名	所 属 機 関 等
事務局員	佐藤 能行	みずほ情報総研株式会社 社会システム評価研究部 部長
事務局員	佐久間 敦	みずほ情報総研株式会社 社会システム評価研究部 主事研究員
事務局員	冨田 高樹	みずほ情報総研株式会社 社会システム評価研究部 主事研究員
事務局員	中山 和郎	みずほ情報総研株式会社 社会システム評価研究部 研究員

本報告書のポイント

○ 検討の必要性

- 情報システムの安全性を確保するためには、コンピュータシステムの中核となるオペレーティングシステム（OS）のセキュリティ機能について検討することが有益

○ セキュア OS とは

- OS の機能について、セキュリティの確保という観点から、様々な機能向上や新機能の研究開発が行われてきており、「セキュア OS」という製品群が存在
- UNIX や Linux、Windows では、システムの設定や管理のためにオールマイティな権限を持つ管理者（スーパーユーザ）が存在し、これがセキュリティ上の問題のひとつ
- コンピュータシステム内の主体（ユーザやプログラム）が持つ権限は必要最小限にすべき（最少特権の考え方）
- UNIX や Linux、Windows のアクセス制御（任意アクセス制御）では、コンピュータシステム内で、セキュリティポリシーが一貫せず統一されたアクセス制御が実現できない可能性あり
- したがって、システムポリシーをコンピュータシステム内に強制できる強制アクセス機能が必要

  ＊システムポリシー： ユーザのアクセス権限、プログラムの動作範囲などを細かく設定する際のアクセス制御方針

- セキュア OS とは、「最少特権や強制アクセス制御機能を中核とした、セキュリティに配慮した OS」

○ 電子政府への適用可能性

- 職員用 PC 等にセキュア OS を適用することも考えられるが、以下の理由から当面はサーバー 側を優先すべき
  1. サーバーには重要情報が集中して保存されており、セキュリティ対策実施の優先度が高い
  2. クライアント側にセキュア OS を適用する場合、既存資産の承継の必要性やユーザインタフェースなど、セキュリティ以外の要素の影響が大きい

- 情報の電子的提供や申請・届出等手続および政府調達のオンライン化を図るため、公開情報サーバーシステムを構築。外部向けサービスは、Web サーバーシステムとして構築されており、攻撃者による脅威に曝されている

- 公開鍵暗号技術をもとに、本人認証やファイルやメールについてなされるデジタル署名の検証などの社会的サービスを実現するために、認証局と呼ばれる機構が必要。認証局システムは、インターネット越しのユーザ（国民・事業者）との関係において、信用の基礎となるシステムであるので、アクセス制御を含めて十分にセキュリティが確保されることが必要

- 省庁の行政業務を支援する典型的なシステムが文書管理システム。職員の地位や権限に応じて、適切にアクセス制御が行われることが必要

○ セキュア OS の適用形態

- 電子政府でのセキュア OS の利用について、以下のモデルを想定。このモデルごとに、セキュア OS の適用効果、適用の問題点、運用への影響、業務への影響などについて、十分検討することが必要
  ① OS のみのセキュリティ強化
  ② OS・ミドルウェア・アプリケーションを個別に強化
  ③ セキュア OS の機能を利用したミドルウェアとアプリケーションの強化

○ セキュア OS 活用のための検討課題

- OS に期待される主要な機能のひとつは、様々なアプリケーションがその OS 上で実行できること。セキュア OS がより一般的に利用されていくためには、一般的な商用及びオープンソースのアプリケーションが動作することが重要

- セキュア OS の管理・運用には高度の知識を必要とするため、サポートを提供する企業が限定され、製品によっては十分なサポートを受けることができるか不明なものも存在

- セキュア OS は、システムポリシーに従って構築されるもの。このポリシーを十分使いこなして設定しなければ十分なセキュリティは確保できず。ポリシー構築・設定の担当者のためのオペレーション教育の充実や、その際に準拠すべきガイドラインの策定が必要

- システムポリシーの設定は煩雑な作業になりがちであり、現行のセキュア OS 導入の阻害要因のひとつ。セキュリティの設定は、ベンダーではなく可能な限り運用主体自身が行えることが望まれるところ。エンドユーザでも設定しやすいツールの登場が、セキュア OS の導入

範囲拡大に不可欠

- 100%セキュアなシステムは存在せず。システムの安全性とは、セキュリティを提供する機能に対してどこまで信頼できるかの問題。今日開発されているセキュア OS で、セキュリティ評価・認証制度に基づく認証を取得したものは少数。安心してセキュア OS を利用するためには、こうした認証は不可欠

○ 検討の方向性

- 電子政府には、軍事情報、外交情報、捜査情報等の秘匿性の高い情報はもとより、個人情報のように、その漏洩防止に万全の対策を講ずべき情報が多数存在。情報の秘匿性に加え、政府のホームページ改ざんのような事態を踏まえて、データの完全性も重要。この秘匿性や完全性を確保するためには、情報システムのセキュリティを OS レベルにおいても向上させる必要があり、セキュア OS を可能な限り導入することが有効

- 電子政府のセキュリティを、可能な限り早期かつ広範囲において向上させるためには、その保有する情報資産の重要性と導入の容易性を総合的に勘案して、当面セキュア OS を導入すべきシステムを具体的に検討することが必要

- 電子政府における OS のセキュリティ要件を検討する中で、既存の製品では必要とする要件を満たさないことも想定されるところ。この場合は、わが国において「セキュア OS」の開発を行うことも選択肢のひとつ

第１章 セキュア OS 概説

情報セキュリティにかかわるインシデントには、「コンピュータを乗っ取られる」という言い方がなされる極めて深刻なものがある。このようなインシデントを正しく理解し、対策を考えていくためには、OS というコンピュータにおける最も基本的なプログラムの機能を理解することが必要不可欠である。また、OS の機能自体についても、セキュリティの確保という観点から、従来から様々な機能向上や新機能の研究開発が行われてきており、いわゆる「セキュア OS」という製品群が存在する。

第１章では、コンピュータシステムにおいて OS はどのような役割を果たしているのか、「セキュア OS」とはどのようなもので、一般的な OS と比較して何が違うのか、「セキュア OS」の歴史を概観しつつ説明することとしたい。

１.１ OS の基本機能

（1）基本ソフトとしての OS

OS とは、オペレーティングシステム（Operating System）の頭文字をとったものである。

コンピュータシステムは、CPU（中央処理装置）という計算装置やデータを保存しておく記憶装置、表示装置（ディスプレイ）といったハードウェアとワープロソフトやインターネットのためのブラウザ、電子メール用ソフトといったソフトウェアから構成されている。

このソフトウェアをさらに区分すれば、いわば各種のソフトウェアの基盤、土台ともいうべき OS と、その他のソフトウェアに分かれる。

この OS という基盤、土台の上で動き、ユーザがコンピュータにさせたい仕事（例えば文書の作成）を処理するソフトウェア（ワープロソフト）を、アプリケーションソフトウェア（応用ソフトウェア）あるいは単にアプリケーションと一般に呼んでおり、したがってこれとの対比で OS のことを、基本ソフトウェアと呼ぶことがある（図１－１）。

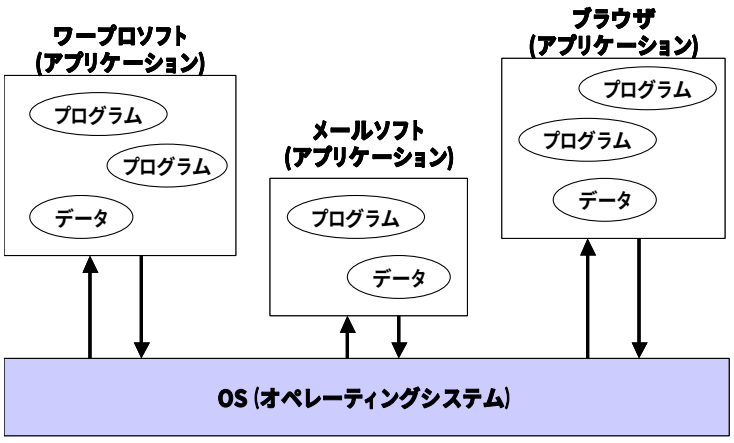


図 １－１　オペレーティングシステム

（2）OS のファイル管理

OS はコンピュータ内の様々なソフトウェアやデータを、管理し制御している（付録 A 参照)。一般的にその管理や制御は、ファイルという単位で行われている。アプリケーションや OS そのものも、このファイルの集合体である。

コンピュータハードウェアの観点からいえば、各種のアプリケーションやデータは補助記憶装置と呼ばれているハードディスクの上に、デジタルデータ（0 と 1 の組み合わせ）という形で散在している。これを OS が管理するために適切な単位であるファイルというまとまりにし、さらにこれをフォルダもしくはディレクトリという、いわば入れ物、箱を作って効率よく情報の出し入れができるようにしているのである（図１－２)。

コンピュータセキュリティの観点からは、このファイル管理が問題となる。悪意の第三者は、コンピュータ内のデータを不正に入手したり、あるいは破壊したり、またデータの一部を書き換えたりしようとする。したがって、このような行為を防ぐために、OS の管理の対象であるファイルをどう保護するかが、コンピュータセキュリティ対策の重要課題となる。

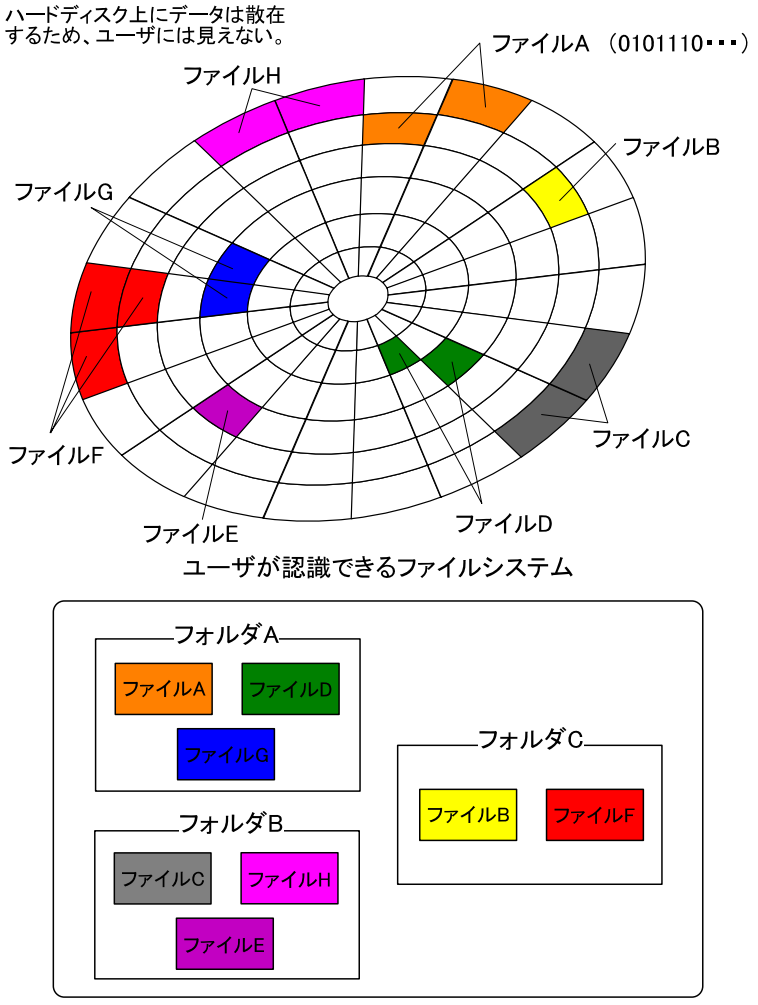


図 １－２　OS のファイル管理

（３）アクセス制御

ユーザ A が作成したファイル B があるとして、このファイルを他のユーザには見せたくないが、ユーザ C だけには見せてよいと考えているとしたら、ユーザ A はユーザ C にだけファイル B を読む権限を与え、他のユーザがファイル B を読めないように設定することができる。ファイルに書き込む権限についても、同様に設定できる。また、プログラムもファイルとして保存しているので、このプログラムを実行する権限も、同様に設定できることになっている。このような OS の機能を、アクセス制御と呼ぶ（図１－３）。

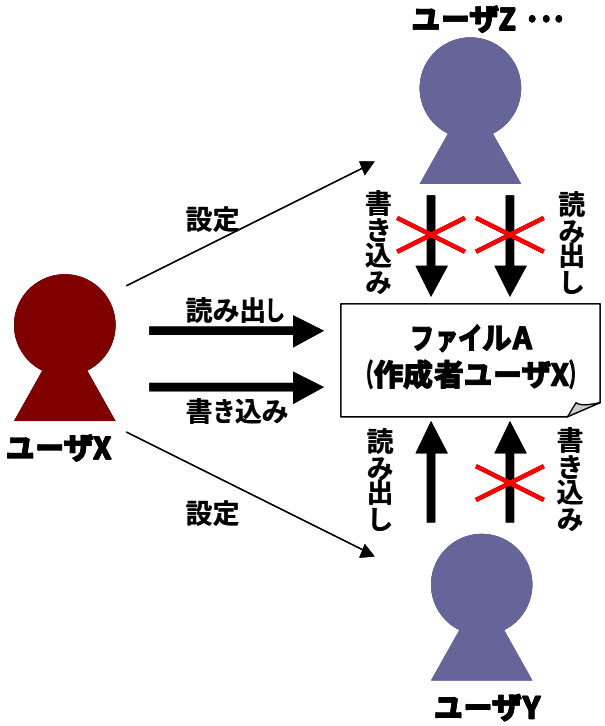


図 １－３　アクセス制御

この例のように、ファイルの所有者（作成者）が自分のファイルに対する他のユーザのアクセス権限を設定することを、任意アクセス制御もしくは自由裁量アクセス制御（DAC：Discretionary Access Control）と呼んでいる。ファイルの所有者が、自由（任意）にそのファイルに対するアクセス制御を決定できる、と言う意味である。

このようなアクセス制御機能は、例えば Unix や Linux、Windows XP といった一般の OS に備わっている。しかし、この DAC 機能だけでは必ずしもセキュリティの確保が十分でないという考え方から、いわゆる「セキュア OS」の多くが開発され製品化されてきたのである。

１.２ セキュア OS の歴史

「セキュア OS」という用語は比較的新しい用語であり、後で再度説明するが、その定義や用語の使われ方も必ずしも一定ではない。むしろ歴史的には、トラステッド OS（Trusted OS）という用語が一般的だった。この用語は、1985 年に米国国防総省の規則として制定された Trusted Computer System Evaluation Criteria（TCSEC）で定義された B1 クラス以上の基準を満たす OS を意味するものとして使われてきた。

しかし、本報告書では以下の理由から、「セキュア OS」という言葉を使うこととする。

- 米国においてもこの TCSEC はその歴史的な役割を終え、国防総省の規則としては廃止され、それに代わって米国政府としても国際基準である Common Criteria を採用してしている。
- 現在入手可能なセキュリティ機能を向上させた OS は、TCSEC の基準を満たしていないものや、TCSEC にはない新しい機能を採り入れた製品が多数ある。

（1）1970 年代

1960 年を通じてコンピュータは、タイムシェアリング機能、マルチプログラミング機能（付録 A 参照）を備えたものとなり、米軍とりわけ米空軍にこのような先端的なコンピュータシステムが導入されつつあった。 これらのシステムは複数のユーザが使用するマルチユーザシステムでもあるから、悪意のあるユーザがセキュリティ上問題のあるプログラムをインストールする危険がある。この当時特に懸念されたのが、１．３で紹介するトロイの木馬であった。トロイの木馬が軍関係のシステムにインストールされることによって、善意のユーザがそれとは知らずに軍事機密情報を漏洩してしまう危険性があったのである。

このため米空軍を中心に資金が提供され、当時の言葉を使えば「リソースシェアリング・システム」におけるセキュリティの研究が活発となった。この成果として、レファレンスモニターというコンセプトやベル・ラパデュラモデル（Bell-LaPadula：BLP モデル）といったセキュリティモデルが提唱され、これが TCSEC の背景にある基本的な物の考え方となった。

この BLP モデルの中核的な考え方は、階層で区分された情報の間では「情報が下の区分に流れない」ようにアクセス制御を行うことである。例えば、「極秘」情報を扱うことができるユーザは、それよりも下位の「秘」ファイルは読むことができるが、「秘」情報までしか扱うことができないユーザは、それよりも上位の「極秘」ファイルを読むことは許されない。他方、「極秘」ファイルを読むことができるユーザは、「秘」以下のファイルに書き込むことは許されない。これを許せば、「極秘」情報が「秘」以下のファイルに漏洩されてしまう可能性があるからである（図１－４および付録 C 参照）。

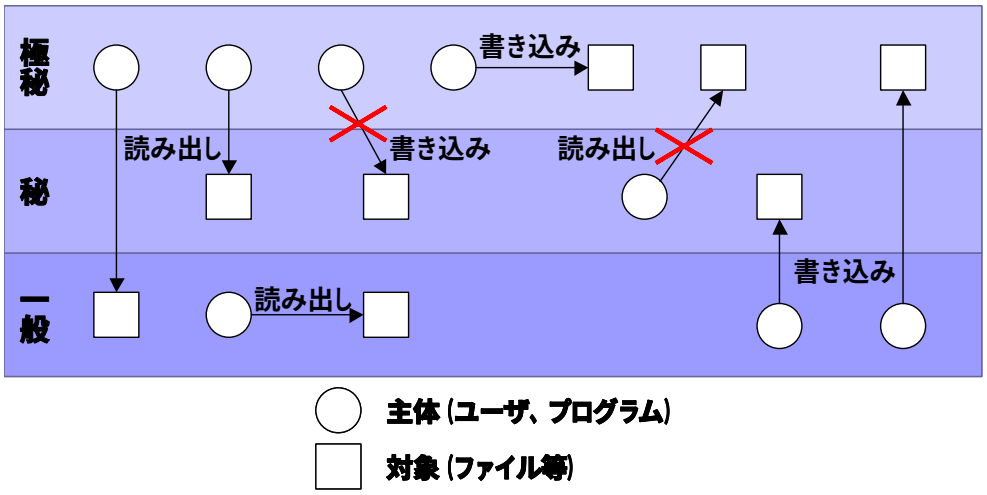


図 １－４　ベル・ラパデュラ（Bell-LaPadula(BLP)）モデル

（２）1980 年代

1970 年代の理論的な研究を受けて、1980 年代にはトラステッド OS が登場する。1983 年には、TCSEC の第一版が公表され、1985 年に国防総省の正式な規則として制定された。TCSEC では、セキュリティ評価の基準として、レベルの低い方から順に D、C1、C2、B1、B2、B3、A1 というクラスを設け、それぞれの評価基準を定めた（付録 C 参照）。

この TCSEC で、BLP モデルを実現するアクセス制御が、B1 クラス以上に要求されることとなった。そして TCSEC では、このようなアクセス制御を前述した任意（自由裁量）アクセス制御と区別して、強制アクセス制御（MAC：Mandatory Access Control）と呼んでいる。

この当時は、米国政府特に米軍の調達を念頭において開発されたトラステッド OS が多く、民間で利用するといった意味の商用 OS ではなかった。代表的な製品としては、SCOMP(Secure Communications Processor)が挙げられる。この OS は、TCSEC で規定された最上位のクラスであるＡ１に認定された唯一の製品であり、米軍のシステムで採用された。

また、1970 年代を通じて研究開発が行われていた MULTICS という汎用 OS に対しても、TCSEC の基準に基づいてセキュリティ機能を向上させる研究プロジェクトが米空軍の資金提供のもとで行われ、B クラスの認定を受けている。この他にも、数多くの OS が TCSEC の認定を受けているが、この時代のトラステッド OS は基本的には軍用を念頭においたものだった。

（３）1990 年代

1990 年代に入ると、一般の企業向けのトラステッド OS の開発も行われるようになってきた。例えば、銀行業務のオンラインシステムのセキュリティを確保するためにトラステッド OS を採用する金融機関が増えたり、インターネットビジネスの発展に伴ってインターネットサーバーの

セキュリティ確保のためにトラステッド OS を採用する企業が現れるなど、一定の民間需要が期待された。サン･マイクロシステムズ社の Trusted Solaris やヒューレット･パッカード社の Virtual Vault、アーガスシステムズ社の PitBull などの代表的な製品は、この時期に開発されている。

また、TCSEC には一般の商用 OS レベルのセキュリティ機能としては十分と考えられるＣクラスの評価基準もあることから、例えばマイクロソフト社の Windows NT のような製品も TCSEC の認定を受けている。トラステッド OS のような高機能ではないものの、TCSEC の認定を受けることでその製品が信頼性の高いものであることを公的に証明してもらうことが、その製品のセールスポイントとなることを期待してのことと考えられる。

また 1990 年代の注目すべき傾向は、米国の TCSEC 以外にもコンピュータ製品のセキュリティ評価基準作りが各国で行われたことである。

代表的なものとして、英国、ドイツ、フランス、オランダが共同して 1991 年に策定した ITSEC(IT Security Evaluation Criteria)がある。Windows NT は、この ITSEC の認定も受けており、サン・マイクロシステムズ社の Solaris 2.6 / 8 および Trusted Solaris 2.5.1 / 8 も認定を受けている。

また 1993 年には、米国、カナダ、英国、フランス、ドイツ、オランダの六カ国が各国のセキュリティ評価基準を統一するための活動に着手し、この成果として CC(Common Criteria)と呼ばれる評価基準が作成され、1999 年にはこれが ISO/IEC 15408 として国際標準として採用された。我が国においても、これを平成 12 年に日本工業規格 JIS X 5070 として制定した。CC は、TCSEC とともに後で詳しく説明する。

（4）2000 年代

前述した 1990 年代に開発されたトラステッド OS 製品は、今日でも依然として代表的な製品となっている。しかし、近年の特徴的な動きは、オープンソース・ソフトウェアをベースにしたセキュア OS の研究開発が盛んとなっていることであろう。

代表的なプロジェクトとしては、UNIX 系の FreeBSD という OS をベースとした Trusted BSD がある。また、米国防総省の機関である NSA (National Security Agency)が中心となって、Linux をベースとした SELinux (Security-Enhanced Linux)を開発している。これは、例えば Linux の有力なディストリビューションである Fedora Core 等に最近組み込まれるようになり、今後広く普及すると考えられている。また、フランス政府も、やはり Linux をベースにしたセキュア OS を独自に開発するプロジェクトを開始した。

さらに近年の傾向として、一般的な OS と分類されてきたような OS のセキュリティ機能が格段に向上してきたことが指摘できるであろう。例えばサン・マイクロシステムズ社の Solaris 10 やヒューレット・パッカード社の HP-UX 11i という最新バージョンの OS は、多様なセキュリティ機能を搭載している（付録 F 参照）。

１.３ 脅威

今日のコンピュータシステムは、インターネットの普及ともあいまって様々な脅威にさらされている。したがって、このような各種の脅威に対する対策も、相当程度実用化されてはいる。しかし、これらの対策も完全ということはありえない。本節では、このような脅威のなかで、一般的な OS では対抗しにくい脅威の典型例を紹介するとともに、現在、実用化されている一般的な対抗策だけでは十分脅威に対抗できないケースを紹介することによって、「セキュア OS」の必要性、有効性を説明することとしたい。

１.３.１ OS に関わる典型的な脅威

（1）トロイの木馬

トロイの木馬とは、利用者に対して正常なプログラムを装ってコンピュータに仕掛けられ、データの削除やファイルの流出などの活動を行い、利用者に被害を与えるプログラムである。その活動内容や活動時期は様々で、トロイの木馬は一見正常なプログラムとして動作しているように見えるため、通常の利用では、利用者がトロイの木馬が活動していることに気づくのは難しい。前述したように、1970 年代に米軍が最も懸念した脅威が、このトロイの木馬であった。そこで、情報漏洩の観点から、なぜトロイの木馬が脅威となるのか、ワープロソフトにトロイの木馬が仕掛けられているケースで説明することとする。

ユーザ A は、自分が作成した「秘」の情報が入ったファイル A に対して、read 権も write 権も有している。「秘」の情報を扱うことができないユーザ B は、「秘」の情報が入っていないファイル B に対しては、read 権や write 権を有しており、ユーザ A も read 権と write 権を有しているとする。

任意アクセス制御(DAC)が実現していれば、ユーザ B はファイル A にアクセスできないので、ファイル A の秘情報は漏洩することはない（図１－５）。

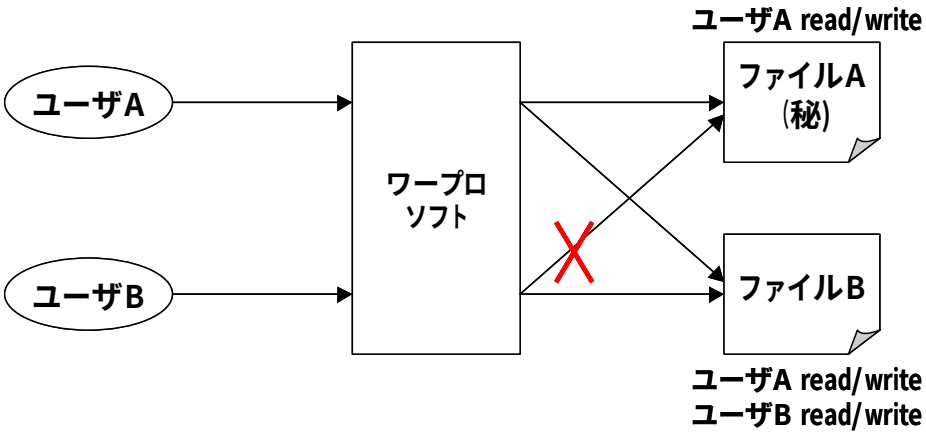


図 １－５　任意アクセス制御による秘情報の漏洩防止

しかし、このワープロソフトに、正規のユーザが行う通常の操作（例えば読出し）の裏で、そのユーザに気付かれない形で異なる操作（例えばコピー）を行うようなトロイの木馬が仕掛けられていると、情報は漏洩してしまう（図１－６）。

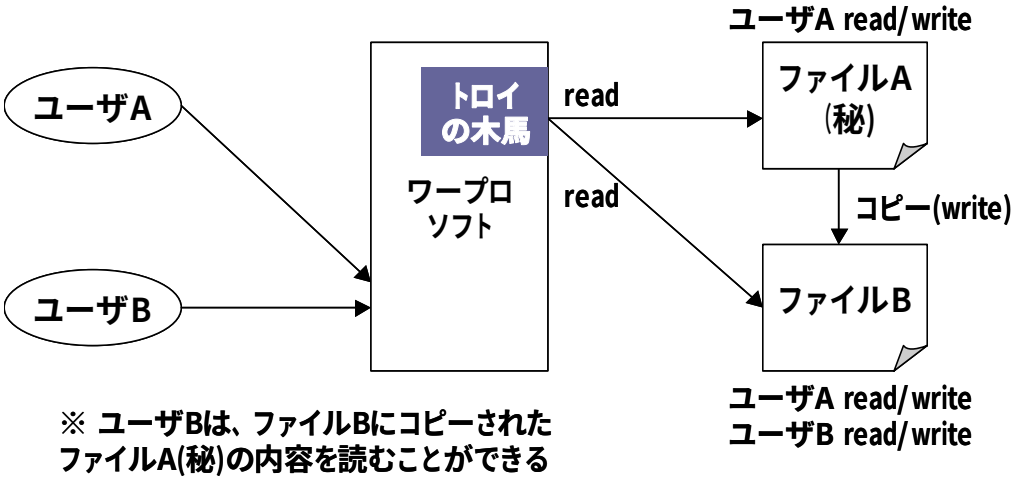


図　１－６　トロイの木馬による秘情報の漏洩

ユーザ A がファイル A を読み出すと、このトロイの木馬を仕掛けられたワープロソフトは、ファイル A についてユーザ A は write 権があることから、ファイル A の内容をファイル B にコピーしてしまう。本来なら秘情報にアクセスできないユーザ B は、ファイル B にアクセスすることで、ファイル A の秘情報にアクセスできるようになる。これが 1970 年代の米軍が懸念した事態だった。

しかし、前述した BLP モデルを実現する強制アクセス制御（MAC）の下では、たとえファイル B についてユーザ A が read 権や write 権を有していても、ファイル A が「秘」に区分されファイル B が「一般」であれば、ファイル A からファイル B へのコピーは許可されない（図１－６）。したがって、ワープロソフトに仕掛けられたトロイの木馬は、所期の目的を達成できないこととなる。

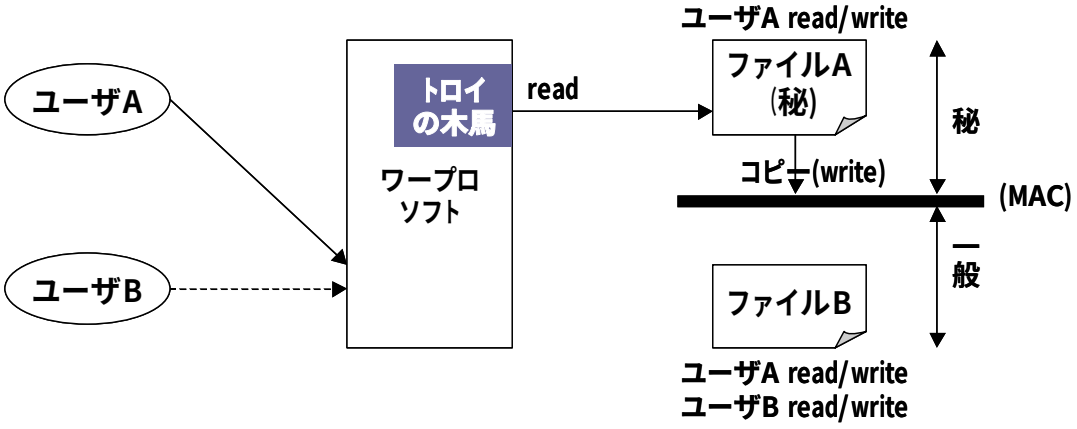


図　１－７　強制アクセス制御による秘情報の漏洩防止

（２）バッファ・オーバーフロー（権限の昇格、管理者権限の奪取）

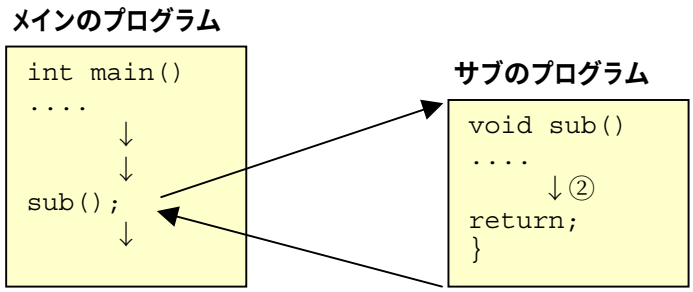
プログラムの作成には、「バグ」と呼ばれるソフトウェアの欠陥が伴いがちである。「セキュリティ・ホール」と言われるプログラムにおけるセキュリティ上の脆弱性の大半は、この「バグ」だと言われている。そしてこの「バグ」の相当部分が、主要な OS のプログラム言語であるＣ言語やＣ++言語で発生しやすい「バッファ・オーバーフロー」の問題である。

プログラムを実行する際には、メモリ（記憶装置）上に、そのプログラムに実行上必要となるデータを一時的に保管するための領域（バッファ）を用意しておく必要がある。したがって、プログラマは自分の作成しているプログラムに応じて必要と考えられる容量の領域を用意することとなる。しかし、何らかの理由でこの領域の容量を超えたデータが入力されると、この領域から溢れたデータが、その領域の周囲にあるメモリ情報を上書きして壊してしまい、プログラムが予期せぬ動作をする。このような現象を「バッファ（領域）がオーバーフローした（溢れた）」と言い、この現象を利用する悪意の第三者の攻撃をバッファ・オーバーフロー攻撃と呼んでいる（図１－８）。

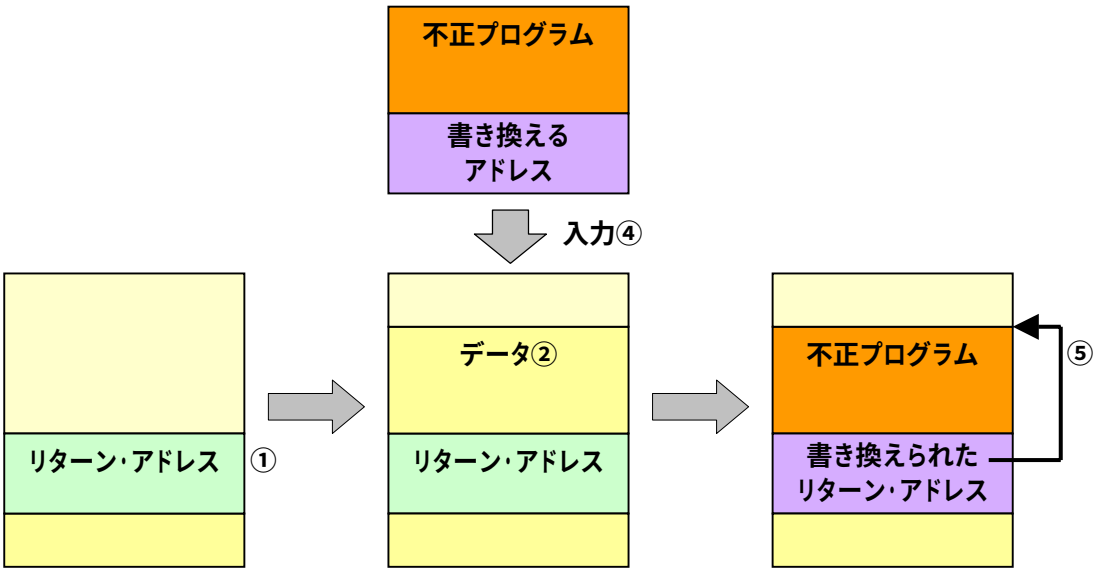
このオーバーフローして上書きされるデータとして、攻撃者が不正プログラムを用意しておけば、本来のプログラムに代わってこの不正プログラムが実行されることとなる。本来のプログラムが管理者権限で実行されるものであれば、攻撃者が仕組んだプログラムも管理者権限で動作することとなるので、およそあらゆる命令を実行することができるようになり、コンピュータはいわば「乗っ取られた」状態となる。

セキュア OS では、この管理者権限そのものを分割するので、仮にその分割された管理者権限を奪取されても、被害を最小限に抑えることが可能であり、また個々のプログラム（アプリケーション）についても、そのアクセス権限を必要最小限のものに限定できるので（最少特権の考え方）、仮にそのプログラムに何らかのセキュリティ・ホールがあっても、やはり被害を局限化することができるのである。

○プログラムの動き



○バッファ（領域）の動き



複雑な処理が必要なプログラムは、主要な処理を行うメインのプログラムと一定の処理を行うサブプログラムで構成される。この処理の流れの中で「一時的」に使用するデータを格納する場所が、スタック領域である。

① メインプログラムの処理の途中で、サブプログラムの処理を終了した時に、メインプログラムに戻るときの場所を「リターンアドレス」としてスタック領域に格納する。

② サブプログラムを処理するために必要な「一時的」なデータをスタック領域に格納する。

③ 通常であればサブプログラムの処理が終了すれば、①のリターンアドレスに従ってメインプログラムの一定の場所に戻って処理が続行される。

④ しかし、悪意の第三者によって不正プログラム及び新しいリターンアドレスが入力されると、②のデータや①のリターンアドレスが上書きされてしまう。

⑤ 書き換えられたリターンアドレスによって、不正プログラムの実行が開始されてしまう。

図 1－8　バッファ・オーバーフロー

（3）内部犯行

情報漏洩の要因には、正規のアクセス権限を持つ内部のユーザによる犯行が多いとされている。例えば、一般的な OS では、管理者権限を持つユーザはアクセス制限を受けずにすべてのリソースにアクセスできる。また、管理者権限を持たなくても、正規に割り当てられたアクセス権限の範囲内で不正アクセスを行うこともできる。一般的に内部ユーザは、そのシステムについての内部情報をより容易に入手できる立場にあり、さらに不正プログラムを持ち込むことも比較的容易である。このため、外部から侵入されなくても内部の悪意のあるユーザにより、プログラムの権限を奪取され、管理者権限を奪われる可能性がある。

このような被害を最小に食い止めるには、ユーザやプログラムに対して必要最小限のアクセス権限のみを付与する機能が必要である。セキュア OS では、ユーザの役割毎に必要最小限のアクセス権限を与えることが可能であり、権限内の不正アクセスの被害を最小に抑えることができる。さらに、セキュア OS は、ユーザだけでなくシステム内の各種プログラムについても、個々のプログラムごとに必要最小限のアクセス権限を付与する機能を有しており、不正アクセスの被害を局限化することができる。

また、一般的な OS では管理者権限があれば自らログファイルを改ざんすることで内部犯行の証拠を消去することができてしまうが、セキュア OS ではログファイルに追記のみ可能な属性を与えることで、改ざんや削除を不可能にすることができる。これにより、管理者による内部犯行を抑止する効果が得られる。管理者の内部犯行への対策としては、このほか「デュアルロック」と呼ばれる機能を備えた製品もある。この機能によって、システム管理上の重要な設定は二人以上の管理者が行う必要が生じるため、これまでの OS では管理者が誰にも気づかれずに実施し得るような内部犯行を抑止することが可能となる。

1.3.2 防御方法

（1）ファイアウォール

ファイアウォールは、ネットワークを利用して外部からの侵入を防ぐためのシステムである。外部からの通信元と通信先の IP アドレスとポート番号を見て、アクセスの可否を決定する。不必要な通信を遮断できるものの、アクセスを許可した通信内容については関知しない。例えば外部から公開 Web サーバーへの HTTP のアクセスを許可しているとしたら、このサービスを利用した攻撃を防ぐことはできない。このため仮に侵入を許した場合にも、その被害を最小限に抑えるためには、セキュア OS の導入などによる対処が有効である。

（2）侵入検知システム（IDS: Intrusion Detection System）

IDS はネットワークを流れるパケットやサーバーのログを監視し、侵入行為を検知するシステ

ムである。既知の不正アクセスのパターンを利用した不正検知と過去の履歴を基にした異常検知に分けられる。不正アクセスの検知に有用ではあるものの、誤検知や見逃しは避けられず、またそもそも未知のパターンによる不正アクセスを検知することはできない。したがって IDS だけでは防御は不十分である。このため、侵入を許した後の被害を最小限に抑えるために、セキュア OS の導入が有効である。

（3）暗号

古くから通信内容を盗聴されないようにするために、暗号が利用されてきた。最近では例えば Windows XP などで採用されている NTFS というファイルシステムでは、ファイルやフォルダを暗号化してハードディスクに保存できるようになっている。

したがって、重要な情報が入っているファイルが仮にシステムの外部に流出したとしても、暗号化してあればその内容が外部の者に知られるリスクは回避できる。また暗号化してあればそのファイルの内容が改ざんされたかどうかは、すぐに判断できるようになる。

しかし、この暗号化といえども、万全ではない。そもそも暗号化するのも、また暗号をもとに戻す、すなわち「復号化」するのも一定のプログラムである。このプログラムを改ざんなどから保護するためには、やはりシステム内の厳格なアクセス制御が必要である。

また暗号化によっても、ファイルやフォルダの改ざんや消去そのものは阻止することはできない。このような被害を局限化するためには、やはりセキュア OS の導入が有効である。

（4）パッチ

パッチとは、プログラムの修正内容を集めたファイルである。パッチをプログラムに適用することでプログラムを修正し、バグなどの不具合を解決する。プログラムのバグやセキュリティの脆弱性の発見後、パッチを開発して配布を行うため、パッチを適用するまでは脆弱性を利用した攻撃には対処できない。多くのソフトウェアは潜在的に何らかの脆弱性を持つことが予想されている。このため、未知の脆弱性を利用した攻撃を受けた場合にでも、その被害を最小限に抑えるためにセキュア OS を導入し、対処することが有効である。

以上のことから、一般の OS と既存の防御技術では、対処できない不正アクセスがあり、また例えば、管理者権限を持つ内部ユーザによる犯行も阻止できない。したがって、たとえ侵入を受けたとしても、被害を最小限に抑えるためのきめ細かいアクセス制御を実現できるセキュア OS の導入は有効な対策である。

１．４ 「セキュア OS」とは何か

前述したとおり、トラステッド OS という用語は、TCSEC の B1 以上の認定を受けた OS もしくはそれに相当する機能を有している OS を一般に意味していた。そして講学上、トラステッド OS が備えるべき機能として、①ユーザの識別と認証、②MAC 機能と DAC 機能、③オブジェクトの再利用保護、④完全な仲介、⑤監査ログ、⑥トラステッド・パス、⑦侵入検知、が挙げられる。

しかし、そもそも TCSEC 自体は廃止されていることや、例えば SELinux のように極めて粒度の細かいアクセス制御を実現した OS が登場していることから、本報告書ではトラステッド OS という用語よりも、「セキュア OS」という用語を使うこととした。

ただし本報告書は、この「セキュア OS」の厳密な概念定義をすることが目的ではない。むしろ重要なことは、近年の OS 開発においては、セキュリティ機能に配慮した製品開発の傾向にあるという認識に立って、このような多様な「セキュリティに配慮した OS」を電子政府においてどのように利活用すべきか検討することである。このような意味からも、「セキュア OS」を厳密に定義することはしないが、これから検討を進めるうえで「セキュア OS」のイメージを共有することは必要である。したがって、この「セキュア OS」のイメージについて若干触れておきたい。

UNIX や Linux では root、Windows では Administrator という、システムの設定や管理のためにオールマイティな権限を持つ管理者（スーパーユーザ）の存在がセキュリティ上の問題であることは、バッファ・オーバーフローを例に説明した。したがって、この管理者権限を可能な限り分割するほうが、セキュリティの観点からは望ましい。プログラムについても、管理者権限で動くものがあるが、これも可能な限り権限を制限すべきである。より一般的に言えば、コンピュータシステム内の主体（ユーザやプログラム）の持つ権限は必要最小限にすべきだ、ということができる。この考え方を、最少特権（least privilege）と呼ぶ。

また、任意アクセス制御（DAC）についても、セキュリティ上問題がある（付録 C 参照）。コンピュータシステム内で、セキュリティポリシーが一貫せず統一されたアクセス制御が実現できない可能性があるのである。したがって、広い意味での MAC 機能（狭い意味では MAC は BLP モデルを実現する機能）もしくは非任意アクセス制御とでも呼ぶべき機能、すなわち何らかのシステムポリシーをそのコンピュータシステム内に強制できる機能が必要である。

したがって、あえて「セキュア OS」のイメージをまとめれば、「最少特権や広義の強制アクセス制御（MAC）機能（非任意アクセス制御）を中核とした、セキュリティに配慮した OS」と言うことができるだろう。以下本報告書では、MAC 機能はこの広義の MAC 機能の意味で使用することとする。

第２章 セキュア OS の導入

本章は、組織が情報システムに対してセキュア OS を導入する際、どのような点を考慮する必要があるかについて整理する。またセキュア OS の具体的な設定管理や運用の操作例を示すことを通じて、その特徴を明らかにする。

２.１ セキュア OS を導入する際に考慮すべきポイント

（１）セキュア OS の選定

セキュア OS の導入にあたっては、まず導入の対象と目的を明らかにすることが重要である。システム全体のセキュリティにおいて、システムのインフラを構成する OS のセキュリティはまさに「要」であり、セキュア OS 導入の影響はシステム全体に及ぶ。

セキュア OS 導入の目的を明らかにした上で、用途と目的にあった OS を選定しなければならない。セキュア OS においては、強制アクセス制御をはじめとする種々のセキュリティ機能が拡張されているが、そうした機能には予め適切な設定の実施を必要とするものが多い。いたずらに高機能な OS を選定すると、必要でない機能のために運用管理工数が増大し、システムの状態把握がおろそかになる可能性もあるので注意しなければならない。

（２）「組織の情報セキュリティポリシー」に対する「システムポリシー」の設定

組織では、情報セキュリティ対策を自組織の活動の中に位置付け、どのように対処すべきか明らかにし、徹底を図ってゆく必要がある。組織が保有する情報資産（情報及び情報システム）を様々な脅威から保護し、情報セキュリティの基本要素である機密性、完全性、可用性を確保、維持するための基本方針や具体的な規定を明文化したものを「情報セキュリティポリシー」と呼ぶ。今や多くの組織において、トップレベルの情報セキュリティポリシーから、詳細な規定を定めた「スタンダード（対策基準・標準)」や「プロシージャ（実施手順)」等を整備している。

この組織の情報セキュリティポリシーを、例えば実際の文書管理システムに反映させるには、システムの実装設計に先だって当該文書管理システムの「システムポリシー」を構築する必要がある。システムポリシーとは、システムリソースとユーザの関係性についての方針であり、代表的な例としてユーザのアクセス権限などを設定するためのアクセス制御方針が挙げられる。システムポリシーはセキュア OS を導入する際、ユーザのアクセス権限や、プログラムの動作範囲を制御する属性などを細かく設定するために必要不可欠なものである。図 ２－１に組織の情報セキュリティポリシーとシステムポリシーの関係を示す。

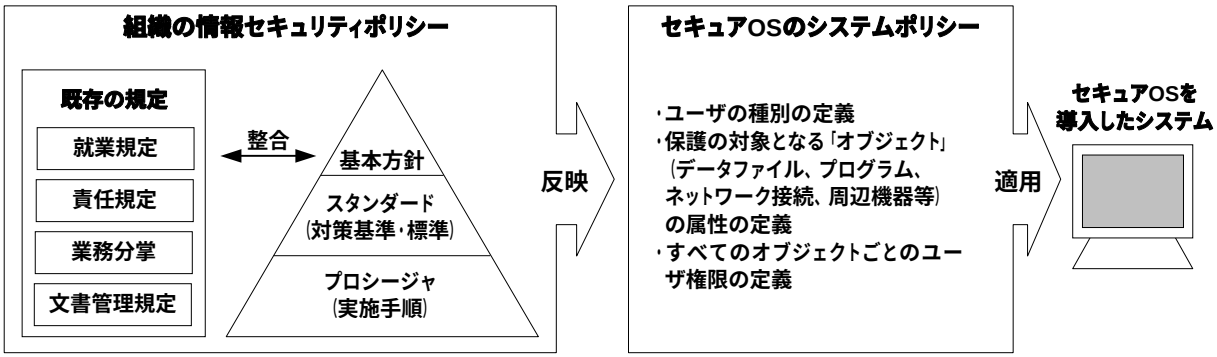


図 2－1　組織の情報セキュリティポリシーとセキュア OS のシステムポリシー

ユーザがデータファイルやプログラムにアクセスしようとする場合、セキュア OS では必ずシステムポリシーと照合し、アクセスの可否が判定される。プログラムの動作も同様にして、属性に応じてその１つ１つの動作がシステムポリシーと照合され、動作可否が決定される。セキュア OS が一般的な OS と異なるのは、ユーザ、データファイル、プログラムにおけるアクセス権限や属性を、システムポリシーによって非常にきめ細かく規定することが可能な点にある。以下にシステムポリシーに設定される項目の例を示す。

表 2-1　セキュア OS のシステムポリシーの設定項目の例

項目の例	内容
分割定義	セキュア OS が管理するすべてのファイルやプログラムに対して、セキュリティ的に分割・隔離する単位を定義する（このような分割機能を「ドメイン」や「コンパートメント」と呼ぶ製品もある）
セキュリティ属性	ファイル、プログラム、周辺機器といった OS の管理する対象（「オブジェクト」という）のセキュリティに関する設定（アクセスの制限の設定、情報の機密度（機密／秘密／公開等）などが含まれるが、設定の内容や方法は製品によって異なる）
ユーザ属性	ユーザの ID、グループおよび役割の設定
ネットワークルール	どのネットワークから接続要求がされたかによって、アクセス権限を変更するためのルール設定

（3）セキュア OS における管理者の権限

標準の Linux / UNIX においては、ユーザの概念は一般ユーザとシステム管理者に区分される。この内システム管理者は、システムに関してあらゆる操作ができるように最高の権限である管理者権限（root）をもつ。

セキュア OS では、ユーザの権限を用途と役割に基づき細かく分割することが可能であるため、

必要最小限の権限のみを割り振ることができる（最少特権）。つまりシステム管理者といえども、すべてのファイルに対してすべてのアクセス方法が許可されるというような設定を行わないといった選択が可能である。セキュア OS の導入にあたっては、管理者や一般のユーザの役割を明確にし、それぞれどのような権限を与えるべきか、事前に十分に検討することが望まれる。

（4）運用管理

このセキュア OS の高度なアクセス制御機能を有効に活かすためには、システムポリシーを常に運用管理し続けなければならない。つまり一般的な OS においてもファイルやディレクトリの割り当てや、導入するパッケージの構成、サービスの起動内容、ログの出力内容等を考慮することが求められるが、図２－２に示すようにこれら一般的な OS における運用管理に加えて、セキュア OS のシステムポリシーの運用管理を行うことが必須となる。

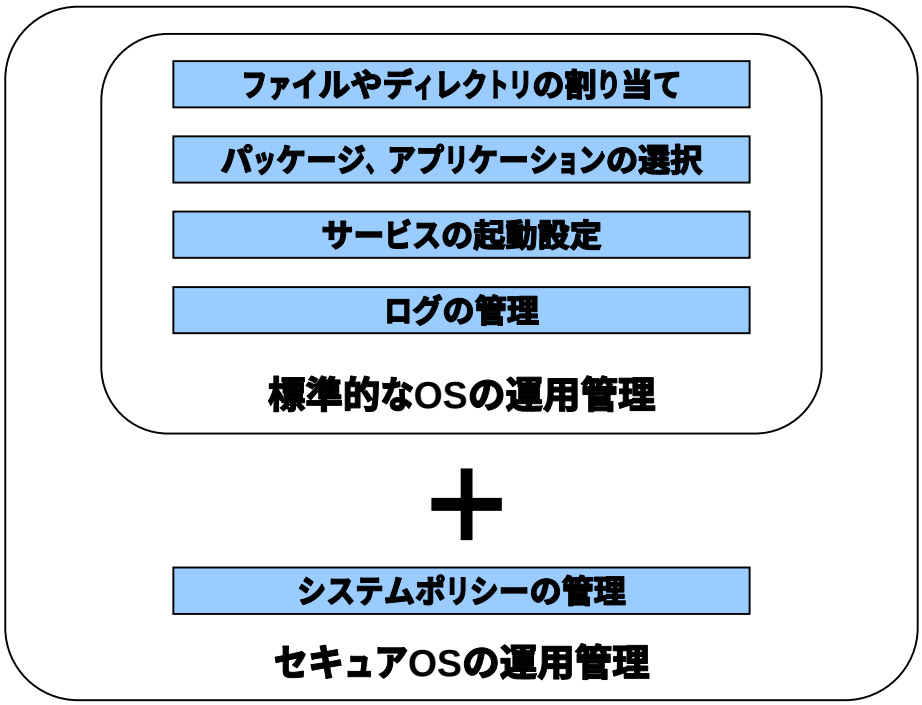


図　２－２　セキュア OS と一般的な OS との運用管理の違い

2.2 セキュア OS の設定管理と運用

セキュア OS の製品ごとに、設定管理・運用の方法や機能、用語などは若干異なっている。ここでは、いくつかのセキュア OS 製品を例にとって設定管理・運用の方法について紹介する。

2.2.1 ロール等の設定

多くのセキュア OS では、ユーザのアクセス権限を細かくコントロールする手段として、ユーザの役割に応じた「ロール」を定義し、ロール毎にアクセスできるデータやプログラムを制限する「ロールに基づくアクセス制御（Role-Based Access Control：RBAC）」機能を利用することができる。

この RBAC を利用するためのシステムポリシーの設定作業について、以下に SELinux の例を中心に示す。

（1）システムポリシーの定義

SELinux の場合、システムポリシーの定義ファイルは階層構造を持つテキストファイル群から構成される。該当する部分を、テキストエディタやサードパーティから提供されている設定ツール等を用いて編集する。

SELinux のアクセス制御機能では、システムポリシーを構成する各種ファイルにおいて、以下の各項目を設定することにより、プログラムの用途に応じて適切な役割を定義し、アクセス権限を変えることにより、システムをより安全に運用することができる。

- 各ユーザが、どの役割（ロール）を使うことができるか
- 各ロールが、どのプログラムを利用することができるか
- 各プログラムが、ファイルや通信先などに対してどのようなアクセス（読み込み、書き込み、実行等）をすることができるか

注）SELinux ではアクセス権限の記述を、プログラムを対象とする「ドメイン」、ファイルや通信先等を対象とした「タイプ」という 2 種類のラベルを用いて行う。ドメインとタイプによるアクセス制御の仕組みを、「Type Enforcement（TE）」と呼ぶ。

RBAC 機能は、このうちユーザとロールとプログラムの関係を制御する部分に相当する。SELinux の RBAC 機能は、OS 上で動作する個々のプログラムの単位で、権限の設定を行う点が特徴的である。ロールの定義と内容、それに基づく制御方法はセキュア OS の種類によって異なる。

（2）システムポリシーの机上での検証作業と運用方式の検討

システムポリシーによる情報保護の方針に誤りがないことを机上検証する。特に、運用時を想定して、実際の組織の役割と対応付けを行うことが重要である。

一部のセキュア OS 製品には、運用時の組織内部の不正を防ぐための仕組みとして「デュアルロック」と呼ばれる機能をサポートしているものがある。デュアルロック機能の下では、重要な設定変更を行う場合は、指定人数の管理者による操作が必要になる。（図 2－3参照）この場合でも、管理者同士が結託して不正を行うことが危惧されるが、不正が発覚した場合に先に罪を認めたものを免罪にする米国の司法取引のような仕組みを組み合わせることにより、抑止効果を図ることができる。

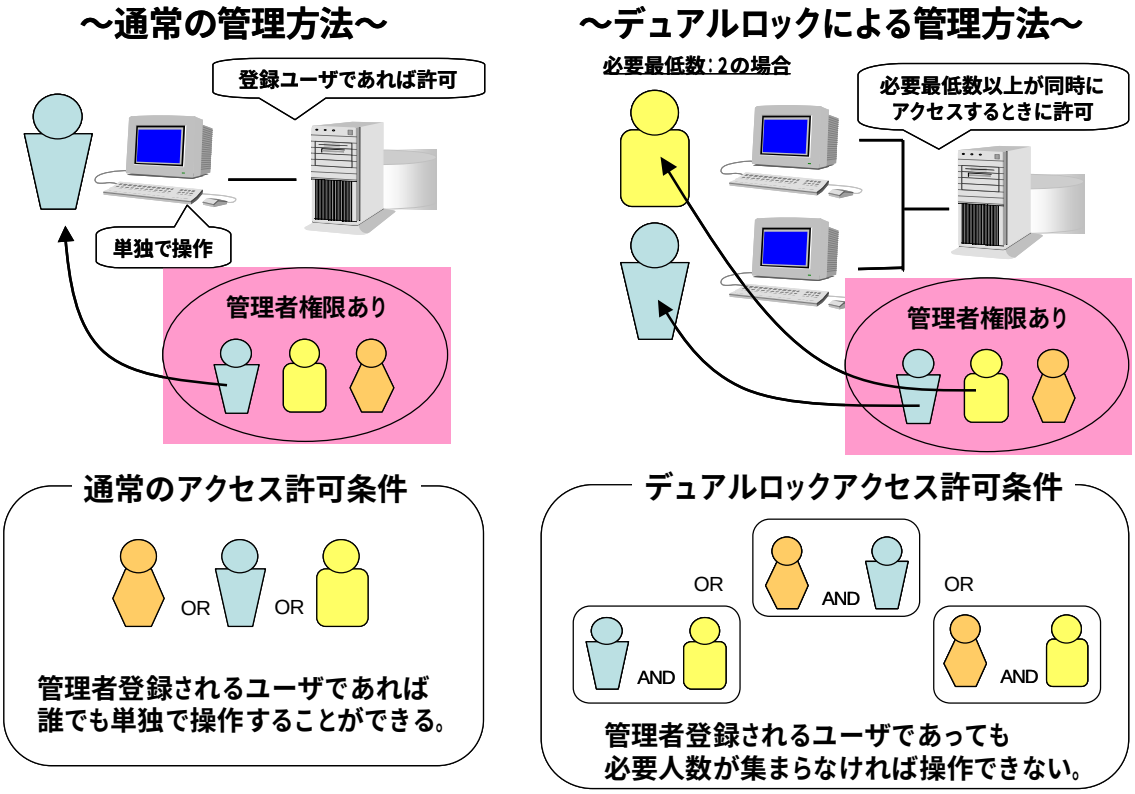


図 2－3　デュアルロック

（3）システムポリシーの最適化

システムポリシーは、一度で適切なものを策定できるとは限らない。実際の運用では、作成したシステムポリシーを適用して、OS やアプリケーションが正しく動作するかを確認しながら、システムポリシーを微調整する作業が必要となる。

ポリシーの調整には、監査ログを利用する。そのための具体的手順としては、OS のログとして出力されるポリシー違反（ユーザのアクセス要求がシステムポリシーに合致しなかった場合に出力されるログ）の内容を吟味し、不足していたアクセス条件をポリシーとして追加することにな

る。ただし、やみくもにポリシー違反の内容をそのまま許容するようにしてしまうと、意識しないうちにシステムのセキュリティを低下させてしまう可能性がある。そのため、定期的にシステムポリシーの最適化を実行する作業が必要となる。

２.２.２ マルチレベルセキュリティの設定

Trusted Solaris 等、セキュア OS 製品の中には、上記の RBAC に加えて、「マルチレベルセキュリティ（Multi Level Security：MLS）」の機能を実装しているものがある。マルチレベルセキュリティとは、各ファイルを、例えば「機密」「極秘」「秘」「公開」などとレベル付けをして、各ユーザは自身に割り当てられたレベルよりも上／同等／下のファイルに対して行える操作を制御するものである。そして、その代表的なアクセス制御が、前述したベル・ラパデュラ（BLP）モデルである。以下に、Trusted Solaris におけるマルチレベルセキュリティの設定の例を示す。

（１）システムポリシーの定義

すべてのユーザに適用されるシステムポリシーの定義を行う。ここでは、管理者やユーザの権限を個々に定義するとともに、管理者による管理タスクや提供サービスのプログラムの実行に必要となるアクセス権限を定義する。通常、これらの作業は Solaris の管理ツールを用いて設定を行う。

（２）マルチレベルセキュリティのためのラベル定義

Trusted Solaris では、「ラベル（label）」と呼ばれる仕組みによってマルチレベルセキュリティのコントロールを行う。以下の 2 つの要素から構成される。

- classification： 階層的なセキュリティレベルを設定する（設定例：「TOP SECRET（機密）」、「UNCLASSIFIED（一般）」等、標準 4 種類だが最大 255 種類まで定義可能）
- compartment： データにアクセスするユーザグループを設定する（設定例：「engineering department」「multidisciplinary project team.」等、標準 11 種類だが実質無制限に追加可能）

ラベルの名前と関係の定義については、組織の構成やネットワークの構成に応じた設計と設定が必要となる。ラベルは運用時に頻繁に変更する性質のものではないが、組織やネットワークの変更に応じて、逐次見直しを行うことは必要である。

（３）マルチレベルセキュリティの定義

ファイル、ネットワーキング、および管理者やユーザについて、クリアランス（ユーザの持つ権限の上限）と最低限のラベル制限を定義する。

（4）サービスの最少特権およびマルチレベルの定義

インターネットサービス、データベース、その他のアプリケーションなどに対する、必要となる権限（付録 D 参照)、クリアランスと最低限のラベル制限を定義する。

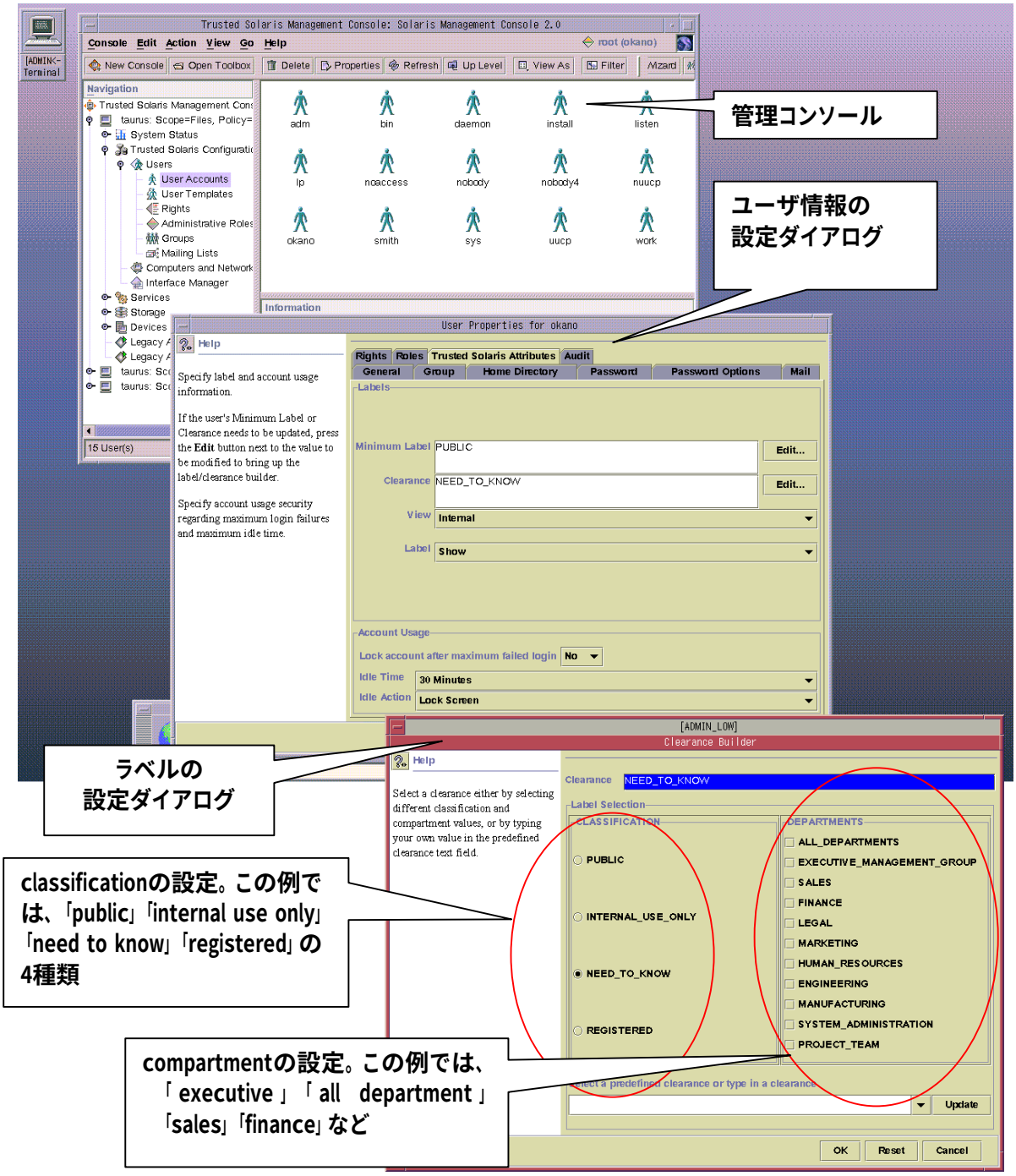


図 2－4　Solaris のシステムポリシー設定のイメージ

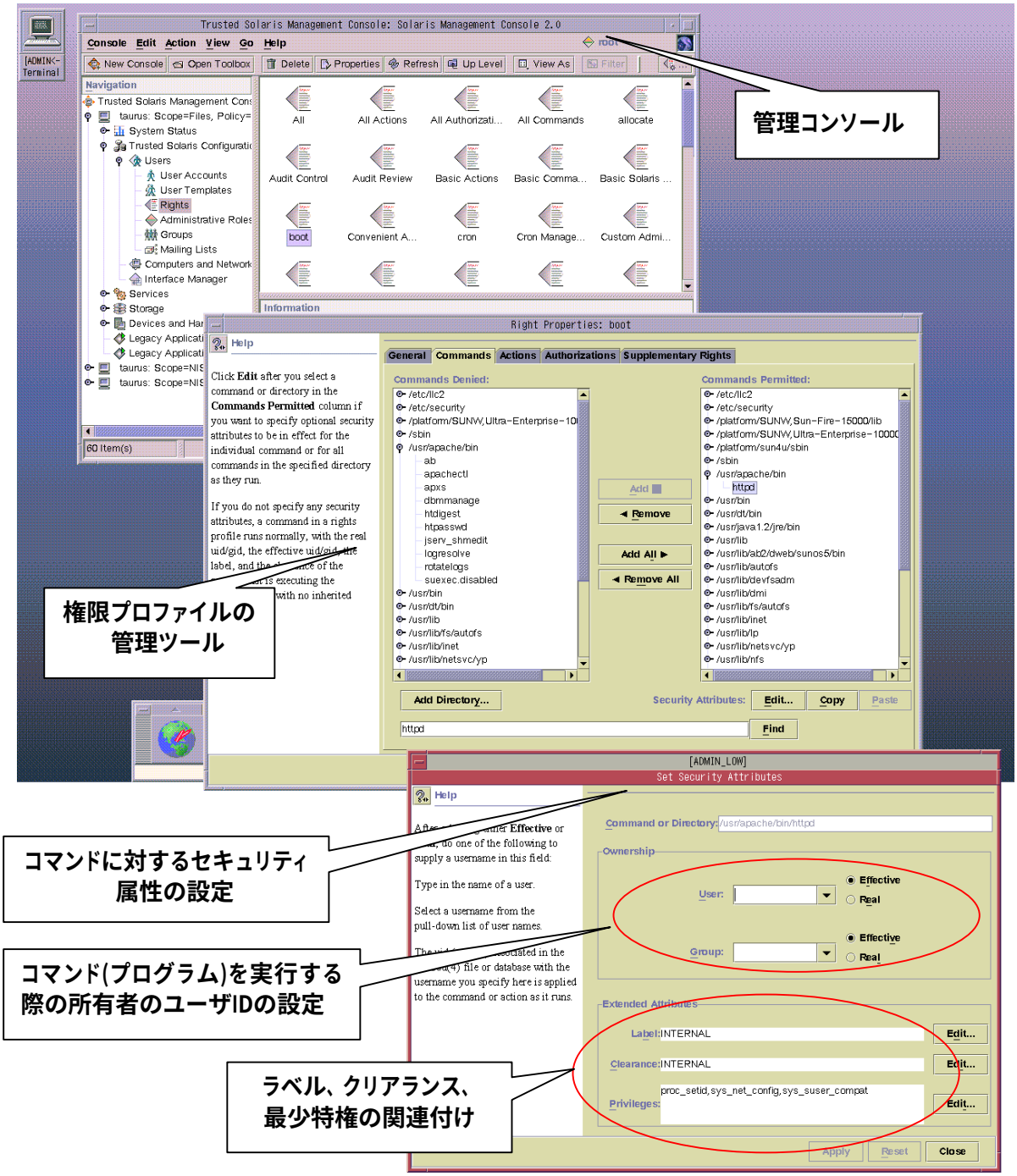


図 2－5　Solaris のシステムポリシー設定のイメージ（最少特権）

２.２.３ 支援ツールの利用

前項で述べたように、セキュア OS の設定管理、運用にあたっては、システムポリシーの運用管理が非常に重要になる。しかし一般の OS に比べればシステムポリシーの設定操作は複雑であり、1 つ 1 つ個別に設定していては業務効率が悪いだけでなく、全体的な設定状態の適切性を判断も難しい。そこで管理者が、業務を円滑かつ確実に遂行するための専用ツールが存在する。目的に応じた専用ツールを利用することで、以下のような効果が期待できる。

① システムポリシーの入力支援ツール

システムポリシーにおける制御の記述単位（粒度）はセキュア OS 製品毎に異なるが、通常の OS に比べると設定項目は広範多岐に渡る。ファイルやディレクトリに対するアクセス制御を例に取れば、GUI(Graphic User Interface）によってフォルダ階層をたどり設定するツールはシステムポリシーの設定をユーザフレンドリーにする効果がある。

② ログおよびシステムポリシーの比較表示ツール

システムポリシーを一通り設定し終えると、ポリシーが意図したように効果を上げているか確認することが重要である。管理者はログを監視して、システムポリシー違反を抽出し、設定されるシステムポリシーと比較することによって、設定の誤りがないか判断する。システムポリシー違反と現在設定されるシステムポリシーの状態を確認するための表示ツールにより、設定を効率よく行うことができる。

③ システムポリシーの解析・検証ツール

セキュア OS が備える機能が多いほど、その論理的な内容の検証を行うことは困難となる。システムポリシーの「正しさ」はあくまで最終的には管理者自身が判断すべきものであるが、解析・検証ツールは、自明な定義の誤り、矛盾した指定、本来不要である権限が有効な状態にある等を自動的に検知して管理者に警告するため、適切なシステムポリシーの設定に寄与する。

以上のようなツールについて、SELinux において実際に用いられているものの例を示す。

（1）アプリケーションの動作の追跡ツール

セキュア OS は、ポリシーの内容に基づき不要なアクセスを禁止し、それにより OS 自体やその上で動作するアプリケーションの不正利用を防ぐことができる。ポリシーの粒度が高ければそれだけきめこまかく制御を行うことが可能であるが、そのためにはアプリケーションプログラムの動作内容とそれに必要なアクセスを把握していることが前提となる。Apache のサーバーが html ファイルの置かれているディレクトリや設定ファイルを参照することが必要なことは容易に想像がつくが、CGI や動的に呼び出されるライブラリ、通信に必要なソケット等についての情報は標準の OS では意識しなくてすむものであり、セキュア OS の運用時にはそうした範囲を含めてアプリケーションプログラムの動作の追跡を行うツールが有効である。

図 2－6 は、米国の MITRE 社により開発された SELinux に付属するツールである polgen を用いてアプリケーションが参照する共有ライブラリを表示させた例である。

図 2－6　SELinux の polgen の出力画面

（2）ポリシーの検証ツール

セキュア OS を導入し、ポリシーを策定して運用を開始した場合、ポリシー違反が起こらなければ正しく運用されているかというと必ずしもそうではない。例えばポリシーの定義がほとんどのアクセスを認めるような内容であれば、そのアクセス範囲を利用してのシステムの乗っ取りやホームページ改ざん等が可能となってしまうであろう。また、ポリシー違反が生じた場合に、その違反の内容をそのままアクセス許可として追加してしまうと、同じ違反は確かに発生しなくなるであろうが、そのことにより間接的に認められるアクセス範囲が広がることが考えられる。このような観点から、定義されているポリシーの対象範囲と状態を確認することができ、またポリシーに修正を行う場合の影響範囲を確認できるようなツールが提供されている。

図 2－7 に、米国 Tresys Technology 社が提供している支援ツール setools（参考文献[4]）に含まれる apol を用いたポリシー解析の例を示す。

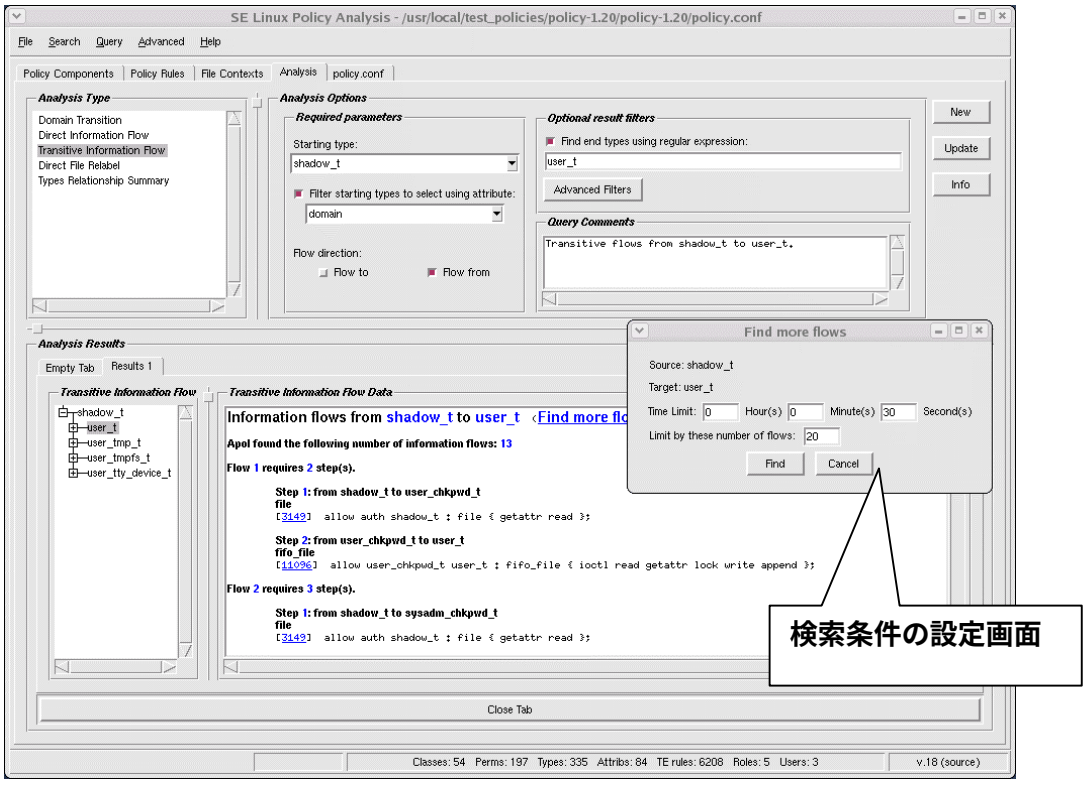


図 2－7　SELinux のポリシー解析ツール apol の画面

（3）ログ監視ツール

ログの監視は標準の OS の運用においても重要であるのは自明であるが、セキュア OS におけるログは不正なアクセスが行われているかどうかの判断基準として、また文書管理や決裁業務等提供するサービスに必要なアクセスが正しくポリシーに登録されているかの基準として特に重要な意味を持つ。そのため、標準的な OS のログ監視ツールに加えて、

- ポリシー違反に関するエラーを分類し、表示する
- システムの管理者がログを元に発生している事象の内容を解析するために必要な情報を提供する

ような機能を持つツールが提供されている。

図 2－8 に、米国 Tresys Technology 社が提供している支援ツール setools（参考文献[4]）に含まれる SeAudit を用いたログの解析の例を示す。

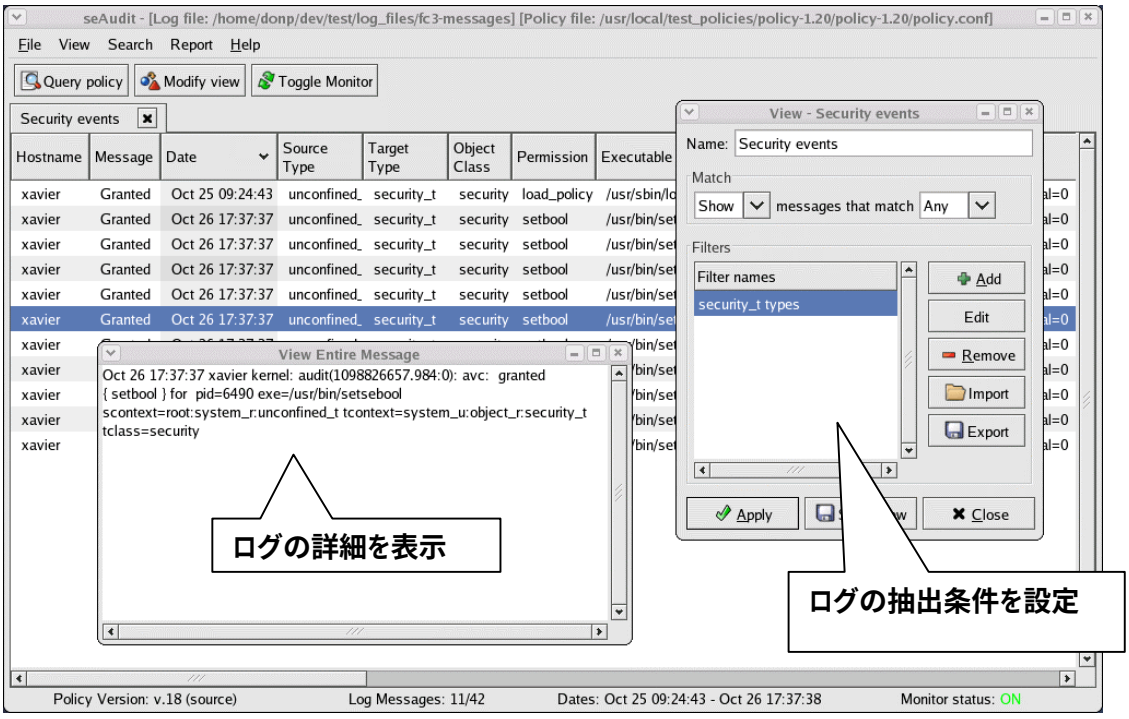


図 2－8　SELinux のログ表示ツール SeAudit の画面

（4）入力支援ツール

セキュア OS の種類にもよるが、セキュア OS においては一般の OS では考えられない多数の項目の設定が必要である。ポリシーの意味内容を検討、策定するのはシステム管理者であるが、その作業を支援するツールを活用することで、効率を高めることが可能となる。

図 2－9に、米国 Tresys Technology 社が提供している支援ツール setools（参考文献[4]）に含まれる SePCut を用いたポリシーのカスタマイズ操作の例を示す。

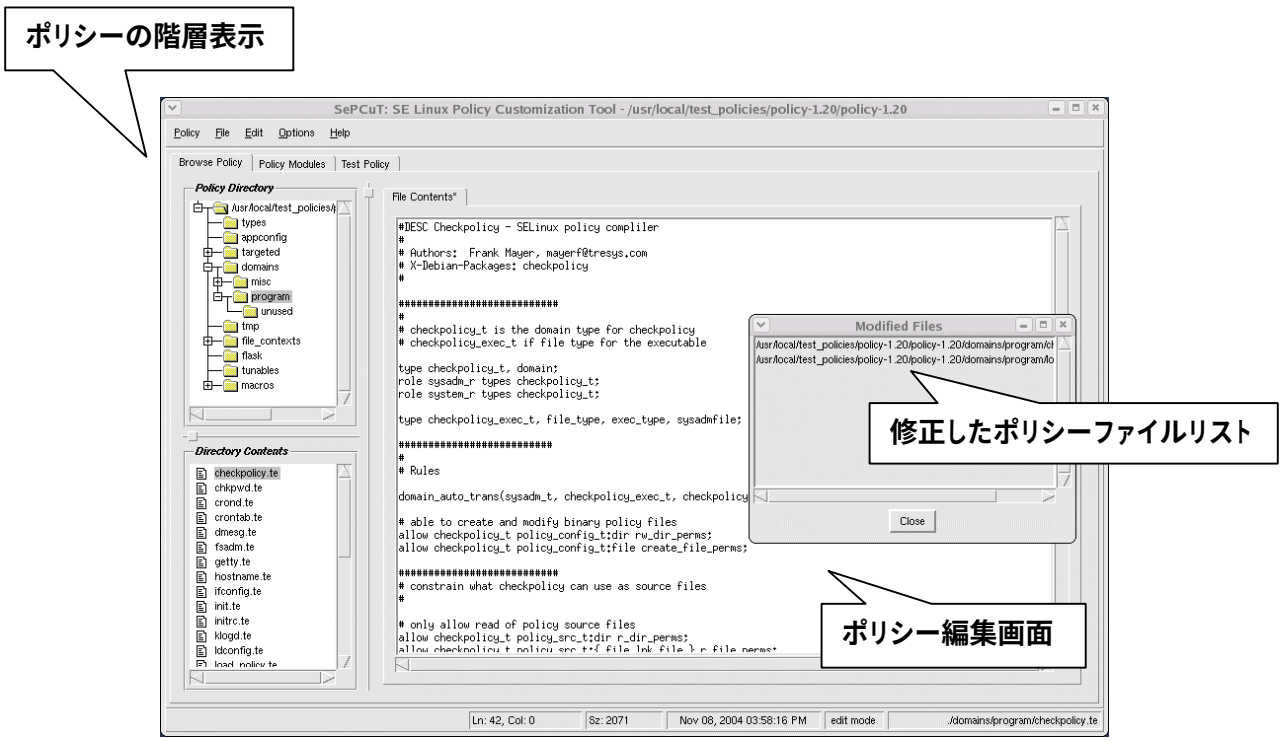


図 2－9　SELinux のポリシーカスタマイズツール SePCuT の画面

（5）運用ツール

セキュア OS の運用は基本的には管理者がコマンドラインで行うが、徐々に GUI 化が進みより効率的に作業が行えるようになりつつある。SELinux を組み込んだディストリビューションである Fedora Core 3 では、SELinux に関する設定がセキュリティレベルの設定メニューに含まれ、制御モードの表示、変更やポリシーの修正が GUI より行えるようになっている。図 2－9 にこうした GUI による設定画面の例を示す。

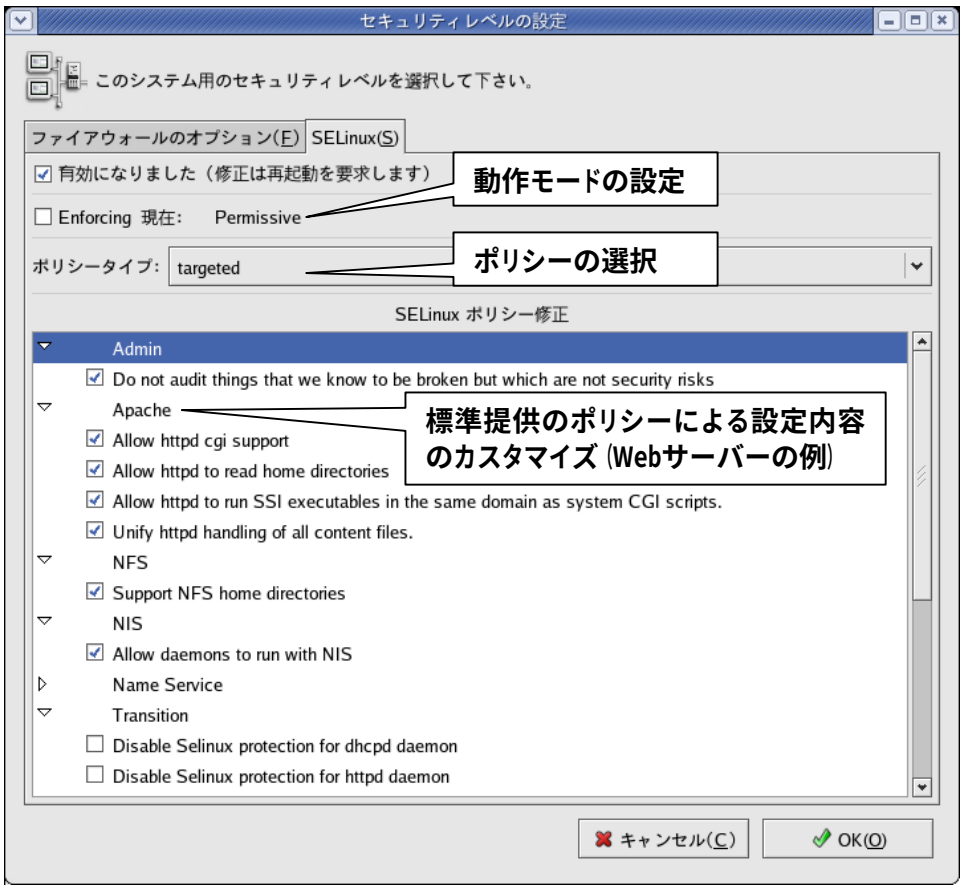


図 2－10　Fedora Core 3 の SELinux 管理画面

第３章 電子政府における各情報システムへの適用可能性

３.１ 電子政府における情報システムとそのセキュリティ

電子政府においては、e-Japan 重点計画に掲げられている行政運営の簡素化、効率化及び透明性の向上や国民の利便性の向上を目的として、情報システムの拡充が推進されている。以下にその主要例を示す。

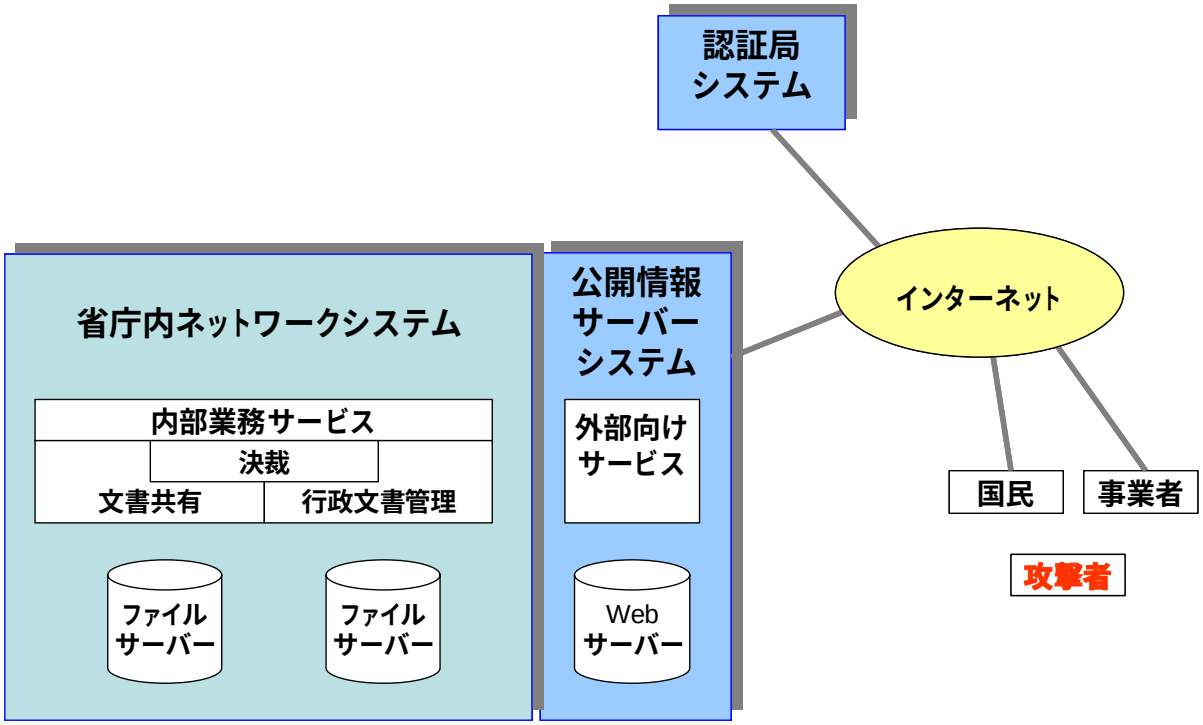


図 ３－１ 省庁における各情報システム

（１）公開情報サーバーシステム

国民の利便性の向上を目的とした公開情報サーバーシステムが構築されている。これによって、情報の電子的提供や申請・届出等手続および政府調達手続のオンライン化が図られており、「電子政府」プロジェクトの対象とされてきた。これらの外部向けサービスは、Web サーバーシステムとして構築されている。国民や事業者の利便性を向上する一方、攻撃者による脅威に曝されている。

（２）認証局システム

申請・届出等の手続および政府調達手続のオンライン化を図るために、PKI（Public Key Infrastructure：公開鍵暗号基盤）と呼ばれる技術を活用している。これは、公開鍵暗号技術をもとに、インターネット越しのユーザ（国民・事業者）についての本人認証や、ファイルやメール

についてなされるデジタル署名の検証などの社会的サービスを実現するものである。このようなサービスを実施するには、ネットワーク上に認証局と呼ばれる機構が必要となる。

我が国においては、公的な PKI サービスとして、GPKI（Government Public Key Infrastructure：政府認証基盤）が整備されており、各省庁は、それぞれの認証局システムを運用している（GPKI の詳細は、http://www.gpki.go.jp/ を参照）。このような認証局システムは、インターネット越しのユーザとの関係において、信用の基礎となるシステムであるので、アクセス制御を含め十分にセキュリティが確保されなければならない。

（3）文書管理システム

省庁における情報システムは、従前より、行政文書管理をはじめとする内部業務や庁内の情報共有を支援するものとして構築されてきた。このような省庁の行政業務を支援する典型的なシステムが文書管理システムである。行政文書管理や情報共有のためには、ファイルサーバーシステムをベースとして構築されることが想定される。職員の地位や権限に応じて、適切にアクセス制御が行われる必要がある。

これらのシステムは、利用対象範囲から見ればそれぞれ、省庁外部、外部と内部の双方、省庁内部に分類され、電子政府における情報システムの典型と考えることができる。

また、単に利用対象範囲だけでなくそれぞれサービスの性格が異なるので、セキュリティに関して想定される脅威も異なる。例えば、公開情報サーバーは公開という性質上、外部からの攻撃にさらされやすく、また文書管理システムは外部からの直接的な脅威は想定されないが、行政業務の中核をなし、機能がマヒした場合の影響が大きい。認証局は、両者の性格を兼ね備えている。したがって、それぞれのシステムについてのセキュリティの確保について、個々に論じるものとする。

なお、電子政府に用いられる情報システムのうち、ここでセキュア OS の適用可能性の検討対象として示した３つのシステムはいずれもサーバー的な役割を主体とするシステムである。クライアント的な役割を主体とするシステム（職員用 PC 等）にセキュア OS を適用することも考えられるが、以下の理由から当面はサーバー側を優先して扱うものとする。

- サーバーには重要情報が集中して保存されており、被害が生じた場合の影響が大きく、セキュア OS の適用等によるセキュリティ対策実施の優先度が高い。
- クライアント側の職員用 PC 等にセキュア OS を適用する場合は、既存資産の承継の必要性やユーザインタフェースなど、セキュリティ以外の要素が影響する面が大きい。

３.２ 公開情報サーバーシステム

公開情報サーバーシステムは、現在のところ例外なく Web サーバーシステムとして構築されている。ここでは、その Web サーバーにセキュア OS を導入することを検討する。

なお、ここでの検討結果は、省庁以外の Web サーバーシステムにおいても適用可能である。

３.２.１ Web サーバーシステム

公開情報サーバーシステムとしてのWebサーバーの基本的な機能は、単純である。ユーザ（国民・事業者等）がブラウザを用いて、Webサーバーシステムにコンテンツを求めるリクエストを送ると、OSを通じて当該コンテンツをブラウザに返すのが基本である。WebサーバーとOSを操作するのは、権限を有するシステム管理者に限られ、コンテンツについては、作成者と管理者が任命される必要がある。

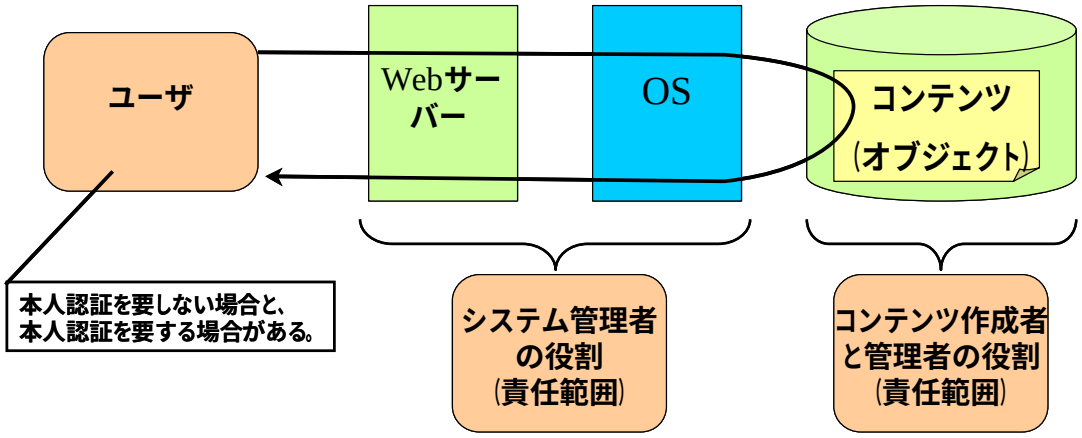


図 ３－２　Web サーバーをめぐる役割

また、電子政府における公開情報サーバーが提供する外部向けサービスには、ユーザ（国民・事業者等）について、本人認証を要するサービスがある。これを実現するために、Webサーバーは、ユーザについて本人認証機能を提供するための機能群を用意しているが、これらについては、OSによるアクセス制御とは独立して行われている形態が一般的である。

このような本人認証を要するサービスにおいては、ユーザが最初にアクセスする際にパスワード等の本人認証手続きが要求され、以降の画面へのアクセスは、Webサーバーシステム内で動作するアプリケーションプログラムによって管理されるセッションクッキー等に基づいて管理可能となっている。

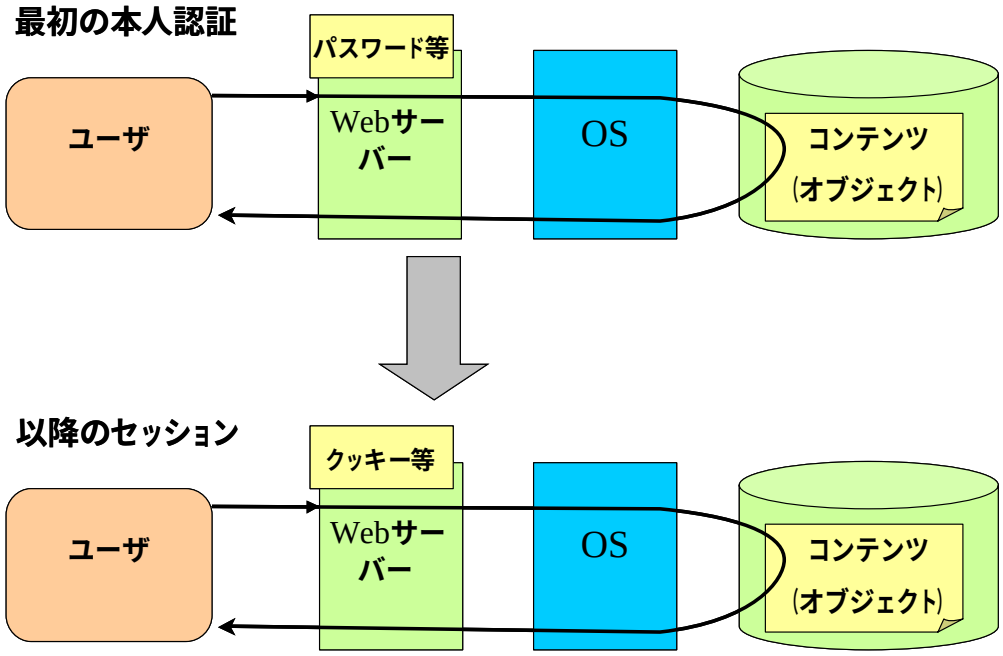


図　３－３　本人認証を要するサービス

このようなWebサーバーシステムについて、様々な脅威が想定される。次節では、代表的な脅威を説明する。

３．２．２ Web サーバーシステムに係る脅威

Webサーバーの基本的な部分については、次のような攻撃が脅威として想定される。

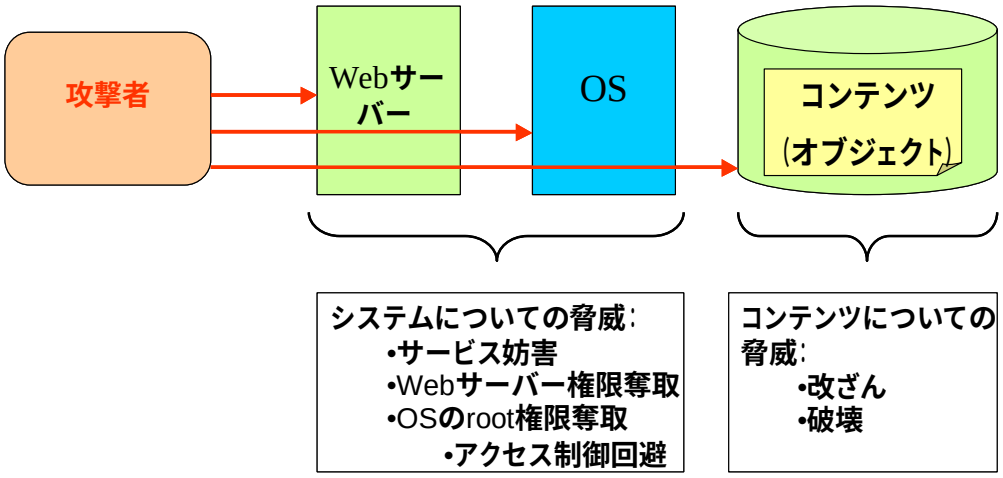


図　３－４　Web サーバーについての脅威

Webサーバーの基本的な部分についての典型的な攻撃事例を掲げる。

（1）攻撃事例 1

Web サーバーシステム内で、Web サーバー以外に、コンテンツの掲載等のために FTP（File Transfer Protocol）サーバーが動作するようにするシステムとなっているとする。その FTP サーバーのプログラムに「バッファオーバーフロー脆弱性」と呼ばれる脆弱性があり、その脆弱性を攻撃者に攻略されて、当該ホストの root 権限を奪取されてしまうことで、コンテンツを改ざんされたり、破壊されたりしてしまった。

（2）攻撃事例 2

攻撃者は、「広く使われている Web サーバーにサンプルとして付属している動的なコンテンツを生成するプログラムに脆弱性がある」という情報を得ていた。そこでその攻撃者が、ブラウザに表示されたフォーム（もしくは URL ボックス）から、その脆弱なプログラムにアクセスしつつ、あるコマンドを入力した。その結果、当該システムのパスワードファイルの内容がブラウザに表示された（過去に、Apache という Web サーバーにサンプルとして付属していた phf というプログラムに、このような脆弱性が存在していた）。

３．２．３ Web サーバーシステムへのセキュア OS の適用

「攻撃事例 1」に示したような攻撃に対しては、多くの強制アクセス制御機能を有するセキュア OS は有効である。この攻撃は、管理者権限（root）の万能な特権に起因している。通常の OS の場合、何らかの手段によって root が奪取されてしまうと、その特権によって、あらゆるアクセス権限の設定が無視できるようになってしまう。しかしセキュア OS の場合、このような万能な管理者権限を可能な限り分割していることから、その影響範囲を限定することができる

「攻撃事例 2」に示したような攻撃の可能性は、プログラムによるシステム資源のアクセスの目的が個別に考慮されていないことに起因している。このような攻撃に対しても有効なアクセス制御機能をもつセキュア OS が存在している。動作するプログラム単位の最小な粒度で厳密にアクセスを制御する設定ができるもの（例： Type Enforcement を実装しているもの）は、正確に設定すれば、このような攻撃にも対抗できるようになる。

すなわち、ユーザからのアクセスによって起動されるプログラムの権限は、管理者権限のような万能な特権ではなく、Web サーバーの通常処理動作に必要となるプログラムの実行に関するものに限定すれば、サーバー内部に新たな攻撃（通常処理では想定されないアクセス）を仕掛けることはできなくなる。

一般に入手可能なセキュア OS を用いて Web サーバーを設定することに関しては、一般的な OS 上で Web サーバーを動作させるように設定するときに要する知識以外の知識も必要となるので、設定の難易度は高まってしまう。

しかし、このような知識は、用途を設定すれば（例：Web サーバー）、ある程度、共通的に利用可能な知識であるため、ガイダンスやマニュアル類の整備が期待される。多人数のユーザのグループを想定する必要がないので、設定作業自体は、さほど複雑なものとはならないし、ひとたび誰かが設定テンプレートファイルを作成すれば、同じセキュア OS を用いるサイトにおいて再利用可能なものとなる。

３.３ 認証局システム

３.３.１ 認証局システムにおけるセキュリティの考え方

認証局システムは、インターネット越しのユーザとの関係において、ユーザ同士が「本人認証できるようにする基礎となっている情報」を管理している。この「本人認証できるようにする基礎となっている情報」とは、端的には公開鍵（Public Key）と秘密鍵（Private Key）のペアである。その片方である公開鍵は、広く公衆に示してこそ意味があるものであるが、もう一方の秘密鍵は、決して明らかにしてはならない。普及している PKI 技術においては、認証局が自身の存在を証明するために自らの公開鍵で署名した認証局自身のデジタル証明書を公開する一方、自らの秘密鍵のセキュリティを厳重に確保していることを示すために、その認証局が実施しているセキュリティ管理策を宣言することが一般的である。このセキュリティ管理策は、CPS（Certification Practices Statements：認証実施規定）という宣言書の中で明記することが慣行となっている。

認証局においては、自身の秘密鍵以外にも、厳格なセキュリティの確保を要する情報が管理される。例えば、失効したデジタル証明書についての情報である。このような情報が不正に改ざんされたり、破壊されたりした場合、人々がインターネット越しのユーザが本人であると信じる基礎が揺らいでしまう。そのため、インターネット越しの攻撃者による脅威のみならず、認証局の内部の要員による不正行為も防ぐことが求められる。すなわち認証局においては、秘密鍵の暴露や、失効したデジタル証明書についての情報の改ざん・破壊を防ぐためのセキュリティ対策が求められる。

３.３.２ 認証局システムに係る脅威

認証局システムは、次図に示すように公開情報サーバーシステムと、内部の情報システムから構成される。それぞれについて想定される脅威を示す。

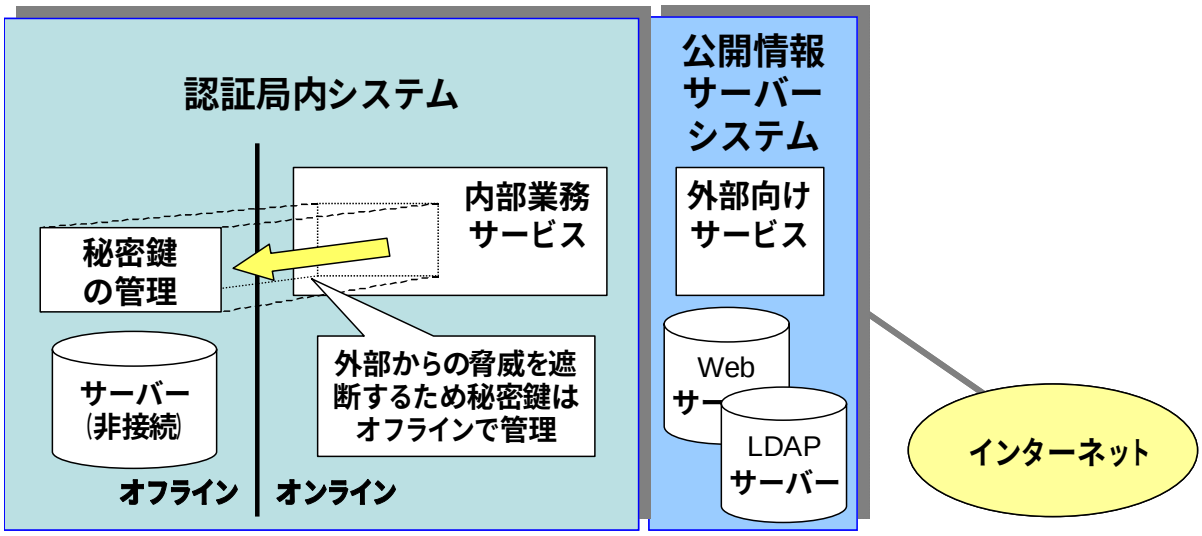


図 ３－５　認証局システム

（１）公開情報サーバーシステム

公開情報サーバーシステムの部分については、インターネットに対して公開されることが原則となることもあり、前節で検討した Web サーバーと同様の脅威が想定される。

このほか認証局システムの公開情報サーバーシステムにおいては、Web サーバーシステム以外に、LDAP（Lightweight Directory Access Protocol）サーバーシステム等も運用されることが多い。これは、失効したデジタル証明書についての情報を提供するために運用されるものである。このほか、デジタル証明書の有効性をユーザがオンラインで検証できるようにするために、OCSP（Online Certificate Status Protocol）サーバーや、SCVP（Simple Certificate Validation Protocol）サーバーも運用されることがある。こうした LDAP サーバー等についても、クッキー情報の漏洩など Web サーバー特有のものを除けば、Web サーバーとほぼ同様の脅威を想定する必要がある。

（２）認証局内システム

秘密鍵については、想定されるあらゆる脅威から遮断して保護することが求められるが、ネットワークに接続されている限り、外部からの攻撃から完全に遮断することは現実的には困難である。

そこでこのような秘密鍵を管理するシステムは、インターネットのような広域ネットワークに

は接続されていないオフラインなシステムとして配備されることが多い。オフラインなシステムについて想定される脅威としては、主として内部者による秘密鍵の改ざん・破壊が挙げられる。

３.３.３ 認証局システムへのセキュア OS の適用

秘密鍵を管理するシステムは、その被害発生時の影響の大きさから、セキュア OS を適用することが適するシステムであるといえる。

上述の公開情報サーバーの部分については、セキュア OS を適用することで、概ね前節の Web サーバーと同様の有効性を発揮することが期待できる。

これに対し、内部犯行が主たる脅威となるオフラインシステムの部分では、運用的なセキュリティ管理策として、「権限の分掌（separation of duties）」が行われることが望ましい。また、誰も単独では、あるシステム資源にアクセスできないようにして、その資源を防護する手順が、管理策として採用されることがある。これをシステムにおいて実装するには、デュアルロック（28 ページ参照）としてトラステッド OS が提供する機能を利用することができる。こうした OS の機能を運用面での対策と併用することで、管理策の有効性を高めることが可能になる。

３.４ 文書管理システム

本節では、電子政府の文書管理システムへのセキュア OS の適用可能性について検討を行う。省庁で用いられている文書管理規定を参考に文書管理業務をモデル化し（付録 E 参照)、これを電子化する場合の文書管理システムのモデルをについて検討した上で、予想されるセキュリティ上の脅威と、これに対処するためのセキュア OS の適用を検討する。

３.４.１ 文書管理システムのモデル

（1）文書管理業務モデル

情報公開法で定義される行政文書を対象としたライフサイクルモデルを、図 ３－６に示す。

全体は３つのフェーズに分類され、まず「決裁」フェーズでは、起案者により行政文書が起案され、一般に複数の決裁者がワークフローに従って決裁印を押印することにより、決裁が行われていく。

最終決裁者による決裁が終了した行政文書は、「管理」フェーズに移り、最終的な行政文書として登録され、「利用」フェーズに移行する。利用フェーズでは、指定された期間、内容が変更されることなく保存されると同時に、利活用のための閲覧が行われる。

利用フェーズの行政文書は必要に応じて管理フェーズに戻され、統合、分割、他課移管、廃棄、あるいは国立公文書館への移管等が行われる。

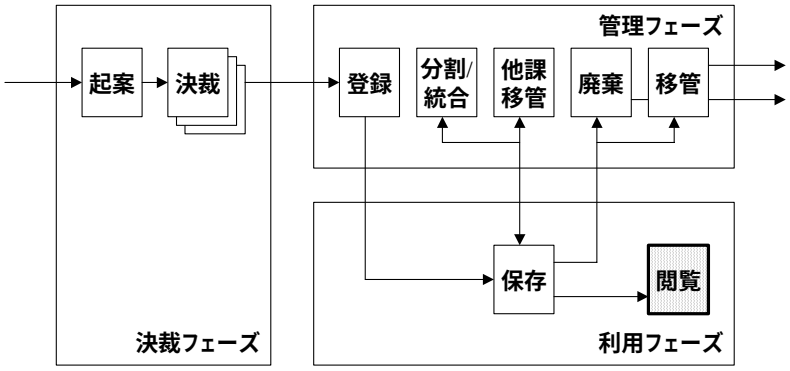


図 ３－６　行政文書のライフサイクルモデル

以下では、「利用」フェーズの中の「閲覧」処理に注目し、検討を行う。

閲覧処理においては、例えば極秘文書、秘文書といった文書の秘密区分の違いに応じて、閲覧できるユーザを限定する必要がある。すなわち、ある文書の閲覧を要求しているユーザが誰なのかを確認し(ユーザ認証)、その結果に基づき閲覧を許可/拒絶する(アクセス制御)必要がある。

（2）文書管理システムモデル

文書管理業務をシステム上で行う場合には、文書管理業務ロジックを実装したアプリケーションプログラムである「文書管理サーバー」と、その実行環境である「オペレーティングシステム(OS)」の２階層で実行される。

文書管理サーバーは OS が提供する基本機能を利用して実現されている。前述の極秘文書や秘文書のアクセス制御を行うために、文書の属性情報（例：秘密区分）を文書管理業務アプリケーションレベルで使用して、ユーザ認証やアクセス制御を文書管理サーバーが独自に行う製品が多い。一方、OS はそのための基本機能を提供するほか、文書管理サーバーとは独立したセキュリティ機能によって文書管理サーバーの振舞いを監視/制御し、また文書管理サーバーの業務データを保護している。このように文書管理システムのセキュリティには、文書管理サーバーが実施するセキュリティと OS が実施するセキュリティの２つのレベルがある。

閲覧サービスに着目した文書管理システムの実行時の制御モデルを図 ３－７に示した。図の上半分を文書管理サーバーが、下半分を OS がつかさどる。

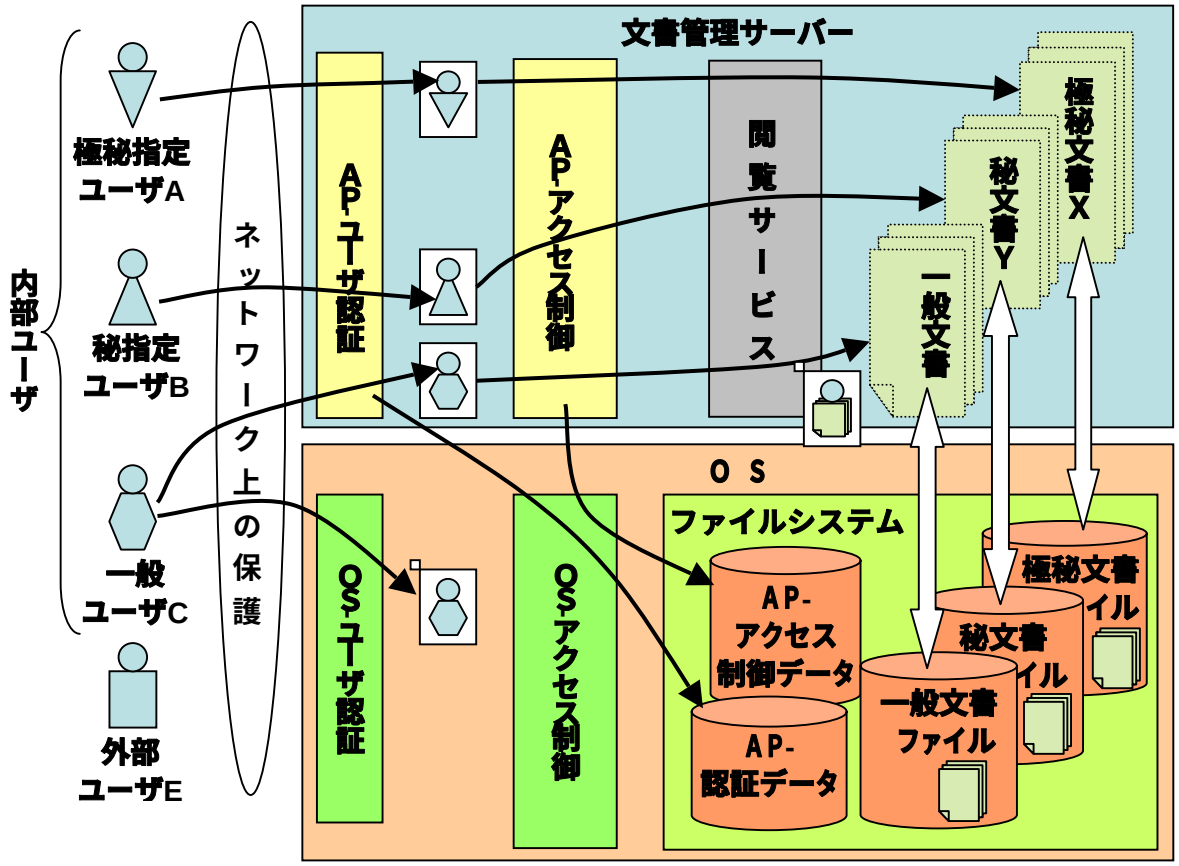


図 ３－７　閲覧に着目した文書管理システムの制御構造

文書管理サーバーはユーザに対して閲覧サービスを提供し、独自のユーザ認証（Application Program (AP)-認証）機能とアクセス制御（AP-アクセス制御）機能を備える。文書としては極秘指定ユーザのみが閲覧できる極秘文書、秘指定ユーザのみが閲覧できる秘文書、および閲覧制限のない一般文書がある。

各文書の実体は OS が提供するファイルシステムに文書ファイルとして格納されている。文書管理サーバーが AP-認証において使用する AP-認証データ(登録ユーザ一覧)や、AP-アクセス制御において使用する AP-アクセス制御データ(文書へのアクセスルール)も OS のファイルシステムに実体が格納されている。OS は OS としてのユーザ認証（OS-認証）機能とアクセス制御（OS-アクセス制御）機能を持っている。

ユーザには、極秘指定ユーザ、秘指定ユーザ、一般ユーザなどのシステム内部のユーザがいる。内部ユーザは文書管理システムのユーザとして文書管理システムに登録されるのが原則であるが、ここでは OS にも内部ユーザが登録されるケースを想定する。

文書管理サーバーも、一般のアプリケーションと同じように OS のアクセス制御のもとで稼動している。たとえば、文書管理サーバーが閲覧サービスを行うときは、文書ファイルから文書を取り出してユーザに提供するが、その際文書管理サーバーによる文書ファイルへのアクセスに対しては OS によるアクセス制御が行われる。

３.４.２ 文書管理システムに係る脅威

文書管理システムにおける典型的な脅威として、以下のものが考えられる（図 ３－８）。

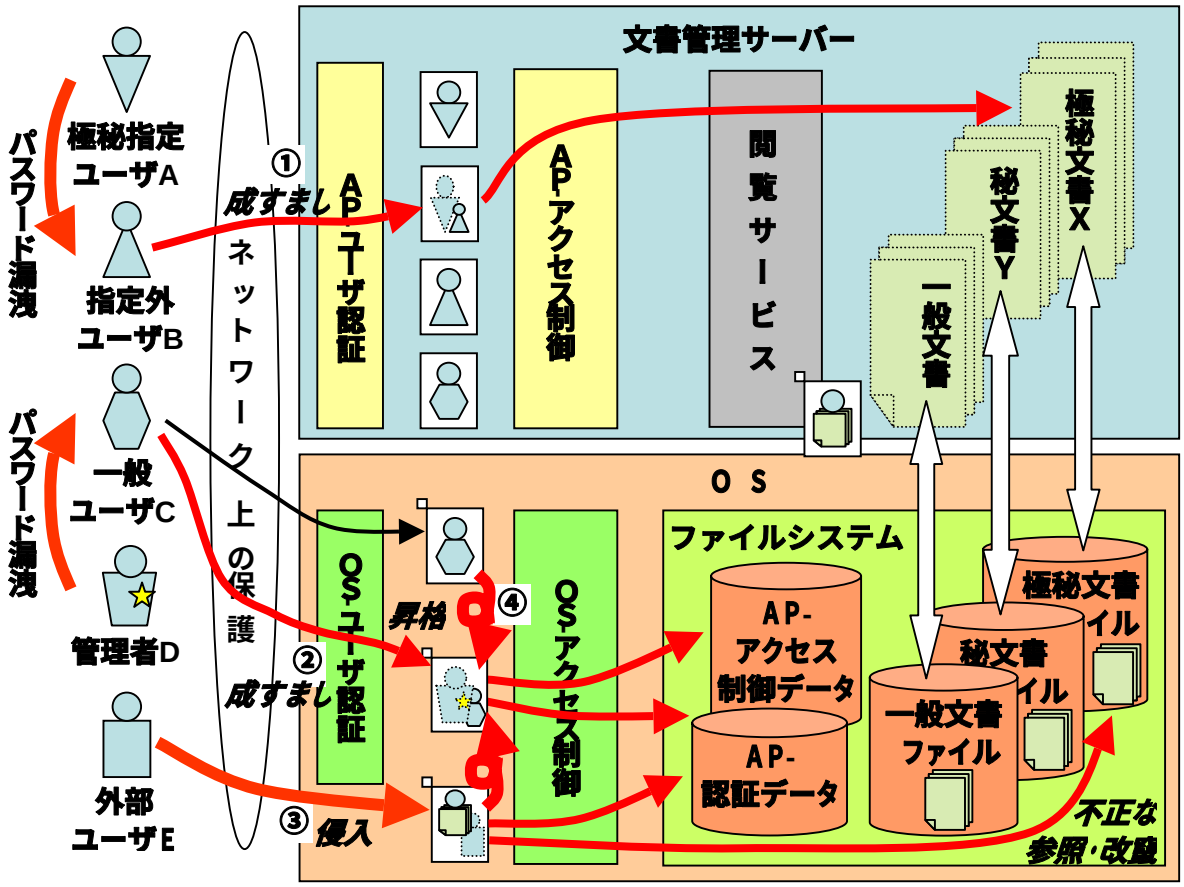


図 ３－８　文書管理システムにおける脅威

① 極秘指定ユーザ A の文書管理システム用のパスワードが極秘指定のないユーザ B に漏洩した。この結果、ユーザ B は極秘指定ユーザ A に成りすまして文書管理システムの利用に成功し、極秘文書 X へのアクセスが可能となった。

② システム管理者 D の OS 用のパスワードが一般ユーザ C に漏洩した。この結果、一般ユーザ C はシステム管理者 D に成りすまして OS へのログインに成功し、文書管理システムの管理用ファイルや文書ファイルに、文書管理サーバーを介さずに直接アクセスすることが可能となった。

③ 外部ユーザがバッファ・オーバーフローなどの手段でいずれかの（文書管理サーバーのもの

とは限らない）プログラムの実行権限の奪取（権限の不正取得）に成功した。この結果、

・奪取した実行権限が文書管理システムのものであれば、文書管理システムのファイルにアクセスできるようになる

・奪取した実行権限がシステム管理者のものであれば、文書管理システムのファイルにアクセスできるようになる

・奪取した実行権限が上記以外でも、次項に述べる権限昇格に成功すれば、文書管理システムのファイルにアクセスできるようになる

④ 一般ユーザ C は OS にログインした後、セキュリティホールを悪用してシステム管理者権限を奪取（権限昇格）した。この結果、一般ユーザ C は文書管理システムのファイル（文書管理システムのセキュリティ管理ファイルを含む）にアクセス可能となった。

３.４.３ 脅威に対する対策

前項にあげた脅威は、「ユーザ認証に関するもの」、「権限のないアクセスに関するもの」に分類できる。

（1）認証に関する脅威への対策のアプローチ

脅威の ①、② は認証の突破がそもそもの問題の始まりであることから、第一に認証メカニズムの強化が必要である。パスワードが漏洩し、成りすましてログインされた場合は、文書管理サーバーや OS には正規のユーザと区別する確実な手段はない。また文書管理システムでどんなに厳重にアクセス権を管理していても、一般的な OS の場合、管理者権限が取られると、すべてのファイルに改竄、破壊の脅威があることに留意する必要がある。

- 被害の極小化

  認証は突破される可能性があることを想定すべきであり、その場合の影響を局限化、限定化し、被害拡大の防止を図ることが重要である。そのために、正当なユーザであっても、元々の権限を大幅に制限しておくことで、被害をその権限の範囲内に収めることが現実的なアプローチとして有効である。このような権限の制限による被害の極小化は、セキュア OS を導入することによって取りうる対策の一つである。

（2）権限のないアクセス（不正アクセス）に対するアプローチ

脅威の③では、不正侵入の例として、バッファ・オーバーフロー攻撃によるサーバープログラムへの侵入を示している。第一章にあるように、バッファ・オーバーフローはプログラムの不良によるセキュリティホールを狙った攻撃方法であり、被害事例、警告事例が極めて多く、しかも後を絶たない。攻撃を受けるのは文書管理サーバーとは限らず、同時に稼動している他のプログラムが攻撃されることもあり得る。

攻撃者は侵入に成功すると、ログインした悪意のユーザ、あるいは侵入に成功した不正ユーザがシステム管理者の権限の奪取を試みる（脅威の④）。これにはパスワードの窃取・悪用、あるいはバッファ・オーバーフロー攻撃などが使われる。これらに対する対策方針は上に述べたとおりである。これらの攻撃に対しては、OS のセキュリティ強化が欠かせない。

（ａ）最少特権による制限

システム管理者の権限奪取に成功した不正ユーザによる不正アクセスに対する対策としては、「最少特権」の考えに基づいてシステム管理者に過大な権限を与えないことである。一般的な OS において、システム管理者は、「何でもできる」強大な権限を与えられることが多い。

それによって実装も運用もシンプルになるからである。しかしその権限が奪取されうるという状況下では、強大な権限の存在自体がシステムのセキュリティを脅かしている。そこで、システム管理者の権限を大幅に弱め、それでもなおシステム管理が可能であるようにすることが考えられる。これはセキュア OS によってのみ実行可能な対策である。

（b）権限昇格の制御

一般的な OS では、「権限昇格」によってシステム管理者権限（root など）を奪取され、すべてのコントロールが奪われてしまうリスクがある。権限昇格による攻撃は、攻撃者がバッファ・オーバーフロー等の脆弱性を悪用して一般ユーザ等の権限を奪って侵入したあと、さらにシステム管理者権限で動作しているプログラムの脆弱性を突くなどして、システム管理者権限（root など）を奪取する、間接的な侵入攻撃の手法に相当する。システム管理者権限が奪われると、すべてのコントロールが奪われてしまう。

権限昇格を制御する機能を備えた OS では、あらかじめ記述された権限以外への変更を一切禁止している。このため余計な権限昇格ができない形となっている。このように権限昇格の制御は、OS レベルでのセキュリティ強化として重要である。

第４章 セキュア OS の適用形態

前章では、当面電子政府の中でセキュア OS の適用可能性のある情報システムについて、代表例を取り上げて、その効果、問題点等について検討した。

特定の情報システムにおいてセキュア OS を適用する場合、その適用方法・形態によってセキュア OS 利用のメリットや課題も異なってくる。本章では適用形態別に、こうしたメリットや課題の比較を行い、それぞれの適用形態を評価することとする。セキュア OS の利用に際しては、十分これらの課題についても検討する必要がある。

４.１ セキュア OS 適用の検討

４.１.１ セキュア OS 適用モデル

セキュアOSの適用形態を分類するにあたって、一般的な情報システムを①アプリケーション、②ミドルウェア、③オペレーティングシステム、④ハードウェアの４階層に分類し、モデル化する（図 ４－１）。

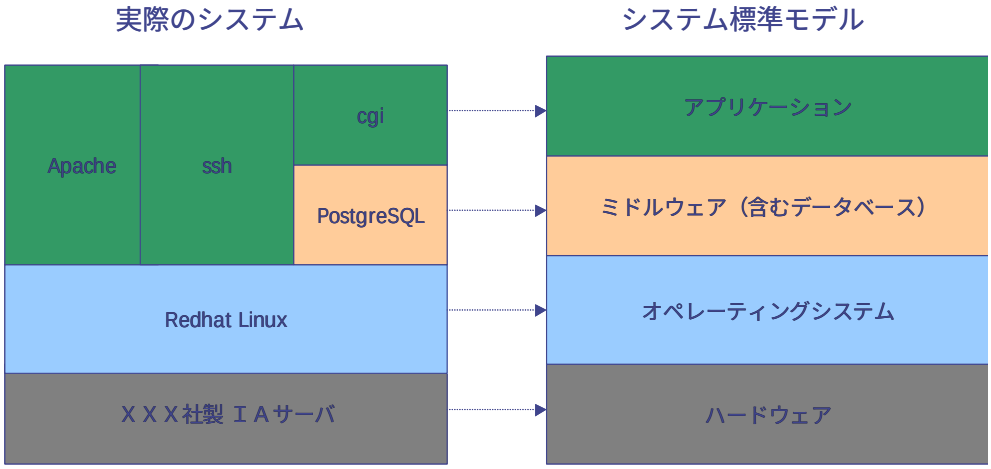


図 ４－１　システム標準モデル

このような情報システムにおいて、セキュア OS の適用形態を以下の 3 つにモデル化して、次節の評価項目にそって、それぞれの評価を行う。

（１）セキュア OS 適用モデル①

- OS のみのセキュリティ強化

（２）セキュア OS 適用モデル②

- OS・ミドルウェア・アプリケーションを個別に強化

（3）セキュア OS 適用モデル③

- セキュア OS の機能を利用したミドルウェアとアプリケーションの強化

4.1.2 セキュア OS 利用形態の評価軸

適用形態の評価軸としてここでは、以下の 5 項目を取り上げる。セキュア OS 利用時には、少なくともここで示した項目について利害得失を検討すべきであろう。

（1）適用効果

セキュア OS を導入することによって、どのような効果を得ることができるのか検討することが重要である。導入の方法や適用する分野によってもその効果は変わってくる。

（2）適用の問題

セキュア OS の適用によって生じる問題点についても利用時には十分考慮しておく必要がある。セキュア OS を導入する形態によっても問題点はかわってくる。セキュア OS に限らずセキュリティ製品は、使いやすさなどとトレードオフの関係にあり、問題点を十分に把握することは、セキュア OS の正しい適用に際して非常に重要な考慮点となってくる。

（3）運用への影響

セキュア OS の中には、これまでのシステム運用のスタイルを変更する必要があるものもある。セキュリティを高めるために運用方法そのものを変更しなければならないのであれば、その影響範囲がどこまでなのかを明確に把握しておくことが重要である。

（4）業務への影響

セキュア OS を適用する業務によっては、その影響が大きなものも予想される。セキュア OS 導入によって業務の再構築を行わなければならないケースなども想定され、業務への影響を十分検討しておくことが重要となる。

（5）維持管理作業への影響

設定などを維持管理していく作業は、セキュア OS の導入によってこれまで以上に工数がかかるものとなる可能性がある。また維持管理作業はこれまでのようにシステム管理者だけが担当すべき項目ではなく、セキュリティ管理者など新しい管理者が必要となる。

４.１.３ セキュア OS 適用モデル①　（OS のセキュリティ強化）

この適用モデルは、アプリケーションやミドルウェアで利用しているデータベースなどは、そのままで、OS の部分のみをセキュア OS に入れ換えるものである。一般的にセキュア OS が利用される際にとられる利用形態で、Web サーバーやファイルサーバーなどにおいて利用される形態である。

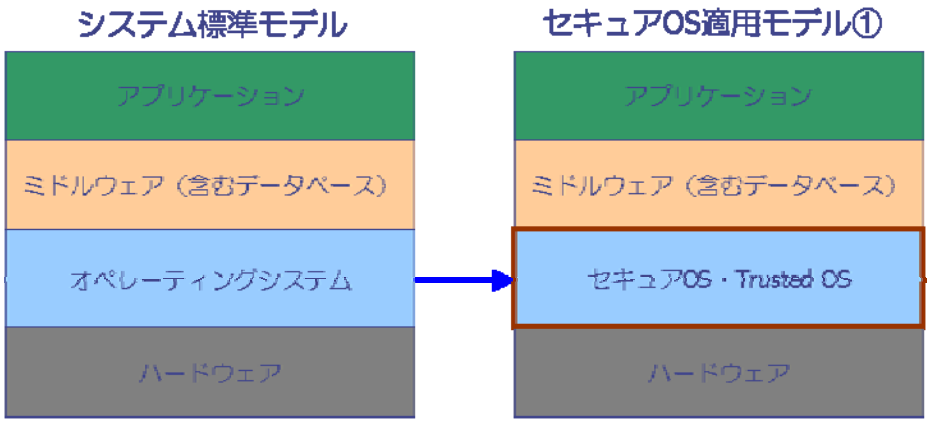


図　４－２　セキュア OS 適用モデル①

（１）適用効果

本モデルの適用による効果としては、以下の各点が挙げられる。

① セキュア OS 上で動作する各アプリケーションを、それぞれ独立した形に保つことができる
② システムの乗っ取り、踏み台化を行いにくくすることが可能である
③ 個々のアプリケーションに脆弱性があったとしても、システム全体が乗っ取られるようなことはない（実装不具合による事故の防止）
④ 個別にアプリケーションのセキュリティ強化を図らなくても、ファイルやプログラムといった OS の管理するリソースの単位では、強固なセキュリティが提供される
⑤ バッファ・オーバーフロー攻撃などによる被害を局限化することができる
⑥ アプリケーションにおいて認証やアクセス制御を迂回しづらくすることができる
⑦ アプリケーションそのものの改ざんを防止することができる
⑧ 業務上重要な情報の保護などの観点から詳細なアクセス履歴の保存が可能であり、信頼性が向上する

（２）適用の問題

本モデル適用による問題としては、以下の各点が挙げられる。

① アプリケーションを矛盾なく動作させるように設定することが煩雑であるものが多い
② 場合によっては（少数ではあるが）動作しないアプリケーションも存在する

③ アクセス制御にオーバヘッドがかかるため、多少パフォーマンスが低下する
④ OS とは独立してアプリケーション内だけで実行されるアクセス制御を強化することはできない
⑤ アプリケーション特有のデータを個別には保護できない

（3）運用への影響

本モデル適用による運用面への影響としては、以下の各点が挙げられる。

① セキュア OS によるアクセス制御のための、システムポリシーの管理運用が必要となる
② システムポリシー定義のために不可欠な各種権限の整理（分解）が必要となる
③ 専用ツールでこれらの作業を支援するものが多く、完成度の高いツールを導入することで、作業量を軽減できる可能性がある

（4）業務面と維持管理作業への影響

本モデル適用による業務面及び維持管理作業への影響としては、以下の各点が挙げられる。

① システムに対して利用者がどのような役割（分類）を与えるのかを、業務要件に基づき検討する必要がある
② 商用の製品などで提供される一般的な形式で、システムインテグレータやベンダーなどのサポートを受けやすく、導入事例も他に比べると多い

４.１.４ セキュア OS 適用モデル②　（OS、ミドル、アプリ強化）

この適用モデルは、OS の部分をセキュア OS にて入れ換えるとともに、アプリケーションやミドルウェアをセキュア OS の機能を利用するのではなく、アプリケーションごとにセキュリティを強化したものに入れ換えるものである。①のセキュア OS のみを利用する形態以上に高いセキュリティが求められる場合に利用が検討される。非常に強固なセキュリティが求められるデータベースなどでは、OS をセキュア OS に入れ換えるだけでなく、データベースも内部に強固なアクセス制御が施されたものが利用される（例えば、Oracle 社などはこのような利用方式が可能な製品を提供している）。しかしながら、強固なアクセス制御が実装されたアプリケーションは一般的には少ない。

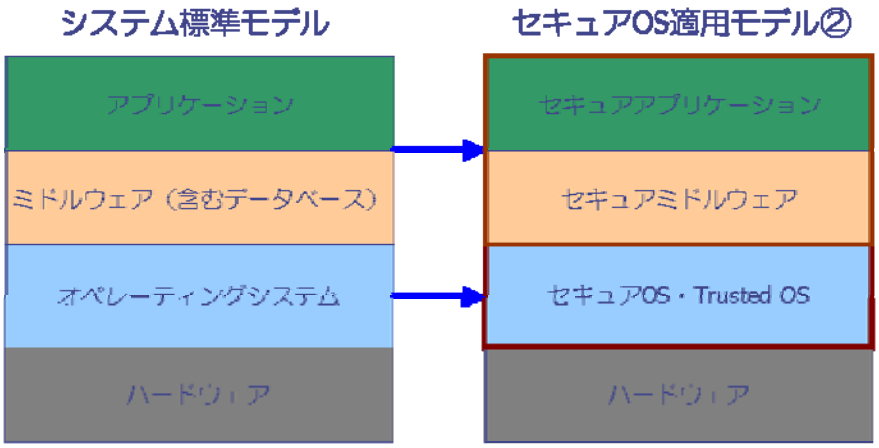


図　４－３　セキュア OS 適用モデル②

（１）適用効果

本モデルの適用による効果としては、以下の各点が挙げられる。

① （セキュリティ強化モデル①の適用効果と同様）

② アプリケーション内部で厳密かつ統一的なアクセス制御を実施することができる（OS では管理できない細かな単位（たとえばファイルの一部）で制御が可能）

③ アプリケーションのアクセス制御は OS のような抽象的なレベルでの制御ではないため、管理者が管理しやすい

④ 設定が困難なセキュア OS を一般的な設定にとどめ、運用上頻繁な更新が必要なアクセス制御ルールに関しては、アプリケーション側で実装するなど柔軟な設定が可能

⑤ アプリケーションのデータを詳細に保護可能

（2）適用の問題

本モデル適用における問題としては、以下の各点が挙げられる。

① アプリケーションに厳密なアクセス制御機能を実装するにはコストがかかる

② アプリケーションに認証やアクセス制御に迂回可能な脆弱性があると、セキュリティ強化の意味がない

③ アプリケーション独自のアクセス制御の機能を構築するため、アプリケーションに互換性（利用法等）がない

（3）運用への影響

本モデル適用による運用面への影響としては、以下の各点が挙げられる。

① OS でのシステムポリシーの管理だけでなく、各レイヤー（ミドルウェア・アプリケーション）それぞれでのシステムポリシーの運用管理が必要

② 通常はアプリケーションレイヤーでも運用のためのツールが提供されるため、通常のアプリケーションよりも大幅に作業量が増加することはない

（4）業務面と維持管理作業への影響

本モデル適用による業務面及び維持管理作業への影響としては、以下の各点が挙げられる。

① アプリケーションでアクセス制御を実施する場合には、OS に比べて非常に柔軟で詳細なアクセス条件を設定可能であるため、業務に合わせた適切なアクセス制御を実施可能

② OS だけでなく、それぞれの階層（アプリケーション）でのシステムポリシーの見直しが発生する

③ それぞれのレイヤーでのシステムポリシーの統制が取れなくなる可能性がある

４.１.５ セキュア OS 適用モデル③　（セキュア OS 機能を利用した強化モデル）

この適用モデルは、OS の部分にセキュア OS を利用し、アプリケーションやミドルウェアをセキュア OS の機能を利用してセキュリティを強化したものに入れ換えるものである。たとえアプリケーションに脆弱性があっても、重要なデータやプログラムは、OS 側の機能で保護されるため非常に高いセキュリティを確保可能である。高い信頼性が求められるシステムへの適用が考えられる。

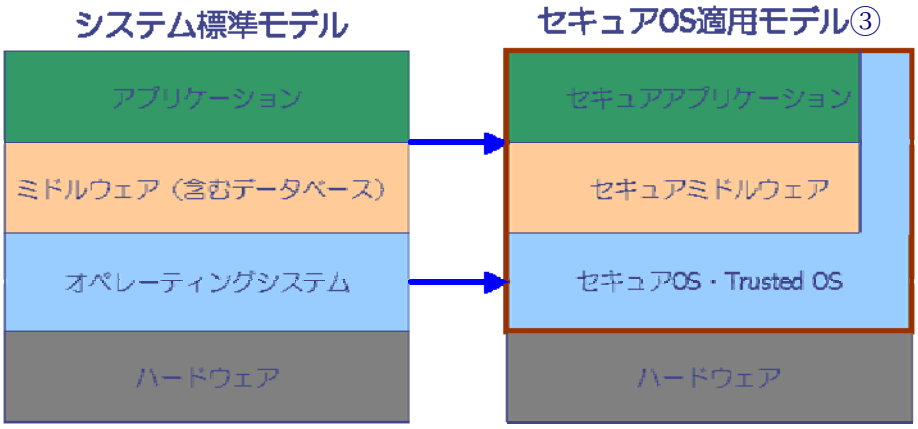


図　４－４　セキュア OS 適用モデル③

（１）適用効果

本モデルの適用による効果としては、以下の各点が挙げられる。

① （セキュリティ強化モデル①の適用効果と同様）

② アプリケーションはすべてのアクセス制御を OS に依存するため、統一的なアクセス制御が可能となる

③ たとえアプリケーションに脆弱性が存在しても、OS のセキュリティ機能で保護されているために被害が局限化できる

（２）適用時の問題点

本モデル適用における問題としては、以下の各点が挙げられる。

① すべてのアプリケーションが有する複雑な業務ロジックに関連する制御を、OS のアクセス制御機能で実現可能か不明

② 現状のセキュア OS では、アクセス制御の対象をファイルやプログラムといった単位で行うため、アクセス制御の粒度が粗くなりがちになる

③ すべてのアプリケーションのアクセスルールを OS 側で設定しなければならないので、設定が非常に煩雑となる（モデル①の場合をさらに上まわる）

④ セキュア OS の機能は標準化されていないため、セキュア OS 製品毎に異なるバージョンのアプリケーションを作らなければならないこともある
⑤ これまでのアプリケーション資産を継続して使えなくなる恐れがある

（3）運用への影響

本モデル適用による運用面への影響としては、以下の各点が挙げられる。

① OS レイヤーによる一貫したシステムポリシーの運用管理が可能
② すべてを OS の提供するアクセス制御などの機能によって定義しなければならないため、OS の運用管理者とアプリケーション側の運用管理とを切り分けるのが困難になる
③ ひとつの運用管理ツールなどでシステム全体の定義が可能となるため、個別に設定を定義管理する場合に比べて一度で全体の設定が可能になり、設定上の矛盾も生じにくくなる

（4）業務面と維持管理作業への影響

本モデル適用による業務面及び維持管理作業への影響としては、以下の各点が挙げられる。

① 非常に強固なセキュリティを確保した業務アプリケーションの構築が可能
② OS とアプリケーションのシステム管理を分けている場合は、OS 等のシステム管理者がアプリケーションに関する管理業務（ユーザ管理等）も実施しなければならない
③ 新たに、「セキュリティ管理者」のような業務にもシステムにも知見のある人材が要求される可能性がある

４.２ 電子政府モデルへのセキュア OS の適用可能性の検討

第 3 章で示した３種類の電子政府モデルを対象に、前節で挙げたモデル①～③の適用可能性について検討する。この検討結果を整理したものを、表 ４-１に示す。

４.２.１ 公開情報サーバーシステム

代表的な公開情報サーバーシステムである Web サーバーの場合、Web サーバー用アプリケーションは OS に付属するものを利用する場合が多く、セキュア OS を適用する場合の障害とはならないことから、モデル①の形態でセキュア OS を適用することは容易と考えられる。

また、Web サーバーと連携するデータベースや各種支援システムにおいて、それぞれのミドルウェアやアプリケーションにセキュリティ強化を行ったものがあれば、モデル②による適用も想定される。モデル②の場合はアクセス制御が多重化することが問題となるが、公開情報サーバーシステムの場合はユーザ数が限定されるため、アクセス制御対応による管理コストへの影響はさほど大きくならないものと見込まれる。

モデル③については、これに該当する Web アプリケーションが存在しないため現状では選択肢とならないが、将来的にセキュア OS のアクセス制御機能と連携するものが登場すれば適用の可能性は考えられる。

４.２.２ 認証局システム

認証局システムについては、外部に公開されていることで OS の脆弱性を突いた攻撃を受けやすいなど、公開情報サーバーと同様の要件を備えていることから、モデル①の適用は有効と考えられる。

モデル①を適用した上で、認証局用アプリケーションのセキュリティが強化された場合は、モデル②の適用により一層のセキュリティを高めることが可能となる。モデル③については、公開情報サーバーシステムと同様、これに該当する認証局用アプリケーションが現状では存在しないため選択肢となり得ない。しかし、認証局システムが不正アクセスを被った場合の影響が重大であることから、将来的にはセキュア OS のアクセス制御機能と連携するものが提供されることが望ましい。

４.２.３ 文書管理システム

文書管理システムの場合、システムにアクセスする利用者及び情報資産が多様かつ多数にわたることから、セキュリティを高めるためにはアクセス制御機能を強化することが重要となる。

文書管理システム用アプリケーションがセキュア OS に対応することで、モデル②により OS レベルも含めたセキュリティの向上が可能となる。ただし、文書管理システムの場合は対象とするユーザ数が多くなることが予想され、アプリケーションと OS とでそれぞれアクセス制御を行うことによる運用管理の負荷の増大は無視できない。

一方、極めて高いセキュリティを要求される分野においては、モデル③をもとに例えば OS においてベルラパデュラ（BLP）モデルを実装し、アプリケーションがこれをもとにアクセス制御を行うことが想定される。こうした実装にすることで、アプリケーションの中で OS とは別個にユーザを設定してアクセス制御を行うため OS のアクセス制御機能を活かせないといった従来のアプリケーションにおける課題を克服し、より実効的なアクセス制御機能を持ったアプリケーションを実現することができる。これにより、文書管理システム全体のセキュリティは著しく高まるほか、アクセス制御を OS で一元管理できるため、モデル②と比較した場合の運用管理の負荷は小さくなる。この方式は設計上 OS との緊密な連携が前提となるため、汎用的なアプリケーションにおいては実現されにくいが、専用のアプリケーションを用いて行っているような文書管理の場合は制約にはならず、導入の検討価値は高いと考えられる。

表 4-1　電子政府システムへのセキュア OS の適用可能性

セキュアOS適用モデル	適用の効果（○） 適用による問題点（×）	適用の影響 （運用・業務・維持管理）	適用の可能性の評価		
			公開情報サーバー	認証局	文書管理
モデル① セキュアOSを導入し、システムポリシーに基づきアクセスを制限	○ 乗っ取りの抑止ないし被害の局限化 × 設定が煩雑	•OS レベルでのシステムポリシーの管理運用が必要 •詳細なアクセス履歴の保存が可能	Webサーバーとしての実装例も多く、標準的な形態である。	OSレベルの脅威を軽減する効果を期待した適用は考えられる。	文書管理システムの特性を考慮すると、アプリケーションのセキュリティを強化せずにOSのみ強化することは考えにくい。
モデル② セキュアOSを導入した上で、ミドルウェアとアプリケーションを独自に（セキュアOSの機能を利用せずに）強化	○ 乗っ取りの抑止ないし被害の局限化 ○ アプリケーション単位での管理が容易 × 二重のアクセス制御のため管理コストが高い × 設定が煩雑	•OS、ミドルウェア、アプリケーションで多重の管理運用が必要 •柔軟かつ詳細なアクセス条件の設定が可能 •アプリケーション毎の維持管理が必要	電子申請等のサービスに用いるアプリケーションやミドルウェアにおいて、セキュリティを強化したものを利用可能な場合は、このモデルが適用可能である。	認証局用のアプリケーションとしてセキュリティを強化したものを利用可能な場合は、このモデルを適用できる。	3.4での議論を踏まえると、アプリケーションとOS双方でセキュリティを強化するこのモデル、または③で実装することが望ましい。
モデル③ OSの部分にセキュアOSを利用し、アプリケーションやミドルウェアをセキュアOSの機能を利用してセキュリティを強化したものに入れ換える	○ 乗っ取りの抑止ないし被害の局限化 ○ 統一的なアクセス制御が可能 × 設定が煩雑 × 複雑な業務ロジックにOS レベルのアクセス制御が対応できるかは不明 × アプリケーションによりOS の選択肢が限定される	•OS での一貫したアクセス制御による、多重の管理負荷の削減 •非常に強固な業務アプリケーションの構築が可能 •アプリケーションのアクセス制御と比較して粒度が粗くなりがち	将来的に公開情報サーバー用アプリケーションとしてセキュアOSの機能を利用するものが登場すれば可能性はあるが、Webサーバーとしては現在のところ見通しはない。	認証局システムが不正アクセスを被った場合の影響が重大であることから、将来的にはセキュアOSのアクセス制御機能と連携するものが提供されることが望ましい。	適切な設計により②よりも高い効果が期待でき、高度なセキュリティを要求されるなど、専用の文書管理システムにおいては今後適用可能性が高まることが予想される。

第５章 課題と展望

５.１ セキュア OS 活用のための検討課題

セキュア OS は近年注目を浴びてきているが、必ずしも普及が進んでいるとは言えない。セキュア OS が広く普及するためには、技術的な課題、ベンダー側、ユーザ側の問題など解決すべき課題があると考えられるが、主なものを以下に挙げる。

（１）商用アプリケーションの動作の制約

OS に期待される主要な機能のひとつは、様々なアプリケーションがその OS 上で実行できることである。セキュア OS がより一般的に利用されていくためには、その上で一般的な商用及びオープンソースのアプリケーションが動作することが重要である。そして単に動作するだけではなく、よりセキュリティを強化した形で動作することが求められる。セキュア OS における商用アプリケーションの動作は、セキュア OS そのものの課題ではないが、より広く利用されるようになるための課題といえる。

（２）ベンダーのサポート能力の不足

セキュア OS の管理・運用には高度な知識を必要とするため、サポートを提供する企業が限定され、製品によっては十分なサポートを受けることができるか不明なものもある。また、一部のセキュア OS では、運用管理のためのツールが整備されていないなどの問題もある。商用のセキュア OS では、設定用に利用しやすい GUI や簡単に設定を構築するための Wizard 機能などが整備されているものもあるが、それでもエンドユーザでの設定作業は困難であり、通常ベンダーのサポートを必要とする。しかしながら、システムの管理・運営はベンダーの支援を受けることがあっても、セキュリティの設定に関しては可能な限り運用主体において行えることが望ましい。そうした意味でさらにエンドユーザでも設定しやすいツールの登場が、今後のセキュア OS の導入範囲の拡大には欠かせない。

（３）オペレーション教育の不足

セキュア OS は、コストの低下や対応ベンダーの増加などの面で、近年非常に導入しやすいものとなってきている。しかしながら、従来の OS を利用している人が、すぐにセキュア OS を利用することは困難である。セキュア OS の導入（第２章）でも述べたが、セキュア OS はシステムポリシーに従って、構築されている。このポリシーを十分使いこなして設定しなければ十分なセキュリティは確保できない。こうしたポリシー構築・設定の担当者のためのオペレーション教育の充実や、その際に準拠すべきガイドラインの策定などが必要となる。

（4）支援ツールの整備

上記（3）で示した通り、セキュア OS を有効に機能させるためには予めシステムポリシーを適切に設定する必要があるが、これは煩雑な作業になりがちであり、現行のセキュア OS を導入することの阻害要因のひとつとなっている。２．２．３において示した支援ツールは、こうした設定の負荷を軽減させるものであり、今後さらに使いやすく、高機能なものが整備されることが期待される。またシステムポリシーの内容に問題があった場合、一般的にはポリシー違反のログからそれを知り、設定内容を確認した上で必要な修正を行い、確認の上システムに反映させるという流れになるが、こうした一連の作業をシームレスに確実に行うための運用時の統合的な操作環境があることが望ましい。

（5）バージョンアップに対する考え方の相違

一般の OS は非常に頻繁にバージョンアップが行われる。例えば Windows や Linux などは、月に何度もバージョンアップがなされることがある。大部分のセキュア OS は、一般的な OS と分類される特定の OS のカーネルコードを修正したものなので、この元になる OS のバージョンアップへの対応が問題となる。これまでは一般的に、セキュア OS のほうがパッチの提供間隔が長い傾向にあった。

セキュア OS は、もともとシステムに何らかの脆弱性があってもある程度セキュリティを確保できるように設計されているため、システムで提供しているサービスに重大な影響を与える脆弱性以外のパッチについては、パッチ適用のためにシステムを停止することによる不具合との比較を踏まえ、パッチ適用を遅らせることも考慮の余地がある。ただし、サービスに重大な影響を与える脆弱性に対応するためのパッチについては、セキュア OS であっても速やかに適用する必要があることは言うまでもない。

また、元になった OS のバージョンアップには、様々な機能向上が含まれている。したがって、今後はセキュア OS のベンダーがこうしたバージョンアップにタイムリーに対応していくことが望ましい。

（6）セキュリティ評価・認証制度への対応

100%セキュアなシステムは存在しない。システムの安全性とは、セキュリティを提供する機能に対してどこまで信頼できるかという問題である。提供されるセキュリティ機能に対して信頼性が高ければ高いほど、セキュリティのレベルは向上すると考えられる。こうした信頼性を評価するものとして ISO/IEC15408（CC: Common Criteria）によるセキュリティ評価・認証制度が存在する（付録 C（3）参照)。これはいかにセキュアかの度合いを評価するわけではなく、提供さ

れるセキュリティ機能の実現状況を外部的に評価するものである。今日開発されているセキュアOS には、現状ではこうした認証を取得しているものが少ない。しかしながらより安心してセキュア OS を利用するためには、こうした認証の取得は必須である。

（7）ポリシーモデル等の基礎的研究

わが国におけるセキュア OS の研究は活発とはいえない。研究対象となり得るテーマは、すでに米国で研究し尽くされているとの意見も聞かれる。セキュア OS の市場はこれまでニッチに近いものであったため、多くの企業はこうした研究に費用を割くことが現実的とは考えていない。大学等では、振る舞いの監視と制限による不正アクセスの阻止に関する研究などの実施例（参考文献[5][6]）があるものの、米国等に比べると本格的な研究は少ない。

しかしながら、実装を踏まえると、例えばセキュア OS 上のインターネットアプリケーションの実装方法や、インターネットアプリケーション保護に適したポリシーモデルにおけるベストプラクティスの検討など、セキュア OS を取り巻く分野において研究すべき課題は多い。

５．２ 検討の方向性

電子政府においては、既存のシステムの多くがクライアント・サーバー型システムであり、当面このアーキテクチャが存続すると考えられることから、サーバー側のセキュリティは引き続き重要な検討課題である。また、近年のネットワーク技術の急速な進展のもと、仮にクライアント側の機能の簡素化が促進され、いわゆるシンクライアントが導入されるような場合、サーバー側のセキュリティはますます重要になる。

一方、電子政府におけるデータ保護のニーズとして、軍事情報、外交情報、捜査情報等の秘匿性の高い情報はもとより、個人情報のように、その漏洩防止に万全の対策を講ずべき情報が多数存在していることから、今後とも秘匿性は重要視されなければならない。さらに秘匿性に加え、政府のホームページ改ざんのような事態を踏まえて、データの完全性も重要であることが広く認識されるようになっている。電子政府において、秘匿性だけでなく完全性も確保するためには、情報システムのセキュリティを OS レベルにおいても向上させる必要があり、セキュア OS を可能な限り導入することが有効であると考えられる。

しかしながら、実際にセキュア OS を電子政府に導入する場面においては、セキュリティの向上というメリットと同時に課題も伴う。例えば、セキュア OS 上で動作するアプリケーションが少ないことから、これまでの情報資産を継承する必要がある場合には、導入できるシステムが限定される可能性がある。加えて、セキュア OS の管理・運用は、一定の専門知識や付加的な業務が管理者に求められることから、人材の育成や業務への影響という観点からも検討が必要となる。

このような状況の中、電子政府のセキュリティを、可能な限り早期かつ広範囲において向上させるには、本報告書の検討を参考とし、その保有する情報資産の重要性と導入の容易性を総合的に勘案して、当面セキュア OS を導入すべきシステムをより具体的に検討することが必要である。そして、それらのシステムのセキュリティを確保するために、何をどのように守りたいのかをより具体的に検討し、そのシステムポリシーを確定することが求められる。これにより、アプリケーションやミドルウエア、とりわけ OS に求めるセキュリティ要件を決めることが可能となる。

最近の OS 製品には、従来セキュア OS に固有であったメカニズムを標準でサポートするものや、マルチレベルセキュリティを包含する新たな概念に基づく保護メカニズムを実装するものも現れており、OS レベルで実現可能なセキュリティ機能は拡大している。

このようなセキュア OS 製品には、政府支援のセキュア OS 開発も含まれている。具体的には、米国政府は 1960 年代から民間委託により、多くのセキュア OS を製品化していることは有名であるが、最近では韓国政府が 1998 年から、セキュア OS 開発を民間会社に委託し製品化を行っており、これらの製品は既に日本でも販売されている。またフランスでも 2004 年から３年間でセキ

ュア OS を開発、評価する計画である（付録 G 参照）。

今後、電子政府における OS のセキュリティ要件を検討する中で、既存の製品では電子政府が必要とするセキュリティの要件を満たさないことも想定される。この場合は、米国、韓国のセキュア OS、あるいはフランス政府によるセキュア OS の開発決定を参考にしつつ、わが国においても「セキュア OS」の開発を行うことも選択肢のひとつとして検討することとなろう。

付録

付録A OS とは何か

（1）基本ソフトとしての OS

OS とは、オペレーティングシステム（Operating System）の頭文字をとったもので、基本ソフトウェアともいう。コンピュータシステムは大別すれば、

① ハードウェア、例えば、ユーザがデータや命令を入力するためのキーボード、その入力されたデータにもとづいて演算をする CPU（中央処理装置）という装置、その演算結果を出力するための表示装置（ディスプレイ）やプリンター

② ソフトウェア 例えば、ワープロソフトや表計算ソフト、インターネットのためのブラウザ、電子メール用ソフトといったソフトウェア

から構成されている。

ソフトウェアをさらに区分すれば、いわば各種のソフトウェアの基盤、土台ともいうべき OS と、その他のソフトウェアに分かれる。

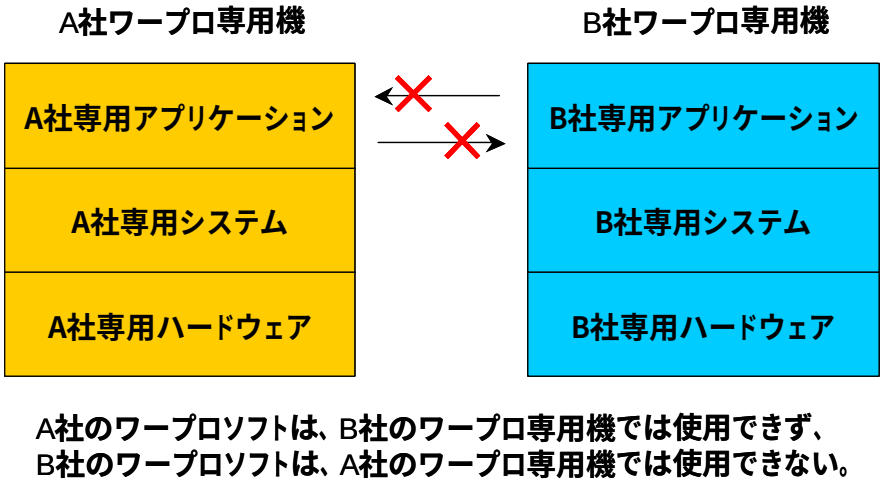
この OS という基盤、土台の上で動き、ユーザがコンピュータにさせたい仕事（文書の作成）を処理するための操作する対象となるソフトウェア（ワープロソフト）を、アプリケーションソフトウェア（応用ソフトウェア）あるいは単にアプリケーションと一般に呼んでいる。したがってこれとの対比で OS のことを、基本ソフトウェアと呼ぶことがある。

（2）OS の特徴的機能

かつて日本の職場や家庭に広く普及していたワープロ専用機と比較すると、OS の主要な特徴や機能を理解しやすいであろう。

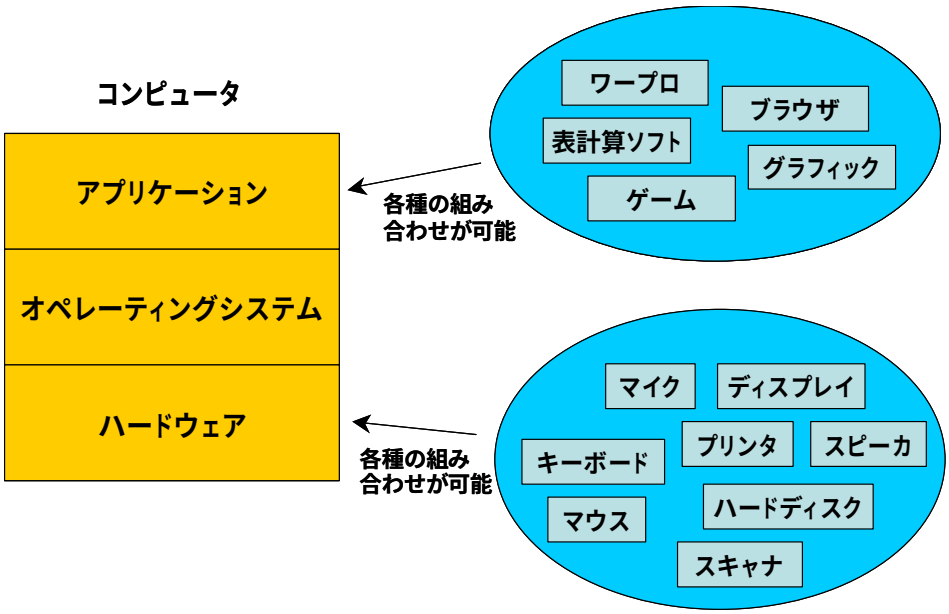
典型的なワープロ専用機の仕組みは、次のように説明できる。あるメーカーの特定のワープロ機種は、特定のワープロソフトだけを作動させることができる専用のハードウェア（演算装置、キーボード、ディスプレイ等）と専用システム（ソフト）から構成されていた（図Ａ－１）。

ハードウェアの点からは、コンピュータとかなり似ており、外見上はラップトップコンピュータやデスクトップコンピュータと大きな違いはなかった。しかし、演算装置や記憶装置の容量や性能は、特定のワープロソフトを動かすためだけのものと、コンピュータのものとでは格段の差があり、また周辺機器とよばれるものも、現代のコンピュータでは、入力装置のマウスや、外部記憶装置と呼ばれる CD-ROM ドライブや MO ディスク、ネットワーク機器としてモデムや LAN があり、ワープロ専用機と比べて多種多様なハードウェアから構成されている。



図Ａ－１　ワープロ専用機の構成

だが、コンピュータとワープロ専用機の最も大きな違いは、そのアプリケーションである。前述したように、ワープロ専用機は基本的にある特定のワープロソフトを動かすものであり、そのソフトしか動かすことができない。しかし、コンピュータは多種多様なアプリケーション（プログラム）を動かせるようにできている（図Ａ－２）。



図Ａ－２　コンピュータの構成

しかも、現代のコンピュータでは、ユーザは複数のアプリケーションを同時並行的に使って、各種の作業を連続的に行うことができるようになった。

例えば、ワープロソフトを使ってレポートを作成しているユーザは、同じコンピュータの画面上で、レポート作成に必要なデータや情報を得るためインターネットを利用してウェブサイトを閲覧したりデータを自分のコンピュータにダウンロードできる。また、レポート作成中に到着した電子メールを見ることができるし、作成したレポートを印刷しながらレポートの校正を同時に行うこともできる。

通常のコンピュータは、その頭脳ともいうべき CPU をひとつしか持っていない。それにもかかわらず、コンピュータがこのような複数の作業を同時並行的に行えるのは、OS がマルチプログラミングあるいはタイムシェアリングと呼ばれる機能を有しているからである。

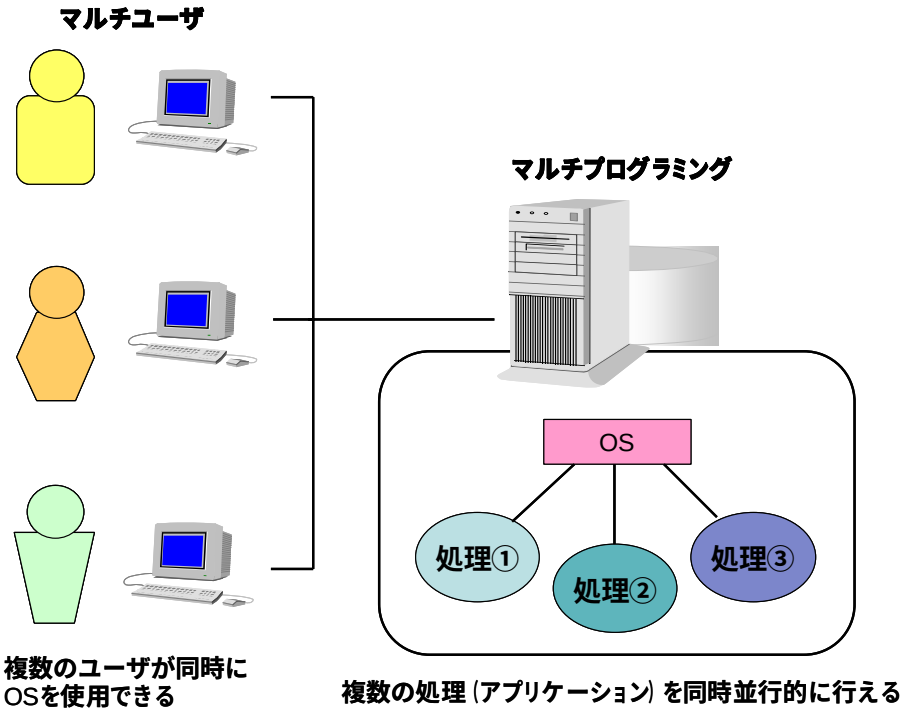
この機能は、簡単に説明すれば以下のようなものである。

アプリケーション A とアプリケーション B を使った一連の作業をコンピュータにさせる時、CPU がひとつしかないのだから、常識的には例えばアプリケーション A を先に動かし、その作業が終わったら次にアプリケーションＢを動かすこととなる。しかし、もしアプリケーション A は必ずしもその作業中のすべての時間を CPU に使っているわけではないとすれば、その CPU が空いている時間をアプリケーション B が使えば、その分だけ CPU が効率的に使えることとなる。

したがって、アプリケーション A とアプリケーション B による一連の作業がその分だけ時間が短縮できることとなる。

次に、ユーザの人数がワープロ専用機の場合はひとりだが、OS の場合原則として複数のユーザを前提としていることが指摘できる。現実には、ワープロ専用機を複数の職員や家族が共用することはあったであろう。しかし、ワープロ専用機は、複数のユーザを前提にしたシステムにはなっていなかったのである。これと同様に、初期のパーソナルコンピュータも、システムとしては複数のユーザを前提としないものが多かった。しかし、現在では、多数のパソコンに採用されている Windows XP という OS も、複数のユーザが使用することを前提に OS のシステムが構築されている。また、より大きなコンピュータ用の OS は、基本的に複数のユーザを前提にしており、例えば Unix や Windows Server 2003 といった OS は複数のユーザが同時にコンピュータを使用することを前提にしている OS である（図Ａ－３）。

これまで説明してきたことに基づいて OS の機能を要約すれば、「OS とは、CPU、メモリーと呼ばれる記憶装置、キーボードやプリンターといった入出力装置などのハードウェアを適正に管理制御することによって、複数のユーザが各種のアプリケーションを通じてコンピュータに行わせたい各々の作業全体を、効率的に進めることができるようにするための基本的なソフトウェア」だといえる。



図Ａ－３　複数のユーザによるコンピュータの使用

コンピュータの生産性を向上させるうえで重要な役割を果たしている OS であるが、このような OS によって作動するコンピュータには、ワープロ専用機では想定できないセキュリティ上の問題が発生することとなる。そもそもワープロ専用機は、個人専用であることを前提にしているのであるから、他人によって情報を盗み見られることはないし、典型的なワープロ専用機は記憶容量が少ないことから通常フロッピーにデータは保存するので、このフロッピーの管理さえしっかりしていれば、情報を他人に盗られることもない。

しかし、コンピュータでは複数のユーザが前提なのだから、あるユーザのデータが他のユーザによって盗み見られないような対策が必要であるし、そもそも悪意のないユーザであっても、例えばミスによって他のユーザのデータ等を破壊してしまうおそれがあるのだから、このようなことに対する防止策も考えなければならない。

また、ワープロ専用機であれば特定のワープロソフトしか使わないのであるから、このソフトが信頼の置けるものならば基本的にセキュリティ上の問題は生じない。ところが、コンピュータでは、さまざまなソフトを使用しており、場合によっては悪意のあるユーザによってセキュリティ上問題のあるソフトが使われることも想定され、これが他のソフトやデータに悪影響を及ぼすことに対する防御策を考える必要があるのである。

（3）OS の汎用性等

前述したように、コンピュータは複数のアプリケーションソフトを動作させることができる。このアプリケーションの種類が多ければ多いほど、そのコンピュータはいろいろな目的に使用できることとなり便利なものとなる。Windows の OS が広く使われている理由のひとつは、この使用できるアプリケーションが豊富だということと考えられる。

また、このアプリケーションを使って作成した文書等のデータも、OS が同じであればどのコンピュータでも使用できることとなる。したがって、多くのアプリケーションが使える OS が、データの共有という観点からもいよいよ便利だということになる。

さらに、このように便利な OS も、ある特定のメーカーが作ったコンピュータ（ハードウェア）でしか使えないとしたら、その便利さも半減してしまう。Windows が普及したもうひとつの理由は、数多くのメーカーのマシーンの上で Windows が動作できるという点であろう。

しかし、ここでまた、セキュリティ上の問題が発生する。数多くのアプリケーションが使えるとなればそれだけ便利になるのだから、メーカーはそのコンピュータに多数のアプリケーションを搭載することとなり、その分セキュリティ上の問題が発生する可能性は増す。さらに問題になるのは、インターネットが普及した現在、ある OS が多数のコンピュータで使えるようになればなるほど、その OS に何かセキュリティ上の問題があれば、その被害は一瞬にして世界中の膨大な数のコンピュータに広がるということである。

Windows の世界的な普及と、コンピュータセキュリティの問題が密接に絡んでいるのは、このようなことが背景にあると考えられる。

付録B 一般的な OS のアクセス制御

（1）ユーザ認証

通常のコンピュータでは、ユーザがそのコンピュータを使おうとするときは、最初にログオンあるいはログインと呼ばれる手順を踏んで、自分がそのコンピュータの正当な使用者であることを証明することが求められる。複数のユーザを想定しているコンピュータにおける、セキュリティを確保するための手段である。したがって、ユーザがひとりであることを前提とした初期のパーソナルコンピュータ、例えばマイクロソフト社の MS-DOS という OS を搭載したコンピュータには、このログオンという手順は用意されていなかった。

ログオンは、ユーザに固有の名前であるユーザ名とパスワードをコンピュータに入力するのが普通である。ユーザが入力したこのふたつのデータを、コンピュータ具体的には OS が事前に登録されていたデータと照合して、両者が合致すれば、そのユーザを正当な使用者として判断して、以後そのユーザの命令を実行することとなる。このようにしてユーザ認証は、セキュリティ対策の第一関門ともいうべき役割を果たす。

この認証の段階で、もうひとつ重要なユーザ管理の仕組みがある。一般的な OS では、ユーザは大別して、管理者（Administrator）というそのコンピュータに関して最大の権限を有するユーザと権限が制限された一般のユーザに分かれている。通常このユーザの区分をアカウントと呼んでおり、OS によってさらにこのアカウントが細分化されている。

例えば Windows XP では、管理者アカウント、制限付きユーザアカウント、パワーユーザアカウント、ゲストアカウントなどに区分されており、それぞれのアカウントに応じて、そのコンピュータで行使できる権限が決められている。UNIX や Linux にも、大別して管理者のアカウントと一般ユーザのアカウントがある。

このように、ユーザの役割に応じてその権限を区別することによって、不必要な権限を個々のユーザに与えないようにしているのが、アカウントという仕組みである。しかし、いずれにしても、管理者というほとんど絶対的な権限が存在することから、セキュリティ上の問題は依然として存在する。この管理者の権限を奪ってしまうような不正な攻撃方法があり、この攻撃を受けたコンピュータは悪意のある第三者の思うがままにコントロールされてしまうのである。

したがって、従来開発されてきた「セキュア OS」では、この管理者権限をどう扱うかが重要な課題のひとつだった。なお、UNIX や Linux のコンピュータセキュリティ関係の本では、この管理者を root（ルート）と呼び、「root 権限を取られる」といった言い方をする。Windows の場合は、この root に対応するのが Administrator あるいは admin（アドミン）と呼ばれているものである。

（2）アクセス制御

UNIX や Linux、Windows といった一般的な OS においても、ユーザ認証を行った後、ユーザはその与えられた権限の範囲内ならば何でもできるような仕組みになっているわけではない。

前述した Windows XP の制限付きユーザアカウントは、コンピュータにあるアプリケーションを使用できる権限を持っている。しかし、コンピュータの管理者が、特定のユーザ A には、例えばコンピュータの維持管理に関連するアプリケーション B は使わせたくないと考えたとしよう。その場合、この特定のユーザ A には、アプリケーション B を実行できないように、ユーザ A の権限を制限する機能が OS にある。

また、ユーザ C が作成した文書からなるファイル D があるとして、ユーザ C は他のユーザにはこのファイルを見せたくないが、ユーザ E にだけは見せてよいと考えているとしたら、ユーザ C はユーザ E にだけファイル D を読む権限を与えることができる。このような OS の機能をアクセス制御と呼ぶ。

以下、UNIX や Linux のアクセス制御をもう少し詳しく説明しよう。

先に述べたように、コンピュータ内のあらゆる情報資産は、ファイルという単位で保存されている。したがって、このファイルに対する各ユーザのアクセスを制御すれば、ユーザに不必要な権限を与えなくてすみ、そのコンピュータのセキュリティは向上することとなる。理想的には、そのコンピュータを利用するユーザ全員に対して個々にアクセス権を設定するのがよいであろう。しかし、UNIX のような OS は多数のユーザが共用することを前提としており、その個々のユーザに権限を設定することは、あまりにも複雑で管理作業が困難となることから、アクセス制御の細かさ（セキュリティ関係の本では、しばしば粒度：granularity という用語を使う）をかなり緩やかなものにしている。

ファイルには、オーナー（所有者）というものが設定される。通常、そのファイル（例えば文書）を作成したユーザがオーナーである。このオーナーが、当該ファイルへのアクセス権を設定することができる（管理者はこの設定を変更する権限を持っている）。

設定できるアクセス権（パーミッション）は、以下の三種類である。

- 読出し（read）：そのファイルの内容を読むことができる。
- 書込み(write)：そのファイルへ書き込みができる。
- 実行(execute)：プログラムとしてそのファイルを実行できる。

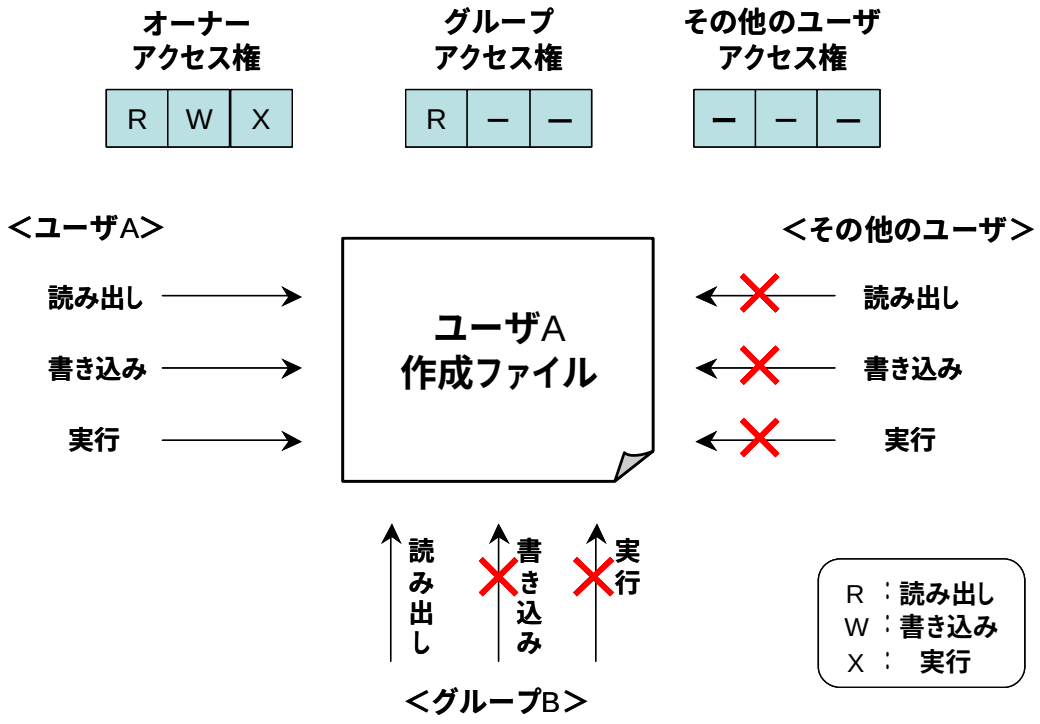
これらのアクセス権を、以下の三つの主体に対して別々に設定できる。

- オーナー：作成者自身のアクセス権である。
- グループ：UNIX では、ユーザは少なくともひとつのグループに属するように設定されている。複数のグループに属することも可能である。これによって、アクセス権の設定を簡便なものにできる。例えば、ユーザ A が作成した文書を、自分が所属するプロジェクトチーム B

のメンバーは見られるようにしたければ、このグループには読出しのアクセス権を設定するのである。

- その他　：その他のユーザ全員を意味する。上記の例を使えば、ユーザ A がファイルをプロジェクトチーム B のメンバーだけが見られるようにし、それ以外のユーザが見られないようにしたければ、この「その他」に対してはアクセス権を設定しなければよいのである（この場合も、管理者はアクセスできる）。

このようなアクセス権の設定は、Linux でも同様である（図Ｂ－１）。Windows XP も基本的には似たようなアクセス権の設定ができるようになっているが、より細かな設定を行うことができるようになっている。



図Ｂ－１　Linux によるアクセス権の設定

（３）通常のアクセス制御の問題点

上に述べたアクセス制御の方法は、ファイルの所有者（作成者）が、そのファイルに対する他のユーザのアクセス権を設定する権限（責任）があるとするものであり、一般的にこのようなアクセス制御を任意アクセス制御（DAC：Discretionary Access Control）と呼んでいる。この任意アクセス制御によって、基本的にはコンピュータ資産を保護できると考えられてきた。

しかし、このアクセス制御に限界があることも否定できない。次に、この DAC 機能では対応できない問題の一例を説明しよう。

あるコンピュータシステムにおいて、ユーザ A が作成したファイル B を、ユーザ C に対しては秘匿する方針があるとする。例えばファイル A の「グループ」にユーザ C が所属していないのなら、「その他」がアクセスできないようにパーミッションを設定すれば、ユーザ C はファイル A にアクセスできない。しかし、以下の操作によって、結果的にユーザ C がファイル B にアクセスできるようになる。

① ユーザ D はファイル B へアクセスできるとする。
② ユーザ D は、ユーザ C に協力的か、もしくはファイル B をユーザ C に秘匿するという方針を知らないとする。
③ ユーザ D はファイル B を読み出して、自分が所有するファイル E にコピーする。これは、ユーザ D がファイル B へアクセスできるので可能である。
④ ユーザ D は、ファイル E のパーミッションをユーザ B がアクセスできるように設定する。例えば、ファイル E の「グループ」のメンバーをユーザ D とユーザ C として、この「グループ」にアクセス権を設定する。
⑤ ユーザ C はファイル A へアクセスできるようになる。

このことは、DAC 機能だけでは、例示した「ファイル B をユーザ C に秘匿する」という方針が、このコンピュータシステム全体に貫徹できないことを示している。これは DAC 機能の限界の一例である。1970 年代の米国政府、特に米軍においてこの DAC 機能の限界が問題となり、いわゆる「セキュア OS」の典型であるトラステッド OS（Trusted OS）が誕生することとなる。

付録C セキュアOSの発展

（1）軍用システムのセキュリティ

① レファレンスモニター

1972年、「Computer Security Technology Planning Study」と題する報告書が米空軍へ提出された。報告者のリーダーの名前を使って、アンダーソンレポートとも呼ばれている。

前述したとおり当時の米空軍には、マルチプログラミング機能、タイムシェアリング機能を有するマルチユーザシステムが導入され始めていた。これをそのシステムが扱う情報の秘匿度という観点から見れば、コンピュータシステムのなかには秘密ではない通常の情報から極秘情報まで各種の情報が混在し、またそのユーザについても、極秘情報にアクセスを許された者から秘密情報にアクセスを許されない者まで各種のユーザが混在していた。このような環境においては、悪意の第三者が、例えばトロイの木馬をシステムに設定できたすれば、本来秘密情報にアクセスできないユーザがその秘密情報にアクセスできてしまうことになる。

したがって、このような脅威に直面した空軍が委託した報告書の中心テーマは、「システムが保有するデータのうち秘密情報の割合が、たとえ１％以下だったとしても、セキュリティの本質的性格から、軍はシステム全般にわたって有効なセキュリティ制御を提供」するということだった。

そして、このレポートで提唱されたコンセプトが、レファレンスモニター（Reference Monitor）だった。レファレンスモニターとは、簡単にいえば、コンピュータ内にある各データに対するユーザもしくはユーザが起動したプログラム（アプリケーション）のアクセスが許されるものか否かを監視するものである。

これを実際の仕組み（メカニズム）としてコンピュータシステム内に作り込むときは、そのメカニズムは以下の条件を満たさなければならないとされた。

- メカニズムは、悪意のあるプログラムなどによって修正されたり、無効化されてはならない。
- メカニズムは、すべてのアクセスリクエストをチェックするようになっており、このチェックを回避されてはならない。
- メカニズムは、それが正確に作動するか検証できる範囲内の大きさのプログラムでなければならない。

このメカニズムを適切に機能させるためには、「どの主体（Subject）が、どの対象（Object）をどのようにアクセスし操作できるのか」という具体的なルールをレファレンスモニターに定義しなければならない。個々のコンピュータシステムに応じた、このようなルールをセキュリティポリシーもしくはシステム・セキュリティポリシーと呼んでいる。

このセキュリティポリシーを形式的、数学的に記述しモデル化したものを、セキュリティモ

デルもしくはセキュリティポリシーモデルと呼んでおり、TCSEC はトラステッド OS がこのセキュリティモデルに基づいていることを求めている。

② ベル・ラパデュラモデル（Bell-LaPadula モデル：BLP モデル）

このモデルは、1973 年に David Bell と Lenn LaPadula が、米空軍の委託研究において提案したモデルである。

この BLP モデルは、以下の２種類の区分方法によって秘密情報を管理することを目的としている。

- 機密（Top Secret）、極秘（Secret）、秘（Confidential）、一般(Unclassified)という秘匿の強度に応じたいわば階層的な秘密区分（クラシフィケーション（Classification）と呼ぶ）
- 例えば傍受情報、大量破壊兵器、通常兵器、電子戦、人事、予算などの情報の種類による、いわば非階層的な区分（カテゴリーもしくはコンパートメントと呼ぶ）

そして、このモデルが実現しようとしている情報管理の基本的な考え方（ポリシー）は、以下のように要約できる。

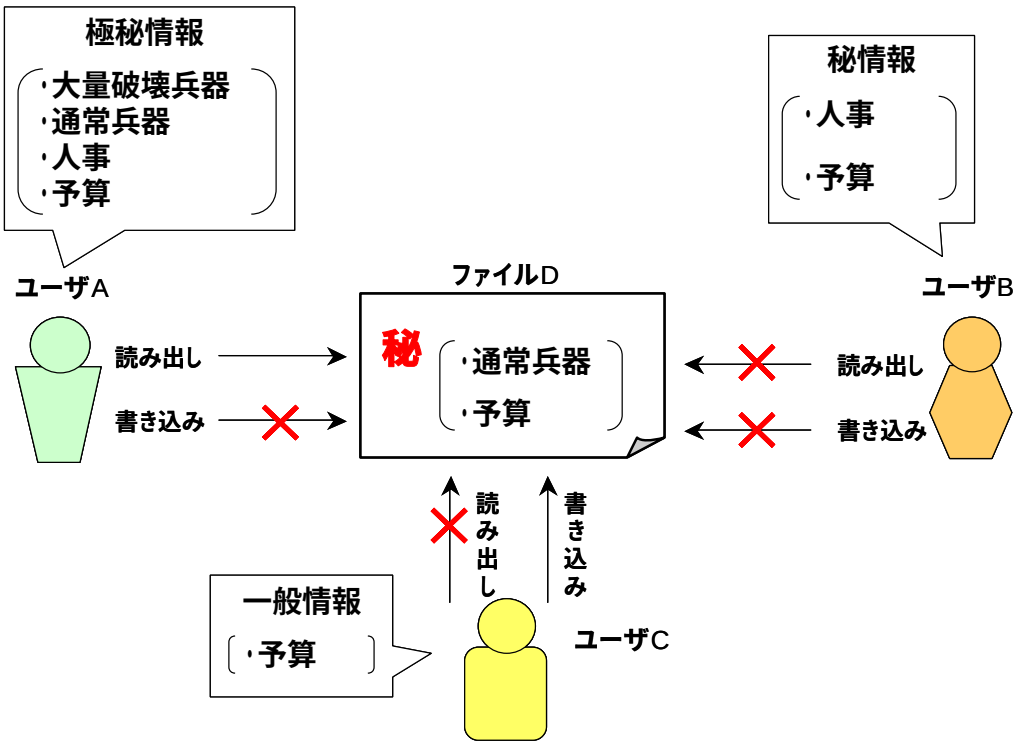
- どのユーザも自分がアクセスを認められている秘密情報の最高レベルと同等か、それより下位のレベルの情報を読むことができる。
- またどのユーザも自分が読むことができる秘密情報の内容を、それより下位のレベルの情報に書き込むことが許されない。ただし、上位のレベルには書き込むことができる。

このポリシーによって、コンピュータシステムの内部で、情報は上に流れることはあっても、下に流れることがなくなり、トロイの木馬のような脅威に対抗できることとなる。

以下の例で、具体的に説明することとしよう。

ユーザＡは、極秘情報にアクセスできる資格をもっているとする（米軍では、この資格のことをクリアランス（Clearance）と呼んでいる）。そして、ユーザ A は大量破壊兵器、通常兵器、人事、予算のカテゴリーに属する情報にアクセスできるとする。他方、ユーザＢは秘の情報にアクセスできる資格をもち、人事、予算のカテゴリーに属する情報にアクセスでき、ユーザ C は秘以上の情報にアクセスする資格がなく、予算のカテゴリーに属する情報だけにアクセスできるとする。

このようなユーザが、秘レベルの通常兵器と予算に関する情報が入っているファイル D にアクセスしようとした場合、BLP モデルでは以下のようなアクセス制御を行うことになる（図Ｃ－１）。



図C－1　BLP モデルに基づくアクセス制御

・読み出し

あるファイルをユーザが読むためには、ユーザのクリアランスのレベルがそのファイルの秘密区分のレベルと同等かそれより高いことが必要である。これに加えて、ユーザがアクセスできる情報のカテゴリーの範囲が、ファイルのカテゴリーの範囲と同じかそれより大きい（包含する）ことが必要である。

この例では、ユーザ A のクリアランスは極秘であり、ファイル D の秘密区分は秘であるから、ユーザ A のクリアランスのレベルはファイル D の秘密区分のレベルより高い。また、ユーザ A のカテゴリーは、{大量破壊兵器、通常兵器、人事、予算} であるから、ファイル D のカテゴリー {通常兵器、予算} を包含している。したがって、ユーザ A は、ファイル D を読むことが許される。

他方、ユーザ B のクリアランスは秘であるものの、カテゴリーは {人事、予算} であるから、ファイル D のカテゴリー {通常兵器、予算} より大きい（包含している）とはいえない。したがって、ユーザ B はファイル D を読むことは許されない。ユーザ C には、秘以上の情報にアクセスするクリアランスがないので、ファイル D を読むことはできない（No read up）。

・書き込み

あるファイルへユーザが書き込みをするためには、ユーザのクリアランスのレベルがそのファイルの秘密区分のレベルと同等かそれより低いことが必要である。さらに、ユーザのカ

テゴリーの範囲が、そのファイルのカテゴリーの範囲より小さい（包含されている）ことが必要である。

この例では、ユーザ A のクリアランスのレベルはファイル D の秘密区分のレベルより上であり、ユーザ A のカテゴリーはファイル D のカテゴリーを包含している。したがって、ユーザ A はファイル D への書き込みは許されない（No write down）。

他方、ユーザ B のクリアランスはファイル D の秘密区分と同じであるが、ユーザ B のカテゴリーの範囲はファイル D のカテゴリーの範囲に包含されているとは言えない。したがって、ユーザ B はファイル D への書き込みが許されない。ユーザ C には秘以上の情報へアクセスできるクリアランスがなく、そのカテゴリーの範囲も｛予算｝であり、ファイル D のカテゴリーの範囲に包含されているので、ユーザ C はファイルへの書き込みが許される。

この BLP モデルの読出しに関するアクセス制御は、比較的理解しやすいであろう。ユーザのクリアランスより高いレベルのファイルを読むことができないのは当然であり、また、ファイル D のカテゴリーには「通常兵器」というユーザ B のカテゴリーにはない情報があるのだから、このファイルの読出しを認めないのも理解できるであろう。このように秘密区分では同じレベルの情報でも、その者がアクセスする必要がない情報へのアクセスを制限するポリシーのことを、need-to-know の原則と呼ぶ。

他方、書き込みに関するアクセス制御は、少々理解しづらいかもしれない。たとえば、ユーザ A はファイル D を読むことができるが、書き込むことはできない。読めるのだから、書き込めてもよいように思われるが、BLP モデルでは、前述したようにコンピュータシステムの内部で情報が下位レベルへ流れることが一切ないようにすることをポリシーとしている。

したがって、ユーザ A がファイル D へ書き込めるとすると、例えばユーザ A は極秘レベルのファイルにアクセスできるのだから、そのファイルの内容を秘レベルのファイル D へコピーできることになり、結果としてファイル D の中に極秘レベルの情報が混在することとなり、秘レベルしかアクセスできないユーザもこの極秘情報を読むことができてしまう。BLP モデルは、このようなルートでの秘密情報の漏洩を防止することを目的にしているのである。

③ TCSEC

1981 年に米国防省内に Computer Security Center が設置された（1985 年に National Computer Security Center と改称）。このセンターの主要な目的は、秘密情報を扱うための信頼できるコンピュータシステムを広く米軍内に普及させることであった。このためにセンターが採った手法は、ベンダー（メーカー）が信頼できるシステムを製造するために使用される一般的、総合的なセキュリティ要件を作ることだった。そして、このセキュリティ要件は同時に、

製品の信頼性を評価する標準的な基準ともなりえたのである。この背後にある考え方は、「製品が作られたら、それは評価することができ、評価されたら、それは調達される」というものだった。

1983 年に、TCSEC が同センターから公表されたが、その表紙がオレンジ色だったことから、一般的には「オレンジブック」と呼ばれるようになった。1985 年には、内容を多少修正した後、国防省の指令として正式に制定された。これが、最初のセキュリティ評価基準である。なお、オレンジブック刊行後も、これを補うための文書が多数刊行されたが、これらの表紙の色に赤、緑、紫などの色が使われたことから、これらの文書はレインボーシリーズと呼ばれた。

TCSEC では、コンピュータシステムに対してセキュリティポリシーを実現させるためのハードウェアやソフトウェアのセットを TCB（Trusted Computing Base）と呼んでおり、アクセス制御だけでなく、例えば暗号だとか監査ログに関するメカニズムも含まれている。この TCB の中でアクセス制御を受け持つのが前述したレファレンスモニター（具体的なメカニズムを意味する場合は Reference Validation Mechanism と呼んでいる）であり、これが TCSEC の基礎にある最も重要なコンセプトのひとつであり、通常このメカニズムは OS に備わっていることが多い。

もうひとつの重要なものが BLP モデルであり、TCSEC で要求される強制アクセス制御（Mandatory Access Control：MAC）は、このセキュリティモデルを前提としたアクセス制御である。

TCSEC では、コンピュータシステムの評価基準を、A～D の Division と呼ばれるレベルに区分し、さらにその Division 内を１～３個のクラス（Class）に細分化している。簡単にそれぞれの Division を説明すれば、D はどのクラスにも評価できない製品が該当し、C は DAC 機能を実現している製品、B は MAC 機能を実現している製品、A は形式的（数学的）な手法でそのセキュリティが証明された製品が該当する。

C1 から A までのそれぞれのクラスには、各種の機能要件と保証要件が要求される。

機能要件とは、例えば DAC や MAC、ユーザ認証、監査ログなどのセキュリティを実現する機能の項目であり、保証要件とは、侵入テストやテストドキュメント、設計ドキュメントなどのセキュリティ機能がどれほど確実なものか保証する項目である。そして上位のクラスになるにしたがって、新しい要件が加わったり、要件がより加重されている（表Ｃ－１）。

表Ｃ－１　TCSEC における評価基準

Division	Class	評価基準	製品例
D		評価した結果、以下のどのクラスの要件も満たさなかったシステム	
C	**C1**	任意アクセス制御が要求される	
	C2	C1 に監査機能（ユーザのログイン等）等が追加される	WindowsNT
B	**B1**	C2 に強制アクセス制御等が追加される	HP-UX BLS Trusted Solaris
	B2	B1 の強制アクセス制御がシステム内の全ての主体（サブジェクト）と対象（オブジェクト）の間の制御に求められる	Multics
	B3	B2 のアクセス制御、暗号、監査等のセキュリティ機能に関するハード、ソフトが分析や検査を受けることができるように、単純化されていることが求められる	XTS-300
A	**A1**	機能的には B3 と同じだが、その機能が確実に実現していることを形式的（数学的）に証明が求められる	SCOMP

• C1 クラス

C1 が求める機能要件は、最小限のもである。パスワード等のユーザ認証と個人単位またはグループ単位での単純な DAC 機能である。保証要件としても、最低レベルのセキュリティ・テストやテストドキュメントなどに限られる。前述した通常の Unix や Linux のアクセス制御は、この条件を十分満たしていることから、このクラスの認定を求める商用の OS はほとんど無かった。

• C2 クラス

C2 は DAC 機能にしても、C1 がグループ単位でもよかった点が強化され、必ず個人単位でアクセス制御が実現することを求めている。監査についても、ユーザ個人の各アクセスについて追跡できることを求めている。その他、テストについても C1 より広い範囲をカバーすることが求められる。商用の OS に求めるセキュリティ要件としては、基本的に十分と考えられることから、マイクロソフト社の Windows NT3.5 や Windows NT4.0、Novell 社の NetWare4 など多数の商用 OS が認定を受けている。

• B1 クラス

B1 から MAC 機能使われるようになり、ラベルの使用が求められる。
ラベルとは、コンピュータシステム内のアクセスの主体（Subject：典型的にはユーザ）とアクセスの対象（Object：典型的にはファイル）のそれぞれに、BLP モデルで説明した秘

密区分とカテゴリーの情報を添付することである。ペーパーの文書ファイルに、他のファイルとの区別を容易にするために、ラベルを貼るのと同じイメージである。

前述の例では、ファイル D には「秘」という秘密区分の情報と｛通常兵器、予算｝というカテゴリー情報がラベルとして添付されると考えるとよい。そして、同じことがユーザ A、B、C にも行われる。この主体と対象双方のラベルを照合し、BLP モデルのルールに従ってアクセス制御を行うのが、TCSEC における MAC である。B1 ではこの他に、MAC 機能以外にもアクセス制御があるならば、それもセキュリティモデルに基づいていることが求められたり、より広範囲なテストが求められている。

B1 の認定を受けた製品には、ヒューレット・パッカード社の HP-UX BLS 8.04 や IBM 社の MVS/ESA、AT&T 社の SystemV/MLS 1.1.2、サン・マイクロシステムズ社の Trusted Solaris 8 といった製品がある。なお、TCSEC ではトラステッド（Trusted）OS という用語は使われていないが、TCSEC 制定後、この B クラス以上の認定を受けた OS もしくはそれと同等の機能を備えた OS のことを、一般にトラステッド OS と呼ぶようになったのである。

• B2 クラス

B2 においては、B1 ではラベルは必ずしもすべての主体や対象に求められていなかったのに対し、すべての主体や対象（例えばプリンターのような周辺機器）にラベルが必要となる。さらにこのクラスで求められるセキュリティモデルは、形式的（数学的）なものが求められる。ソースコードの分析も求められる。ハネウェル社の Multics MR11.0 が認定されている。

• B3 クラス

B3 になると、前述したレファレンスモニターの要件を満たすことが求められる。TCSEC では、認定を受けるコンピュータシステムの TCB を定義する必要があるが、この TCB はセキュリティに関係しないものが混在しないように、単純な構成が求められる。監査メカニズムはセキュリティ関連のインシデントをモニターし、危険な情況になれば警告する機能が求められる。このように B3 クラスは、システムへの侵入に対して極めて強固であることが求められる。B3 の認定を受けた製品としては、Wang Government Services 社の XTS-300 シリーズがある。

• A1 クラス

A１クラスは、機能的には B3 と同じであるが、その機能が確実に実装（製品化）されて

いることを、形式的（数学的）に最高度の証明が求められる。OS としては、ハネウェル社の SCOMP のみが認定された。

なお、認定製品のバージョンを明示したが、認定は特定のバージョンを対象としていることから、バージョンアップについては再度認定を受ける必要がある。

（2）商用システムのセキュリティ

TCSEC の基礎にある BLP モデルは、米軍における軍事情報の漏洩防止というニーズに応えようとするものだった。一般的に、情報セキュリティとは「情報資産の機密性、完全性、可用性を保証すること」と説明される。

機密性（Confidentiality）とは、権限のない人に情報資産が漏洩しないこと、完全性（Integrity）とは、情報資産が改ざんや破壊されることなく正しい状態に保たれること、可用性（Availability）とは、使用権限のある者が常に必要な情報資産にアクセスし使用できることである。軍事においては、この機密性、秘匿性が重要視されてきた。軍事機密が漏洩してしまうのならば、むしろその情報は廃棄したほうがよいとする考え方が成り立ちえたのである。

しかし、民間の企業では、むしろ完全性が重視されてきたといえる。近年、顧客情報の漏洩は大きな社会問題となっているが、一般的には顧客情報の漏洩よりは、むしろその改ざんや破壊のほうがはるかに企業にとってダメージが大きいと考えられてきた。例えば、銀行の預金者の預金情報が流出することも銀行にとって大問題であるが、情報システムの欠陥で預金者の残高が勝手に改ざんされたり破壊されたら、その銀行が受けるダメージは計り知れないものとなるはずだった。

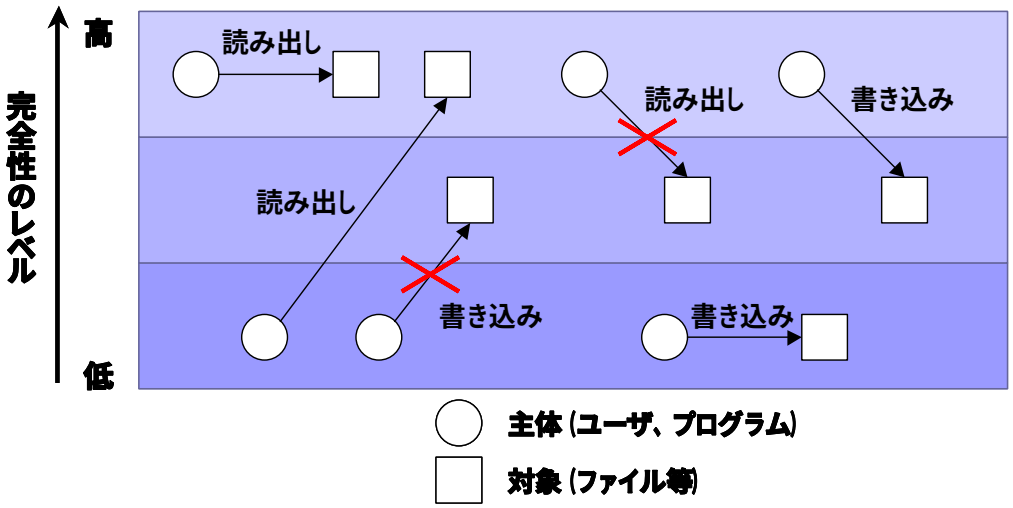
このような軍と民間における情報セキュリティの在り方の違いから、BLP モデルの提唱以降、情報の完全性を重視するセキュリティモデルが提唱されてきた。以下、代表的なモデルを説明することとしよう。

- **Biba モデル**

  Biba モデルは BLP モデルと同様にいわば多階層のセキュリティモデルではあるが、情報の完全性の維持を目的にして、主として情報の書き込み、更新を管理・制御するものである。このモデルでは、下位の階層の情報は上位の情報に比して不完全である、という考え方を前提としており、以下のルールに基づいてアクセス制御を行う（図Ｃ－２）。

  - ある主体（Subject）は、下位の対象（Object）に限り修正・変更できる。
  - もしある主体が、ある完全性のレベルの対象を読み込む権限を持っているとすれば、その主体はその対象の完全性レベル以下の階層の対象にのみ書き込むことができる。

ある主体が下位の対象を読み込み、上位の対象に書き込む権限があるとすれば、その上位の対象の完全性は下位の対象にある完全性の低いデータによって不完全なものになる危険性があるので、そのような上位への書き込みは許可しない（No write up）。したがって、ある主体が下位の対象を読み込むことも無制限には許可しない（No read down）。



図Ｃ－２　Biba モデル

• Clark-Wilson モデル

Clark と Wilson は、コンピュータシステムにおいて完全性を維持するためには、以下の条件が実現される必要があるとした。

- 不当なユーザによる、データおよびプログラムの変更の禁止
- 正当なユーザによる、データおよびプログラムの不正な変更の禁止
- コンピュータシステム内のデータ間の一貫性およびコンピュータシステム内のデータと実世界におけるデータの一致性

コンピュータシステムではデータはデジタルデータとして機器の内部に保存されており、第三者のチェックが入りにくい。したがって、例えば民間における複式簿記のように、取引を２つの帳簿に別々の担当者が記録し、それらを照合することによってこの担当者が共謀しない限り不正な操作が起きないようにするといった、実社会におけるチェック機構に相当する機能をコンピュータシステム内に設ける必要がある。

Clark-Wilson モデルの技術的詳細は省略するが、以上の考え方に基づいて Clark と Wilson は完全性の維持のためには、次の２つのコンセプトが重要だとした。

• 定式化されたトランザクション（well-formed transaction）

データの完全性を保証するには、制限された方法でしかデータを操作できないよう

にするための手続きが必要である。ファイルの読み込みや書き込みといった基本的な操作を自由に組合すこと（例えばDAC）を許せば、誤ってデータを破壊したり、不正にデータを改ざんできることとなる。

このため主体（ユーザ）が、OSが提供するDAC機能などによって対象（データ）に直接アクセスするのではなく、主体と対象を一定の業務処理を内容とするプログラムが仲立ちして、その業務手順にしたがってのみ主体が対象にアクセスできるようにすることによって、誤操作や不正な処理を防止することができる。このような複数の基本的な操作を、一定の手順に則った不可分の処理の集まりとしたものが、well-formed transactionである。

- **職務の分離**（separation of duty）

一連の業務処理を、一人の担当者のみに実施させないという考え方である。複数の担当者による相互チェックは、不正や過誤の防止に有効である。また、担当者の権限を分割することによって、ある権限が不正に行使されたとしても、その被害や影響を最小限にすることができる。

- **RBAC**（Role-Based Access Control）

Biba モデルや Clark-Wilson モデルの他にも、Chinese Wall モデルや BMA（British Medical Association）モデルなど各種のセキュリティモデルが提唱されている。例えば弁護士事務所や会計事務所などにおいて、利害が対立する企業が共にその事務所の顧客であることも想定し得る。事務所の関係職員が競合する企業双方の情報にアクセスできるとすると問題が生ずることが考えられることから、Chinese Wallモデルはこのような場合のアクセス制御を取り扱うものである。また、BMAモデルは、英国医学協会が策定したモデルである。病院内の医師、患者、看護婦などの間で患者情報などに対するアクセス制御をどのように行うかを、英国内の病院における長年の慣行を参考にして作成されたモデルである。

このような各種のモデルの中で近年注目され、各種の OS やアプリケーションにおいても導入されているのが、RBACである。

大組織のコンピュータシステムになればユーザは数千人以上となることもあり、各ユーザに対するアクセス権限の設定を厳格に行おうとすれば、複雑で膨大な作業となる。一定の時期に大規模な人事異動がある組織では、ユーザごとの権限の設定変更などの作業が集中的に発生することから、作業は極めて困難なものとなることが予想される。

また、前述したようにOSが制御するファイルについては、オーナー（作成者）がいる。オーナーが、そのファイルに対する他のユーザのアクセス権限を設定するのが、UNIX や

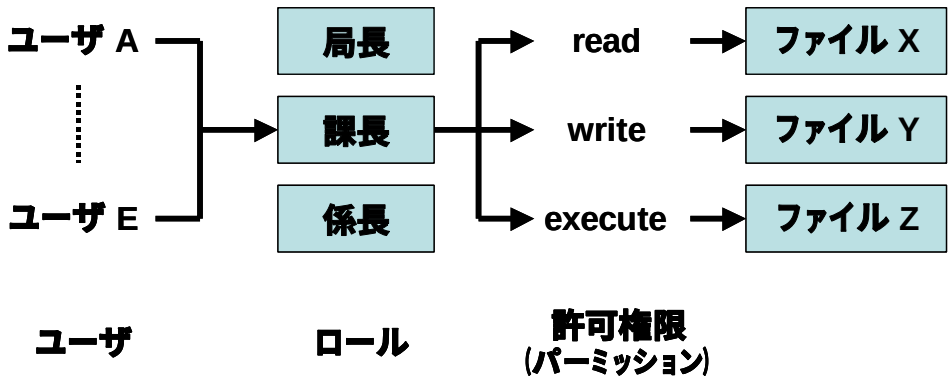
Linux, Windows といった一般的な OS のアクセス制御である。しかし、企業などの組織においては、コンピュータシステム内のファイル（データ）は、そのファイルの作成者が所有するものではなく、その組織の所有物と考えるのが一般的であろう。

さらに、通常その組織の職員は、与えられた職務や役割に応じてその組織が所有するデータに対するアクセスが認められている。例えば、人事関連の情報に一般の職員はもとより、幹部であっても人事に関係ない者はアクセスできないが、人事担当の職員であれば、その職位が低くてもその職務、役割を遂行するためにアクセスが認められているのが普通である。しかも、この人事担当者も人事と関係のない部署に異動すれば、自分が作成した人事関係の書類にはアクセスできないのが通常のセキュリティポリシーであろう。

このような観点から、個々のユーザという単位ではなく、ロール（role：職務、役割）という概念に基づくアクセス制御が提唱されるようになった。これが、Role-Based Access Control と呼ばれているものである。以下、この RBAC における重要なコンセプトについて説明する。

- **許可権限**（permission）

  RBAC では、まず複数のユーザを特定のロールに割り当てることとなる。ある組織ではユーザ A~E が課長の職位にあるとしたら、このユーザ A~E を課長というロールに割り当てることとなる。そして、課長というロールに対して組織として許可した権限がある。この許可権限は、このロールができる操作（OS であれば、read, write など）とその対象（ファイルなど）から構成されている（図Ｃ－３）。



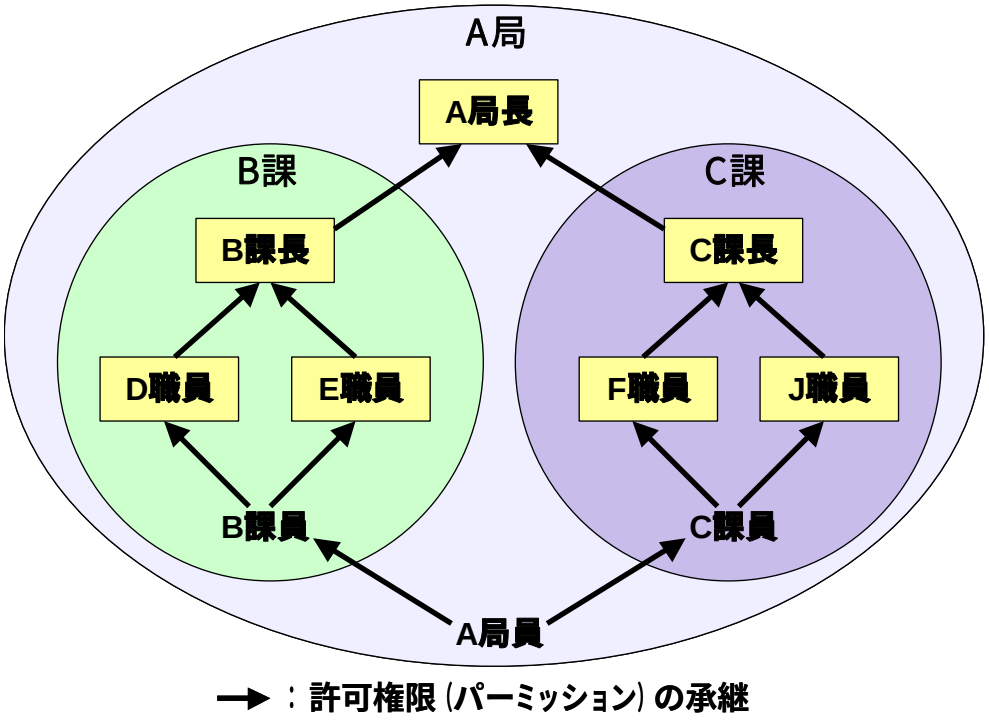
図Ｃ－３　許可権限

ロールは、次にこの許可権限へ割り当てられる。多くの組織において、ユーザのロールへの割り当ては、頻繁に変更される可能性がある。しかし、これは人事異動に伴うものであり、コンピュータシステムにおいても同時にこの変更を実施することは、それほど困難な作業ではないはずである。

他方、課長というロールやそのロールに認められた許可権限は比較的安定しており、頻繁に変更されることは通常想定されない。したがって、このロールや許可権限に関する作業は比較的容易であることから、RBAC によってアクセス権限の設定・変更作業が効率化できると考えられるのである。

- **ロール階層**（role hierarchy）

一般に大きな組織では、そのポスト（ロール）は階層構造になっており、下位のポストの権限は上位のポストの権限として含まれる。課長の権限はその上司である局長の権限に含まれている。このことを RBAC では、局長の許可権限は部下の課長の許可権限を継承している(inheritance)と言うことがある。このような承継の概念を用いて、ロールの階層構造を組み立てることができる（図Ｃ－４）。



図Ｃ－４　ロール階層

A 局は B 課と C 課から構成されているとする。各課には D から J の職員が配属されそれぞの権限が認められている。また、B 課には B 課の職員に共通する権限があるとする。C 課も同様である。さらに A 局の局員全員に共通する権限もあるとする。図から分かるように、例えば A 局長はこれらすべてのロールの許可権限を承継している。また、A 局の局員全員に共通した権限がない場合は、図の A 局員というロールはいらないことになる。

このようなロール階層によって、A 局のアクセス権限の設定変更は、次のように効率化できるようになる。例えば、B 課に新しい任務が与えられていたとする。この任務は E 職員に付与され、それに対応する権限が E 職員固有のものとなったとする。この場合、承継というメカニズムによって、A 局長や B 課長の許可権限は、自動的に E 課員の許可権限を含むことになっているので、A 局長や B 課長の許可権限を個別に設定変更する必要はなくなるのである。

このような特長を有する RBAC には、前述した Clark-Wilson モデルの職務の分離を取り込んでいると考えられ、また UNIX 等の一般的な OS にある管理者権限という強大な権限を適正な適正な範囲でロールに分割することが可能となる。さらに近年は、MAC 機能や DAC 機能を RBAC で実現する研究も行われている。

RBAC は特定のポリシーに基づくセキュリティモデルというよりは、各種のポリシーを実現するアクセス制御の手法と位置づけることができ、OS だけでなく、データベースなどのアプリケーションへも適用されている。このように、様々な民生分野での活用が期待されることから、米国においては商務省の機関である NIST(National Institute of Standards and Technology)が中心となって多数の研究者によて研究がなされている。

（3）国際セキュリティ評価基準

1985 年に TCSEC が米国防省の指令として正式に制定されてから、ヨーロッパを中心に各国で、それぞれの評価基準や評価制度が作られた。

例えばイギリスでは、1987 には商用 IT 製品のセキュリティ評価が開始され、Green Book と呼ばれる評価基準を公表した。これに引き続きフランス、ドイツにおいてもそれぞれ評価基準が作成された。1991 年には、これら 3 か国にオランダが加わり統一された評価基準 ITSEC が刊行され、この基準に基づき各国の評価制度にしたがって製品評価が行われた。さらにカナダでは、1993 年に TCSEC と ITSEC の統合を図った CTCPEC(Canadian Trusted Computer Product Evaluation Criteria)が刊行された。

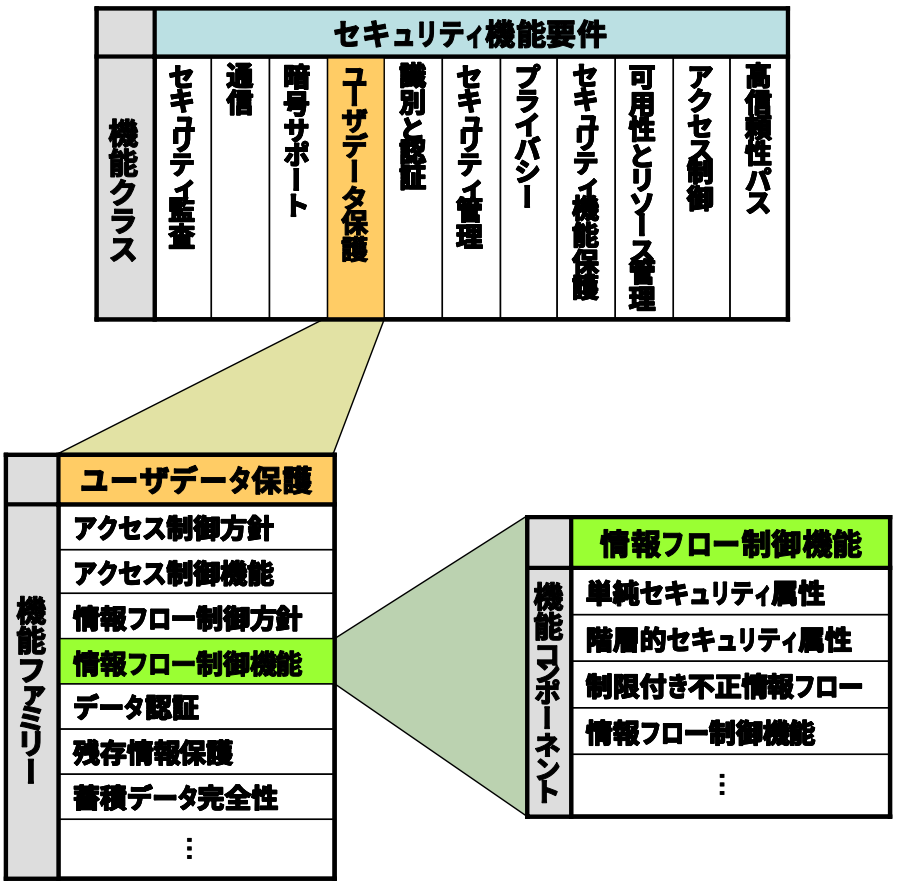
他方米国では、前述した商務省の機関である NIST と国防省の情報セキュリティ機関である NSA（National Security Agency）が共同して新しい評価基準 FC（Federal Criteria）の草案を 1992 年に公表した。これも、TCSEC と ITSEC の統合を図る試みだったが、その後国際標準としての CC（Common Criteria）の作成が、米国やカナダ、ヨーロッパの関係国の間で合意されたことから、FC はこの CC 策定作業に発展的に解消されていった。CC は 1999 年に ISO/IEC15408 として国際標準に採用された。以下、CC の主要な特長について説明することとしたい。

• **機能要件と保証要件の分離**

TCSEC では、前述したようにセキュリティ強度の分類として D から A１までのクラスを設定し、それぞれのクラスに応じて、求められるセキュリティ機能を定めた機能要件とその機能がどの程度確実に実装されるかを確認するための保証要件がセットになっていた。

したがって、例えばベンダー（メーカー）が MAC 機能を備えた OS（B1 クラス）を開発しようと考えた場合、TCSEC の B1 クラスに規定された機能要件および保証要件のすべてを満たす必要がある。その要件のひとつでも満たすことができなければ、B1 の認定を受けることはできず、C2 の認定を受けられるだけだった。このように TCSEC では、機能要件と保証要件がセットで厳格に規定されており、これらの機能を自由に組合すことができなかった。TCSEC と比べた場合の CC の最大の特長は、CC が機能要件と保証要件を分離したことである。このことによって、ベンダー側にも政府等の発注側にも自由度ができ、多様な製品の開発が可能となった。

機能要件に関しては、CC は様々な機能要件のいわばカタログであり、基本的にはこのカタログに記載されている各要件を自由に選択して製品を開発することとなる。機能要件は上位からクラス、ファミリー、コンポーネントとして分類されている（図Ｃ－５）。



図Ｃ－５　セキュリティ機能要件の階層構造

例えば、ユーザデータ保護というクラスは、アクセス制御方針、アクセス制御機能、情報フロー制御方針、情報フロー制御機能、データ認証、残存情報保護、蓄積データ完全性などのファミリから構成される。さらに情報フロー制御機能は、単純セキュリティ属性、階層的セキュリティ属性、制限付き不正情報フローなどのコンポーネントから構成されている。

保証要件は機能要件とは多少異なり、選択の方法に制限がある。保証要件もクラス、ファミリ、コンポーネントとして分類されている。しかし、保証要件については、EAL（Evaluation Assurance Level）といういわば保証の尺度が CC に用意されている。EAL1 から EAL7 までの各レベルに応じて、保証要件がセットされている（表Ｃ－２）。ただし、この EAL に CC に規定されている保証要件項目の中から部分的に選択して付加することもできる。一般的に EAL１から EAL5 までが商用の製品に適用される基準であり、EAL６、EAL7 は軍用の製品に適用するものとされている。

表Ｃ－２　CC（Common Criteria）の保証要件

保証レベル (EAL)	**主要な評価内容**
EAL 1	開発者の支援なしに使用者が行えるレベルの評価 ・仕様書通りであることを確認するための機能レベルのテスト ・ガイダンス文書の検査　等
EAL 2	開発者の一定の協力は必要となるが、通常のビジネスで対応する以上の負担はかけない ・開発者が行った機能仕様書に基づくテスト結果の検査 ・開発者が行った脆弱性評価の検査　等
EAL 3	開発者により実施されたテストの厳密な検査 ・開発者が行った概要設計書に基づくテスト結果の検査 ・開発環境管理の検査　等
EAL 4	セキュリティ専門家の特別な知識や技術を用いずに得られる最高レベルの保証 ・詳細設計書と一部の実装の検査 ・評価者による、低レベルの攻撃によるテスト　等
EAL 5	セキュリティ工学技術の専門的な支援を受けた厳格な開発方法で得られる商業ベースの最高の保障レベル ・すべての実装の検査 ・開発者が行った詳細仕様書に基づくテスト結果の検査 ・評価者による中レベルの攻撃によるテスト　等

TCSEC が主として OS を対象とした基準とされ、実際に認定された製品も OS が中心であるのに対して、CC は OS 以外にもファイアウォールや IDS、データベースソフト、ルータ、バイオ認証システムなど各種製品に広く適用されている。

EAL2 と認定された OS には、レッドハット社の Red Hat Linux3 がある。EAL3 にはレッドハット社の Red Hat Linux AS V3 などがあり、EAL4 にはサン・マイクロシステムズ社の Trusted Solaris8、マイクロソフト社の Windows 2000 Professional Server などがある。

• PP の導入

CC では、製品のカテゴリーごとに、調達側が必要とする機能要件と保証要件を選択して作成するセキュリティに関する仕様書を PP（Protection Profile）と呼んでいる。この PP をあらかじめ各種用意しておくことによって、実際に製品を調達したり開発するときに、この PP を利用すれば、調達側とベンダーがゼロから検討をする必要がなくなり、調達や開発業務が効率的に行われるようになる。実際、CC 制定後各国で様々な PP が作成されている。

特に米国政府は、政府が調達する IT 製品に関する PP を網羅的に整備しており、OS の分野でも各種の PP の正式版やドラフトが作成されている。これらの大部分は、米国防省の機関である NSA が発注し作成されたものである。このなかには例えば、ラベルによるアクセス制御の PP や秘密情報や諜報活動に関わる情報を扱うシステムに採用すべき、マルチレベルの OS に関する PP がある。

あえて TCSEC と比較すれば、ラベルによるアクセス制御の PP は、TCSEC の B1 におけるラベルによる MAC 機能を引き継ぐものであり、マルチ OS の PP は B2 以上に相当すると考えられる。TCSEC は、2001 年に国防省指令としては効力を失い、製品評価も終了した。しかし、TCSEC における B1 以上のセキュリティ要件は、TCSEC にはなかった機能も加えることにより、PP としてより発展進化した形で米国政府の調達基準となっていると言えるだろう。

付録D セキュア OS における権限情報の例

D.1 SELinux

（1） ロール

標準で提供されているロールは以下の 4 種類である。

表 D－1　SELinux で提供されるロール

ロール名	用途
sysadm_r	システム管理者用
staff_r	sysadm_r への遷移許可を有するユーザが管理業務以外で使用
system_r	システムが内部で使用
user_r	管理者以外の一般ユーザが使用

（2） アクセス権限の種類

SELinux でファイルなどのオブジェクトへのアクセス権限の制御に用いられる名前（access vectors, アクセスベクタ）について、ファイルを対象としたものを中心に例を示す。アクセスベクタにはこのほか、通信を対象にしたものなどを含め、合計 100 種類以上（Fedora Core 3 に組み込まれている SELinux の場合）用意されている。

表 D－2　SELinux で提供されるアクセス権限の例

アクセス権限の種類（アクセスベクタ、抜粋）	意味
add_name	ディレクトリに対してエントリ（ファイル等）を追加可能
append	ファイルへの追記が可能
create	新たにファイルやオブジェクトの生成が可能
execute	実行可能（ドメイン遷移を伴う実行）
execute_no_trans	実行可能（同じドメインでの実行）
getattr	ファイルなどの属性の取得が可能
lock	ファイルやメモリのロックが可能
read	ファイルなどからの読み出しが可能
remove_name	ディレクトリからエントリ（ファイル等）を削除可能
rename	ファイルの名前の変更が可能
rmdir	ディレクトリを消去可能
unlink	ファイルを消去可能
write	ファイルなどに書き込みをすることが可能

（文献[2][3]と SELinux の設定ファイルをもとに作成）

D.2 Trusted Solaris

Trusted Solaris 8 の場合、69 種類の最少特権が標準で定義されている。この例を以下に示す。

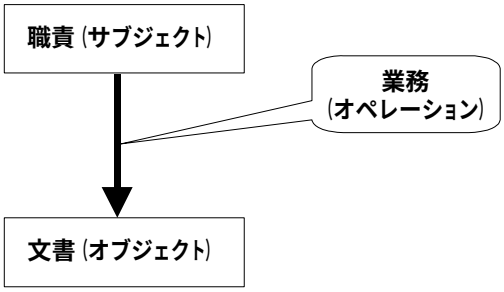
表 D－２　Trusted Solaris の標準で提供される最少特権の例

特権の名称（抜粋）	意味
file_audit	監査用情報の取得／設定を許可する
file_chown	ファイルの所有者の変更を許可する
file_dac_execute	本来実行する権限のないファイルの実行を許可する
file_dac_read	本来読み出す権限のないファイルの読み出しを許可する
file_dac_write	本来書き込む権限のないファイルの書き込みを許可する
file_downgrade_sl	ファイルやディレクトリの機密ラベルをより低位のものに変更することを許可する
file_mac_read	機密ラベルにおいてより低位のファイルやディレクトリの読み出しを許可する
file_mac_write	機密ラベルにおいてより低位のファイルやディレクトリへの書き込みを許可する
file_upgrade_sl	ファイルやディレクトリの機密ラベルをより高位のものに変更することを許可する
sys_audit	監査用プログラムを実行することを許可する
sys_boot	OS を実行しているハードウェアの停止、再起動を許可する
sys_console	コンソール画面への表示を外部デバイスに送ることを許可する
sys_net_config	ネットワーク設定の変更を許可する

（Trusted Solaris 8 マニュアルの「priv_desc(4)」についての記載事項をもとに作成）

付録E 文書管理業務モデルの詳細

行政文書に関する作用のモデルを図Ｅ－１に示す。図のごとく、文書への作用は、作用の主体である「職責(サブジェクト)」から客体である「文書(オブジェクト)」への「業務(オペレーション)」の関係と捉えることができる。



図Ｅ－１　文書の作用モデル

以下では、決裁、管理、利用のフェーズ毎に、文書、職責、業務をモデル化していく。なお、モデル化にあたっては、理解の容易化のため、行政文書に関連する業務の本質を損なわない程度に簡単化し、表現等も一般化している。したがって、実際の業務、名称等とは細部が異なる場合があることに留意いただきたい。

E.１ 決裁フェーズ

（１）文書（オブジェクト）

決裁フェーズにおける行政文書の第一の分類として、最終決裁者の違いによる以下の三つの区分があげられる。

大臣/長官決裁文書：　　最終決裁者が大臣/長官である文書
局長決裁文書：　　最終決裁者が局長である文書
課長決裁文書：　　最終決裁者が課長である文書

また、上記とは異なった切り口による第二の分類として、文書の秘密度によるものがあげられる。「秘密文書区分」としては、それぞれ以下のように取り扱われる四つの区分を定義する。なお、極秘文書と秘文書を「秘密文書」と総称する。

極秘文書：　　秘密保全の必要性が特に高く、その漏洩が国の安全、利益に損害を与える

おそれのある情報を記録した行政文書

秘文書：　極秘に次ぐ程度の秘密であって、関係者以外に知らせてはならない情報が記録された行政文書

取扱注意文書：　秘密文書の指定は要しないが、その取扱いに慎重を期する必要がある情報を記録した行政文書

一般文書：　上記以外の文書

上記二つの分類は、論理的には独立した概念である。したがって、決裁フェーズにおける行政文書は、これら二つの属性を同時に持つことになる。

（2）職責（サブジェクト）

決裁業務における職責は、大きく以下の四つに分類される。

大臣/長官：　大臣/長官決裁文書の最終決裁を行う

局長：　大臣/長官決裁文書の中間決裁、および、局長決裁文書の最終決裁を行う

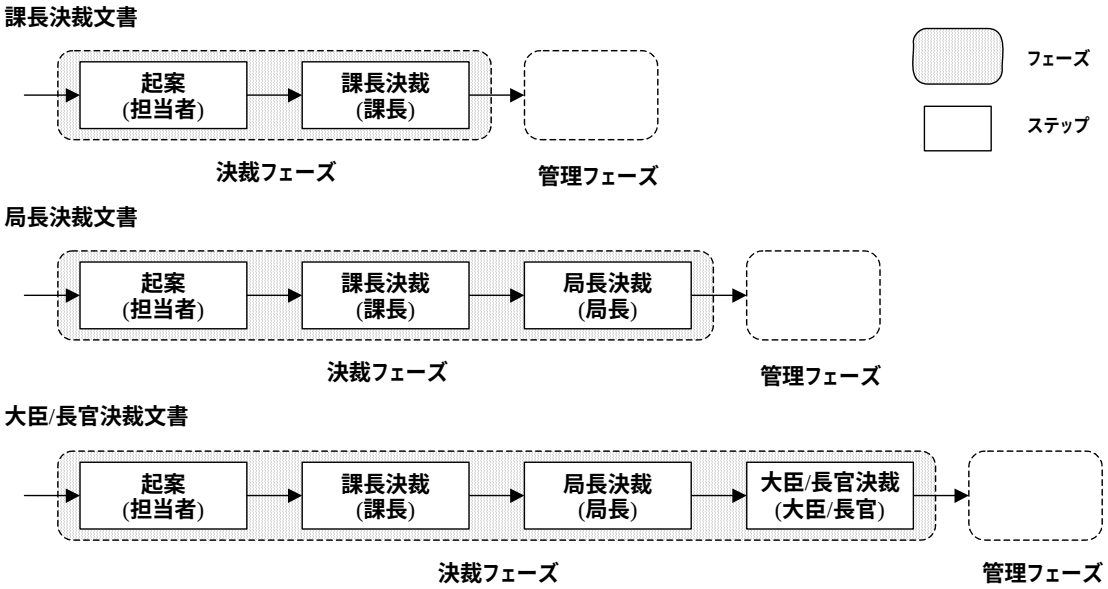
課長：　大臣/長官決裁文書、局長決裁文書の中間決裁、および、課長決裁文書の最終決裁を行う

担当者：　各決裁文書の起案を行う

（3）業務（オペレーション）

決裁フェーズでは、大臣/長官決裁文書、局長決裁文書、課長決裁文書の違いにより、図Ｅ－２のとおりのワークフローをとる。なお、図中、フェーズ(網掛け部分)内のボックス(白抜き部分)を、ステップと呼ぶことにする。

起案ステップにおいては、担当者が、秘密文書区分、秘密取扱期間(秘密文書の場合)、保存期間、開示範囲等の属性の指定を行う。決裁ルートに沿った各決裁ステップにおいては、各決裁者が、起案内容、秘密文書区分、秘密取扱期間(秘密文書の場合)、保存期間、開示範囲等を確認し、決裁する場合には決裁印の押印を行う。



図Ｅ－２　決裁フェーズのワークフロー

（４）特長

決裁フェーズにおける特長としては、以下の要件があることがあげられる。

- 起案文書への閲覧および決裁は、起案文書の現在の決裁状況に応じて、決裁ルートに沿った次の決裁者のみ可能であること
- 指定権限を持つ決裁者の決裁をもって、極秘文書/秘文書等の指定(属性の設定)ができること
- 決裁ステップにおいては、決裁者以外の者が決裁印を使用できないこと
- 同じく決裁ステップにおいて、決裁者が押印したことを後に証拠たり得る形で検証可能なこと

E.２ 管理フェーズ

（1）文書（オブジェクト）

秘密文書と秘密文書以外の二つに大別できる。

秘密文書については、極秘文書か秘文書かにより後述の総括文書管理者または主任文書管理者により指名された秘密文書取扱責任者が、行政文書ファイルの他課移管、分割/統合、廃棄等の管理業務を行う。

秘密文書以外の行政文書については、文書管理者により指名された文書取扱責任者が管理業務を行う。

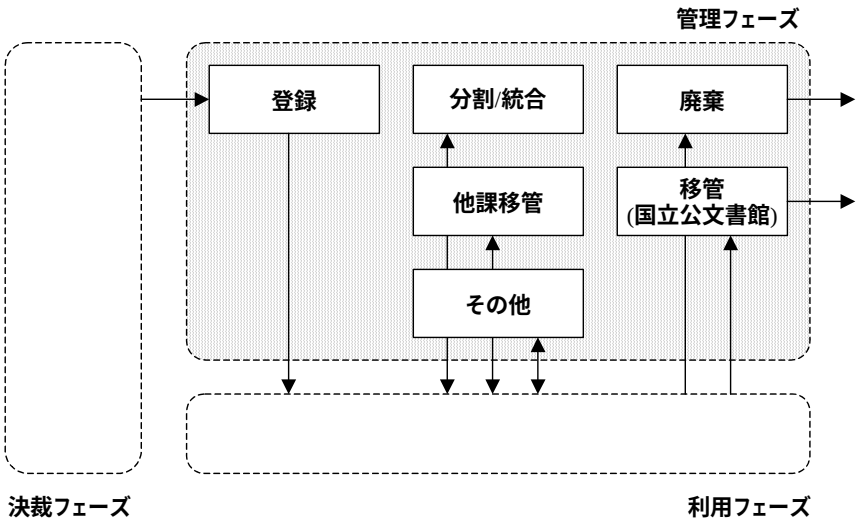
（2）職責（サブジェクト）

文書管理業務に関わる職責として以下を定義する。

総括文書管理者：	省庁における行政文書管理業務を総括する
主任文書管理者：	極秘文書に関する秘密文書取扱責任者の指名を行う
文書管理者：	秘文書に関する秘密文書取扱責任者の指名を行う
文書取扱責任者：	文書管理者からの指名により、秘密文書以外の文書の取扱いについての事務を行う
秘密文書取扱責任者：	主任文書管理者(極秘文書の場合)または文書管理者(秘文書の場合)からの指名により、秘密文書の取扱いについての事務を行う

（3）業務（オペレーション）

図Ｅ－３に管理フェーズのワークフローを示す。



図Ｅ－３　管理フェーズのワークフロー

決裁フェーズを終えた行政文書は、文書取扱責任者または秘密文書取扱責任者により、最終的な行政文書としての体裁を整えられたのち、文書を一意に識別するための番号である文書番号が付与され、管理フェーズの登録ステップで利用フェーズに移される。

利用フェーズで所定の保存期間を経過した行政文書は、管理フェーズに戻され、廃棄ステップにより廃棄が行われる。特に秘密文書の場合には、立会人の立ち会いのもとに廃棄を実行する。

分割/統合ステップでは、行政文書に付与された各種属性の変更に伴って、行政文書ファイルの再整理が行われる。他課移管ステップでは、行政文書の管理元属性の変更が行われる。また、その他ステップとしては、秘密取扱期間経過後の秘密文書区分の変更等が行われる。

（4）特長

管理フェーズにおける特長としては、以下の要件があることがあげられる。

- 秘密文書の廃棄以外のステップは、文書取扱責任者または秘密文書取扱責任者によってのみ実行できること
- 秘密文書の廃棄ステップは、文書取扱責任者または秘密文書取扱責任者と、立会人の両名が揃ったときのみ実行できること
- 立会人の指名が、指名権限のある主任文書管理者または文書管理者によってのみ行えること
- 各ステップの実行事実が、証拠たり得る形で残っていること
- 本フェーズの職責は、決裁フェーズ、後述の利用フェーズにおける職責とは独立したものとして定義可能であること

E.3 利用フェーズ

（1）文書（オブジェクト）

利用フェーズにおける行政文書の第一の分類として、以下の職責によるものがあげられる。

大臣/長官閲覧文書：	大臣/長官のみが閲覧できる文書
局長閲覧文書：	局長以上の職責の者が閲覧できる文書
課長閲覧文書：	課長以上の職責の者が閲覧できる文書
一般閲覧文書：	担当者も含めすべての職員が閲覧できる文書

第二の分類として組織によるものがあげられる。すなわち、特定の局や課に属する者のみが閲覧可能というものである。

第三の分類としてプロジェクトによるものがあげられる。これは、特定のプロジェクトに属する者だけが閲覧可能というものである。ここで、プロジェクトとは課、局といった組織を横断するような比較的規模の大きなものから、ある課に属する特定の一名のみといった小さなものまで定義可能であり、第一や第二の分類ではカバーしきれない分類を可能とする汎用性の高いものと考えることもできる。

第四の分類として、以下のように取り扱われる文書秘密区分によるものがあげられる。

極秘文書：	一連番号を付与し番号毎に配布先を記録。配布先での複製は禁止
秘文書：	秘文書属性指定者の承認により配布先での複製が可能。ただし、複製を含めた配布部数を記録
取扱注意文書：	関係者以外の閲覧禁止
一般文書：	特になし

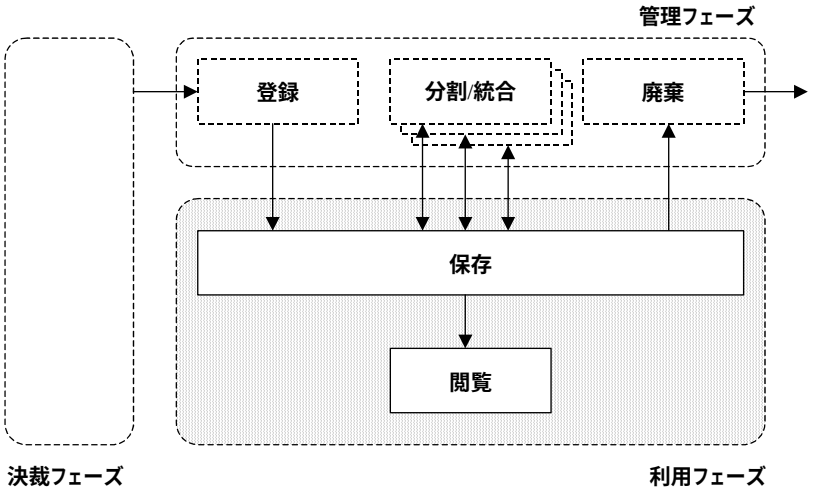
以上述べた第一から第四までの四つの分類は排他的に適用されるものではなく、実際の運用においては同時に適用される場合も多いと考えられる。

（2）職責（サブジェクト）

前述の第一の分類に対応して、すべての文書を閲覧可能な①大臣/長官、局長閲覧文書以下を閲覧可能な②局長、課長閲覧文書以下を閲覧可能な③課長、一般閲覧文書のみ閲覧可能な④担当者の四つの職責が定義可能である。また、第二、第三の分類に対応して、組織、プロジェクト毎に職責が定義可能である。なお、第四の分類に直接的に対応する職責はない。

（3）業務（オペレーション）

利用フェーズは、図Ｅ－４のごとく、行政文書の登録ステップから廃棄ステップまでの間のワークフローと定義する。 なお、文書へのアクセスの発生しない保存ステップも便宜上利用フェーズに含むものとしている。



図Ｅ－４　利用フェーズのワークフロー

保存ステップにおける保存期間としては、1 年未満、1 年、3 年、5 年、10 年、30 年が一般的である。前述のとおり、決裁フェーズを完了した行政文書は内容の変更が原則あり得ないため、この期間内の行政文書に対する操作は閲覧ステップのみである。閲覧ステップにおいては、閲覧要求者の職責と文書の開示属性に基づき、閲覧の可否が決定される。なお、これに加えて、極秘文書の場合には一連番号の付与、配布先の記録が行われる。また、秘文書の場合には配布部数の記録が行われる。

（4）特長

利用フェーズにおける特長としては、以下の要件があることがあげられる。

- 保存中の文書が、何時作成されたものか、改竄されていないかを検証可能であること
- 秘密文書(極秘文書および秘文書)の閲覧においては、たとえ閲覧許可された者であっても、端末上での複製を無許可で行えないこと
- 極秘文書の閲覧においては、万が一漏洩した場合でも、許可された者以外は読めない形式になっていることが望ましい

- 極秘文書の閲覧においては、万が一漏洩した場合の漏洩元追跡のため、個々の文書に一連番号を付与した上で配布し、一連番号と配布先の対となる情報は証拠たり得る形で記録できること
- 閲覧のための検索においては、秘密文書の存在自体が閲覧可能者以外には知られないこと

付録F 一般の OS におけるセキュリティ機能の強化

近年では、一般的用途で用いられる OS においても、OS レベルでのセキュリティ機能が強化されたものが出てきている。ここでは、こうした OS でサポートされるようになってきたセキュリティ機能の一部を紹介する。

① コンパートメント機能、仮想化（Zones）機能

本来の OS 環境の上に複数の仮想 OS 環境（コンパートメント）を動作させる機能である。この機能により、仮想 OS 環境は本来の OS 環境からは隔離された環境として、プログラム、デバイス、ファイルシステム、ネットワーキングなどが利用可能となる。

仮想 OS 環境において、サービスを提供するプログラムが格納されるディレクトリを、本来の OS 環境から read-only で参照可能とする場合、仮想 OS 環境からのファイルの改ざんが不可能となり、システムへトロイの木馬などが仕込まれることを防止することが可能となる。

仮想 OS 環境において、個別のアカウント管理が可能であり、仮想 OS 環境のアカウントとして仮に侵入がされた場合でも本来の OS 環境のアカウントには影響がなく、システムへの侵入を防止することができる。

また、仮想 OS 環境と本来の OS 環境で利用するリソース（CPU やメモリ）の制限をすることが可能であり、サービス使用妨害により、仮想 OS 環境に割り与えられたリソースが使い尽くされた場合でも、本来の OS 環境のリソースに影響がなくシステムを継続して運用することができる。

② スタックバッファオーバーフロー保護機能

この機能は、入力バッファを故意にオーバーフローさせて、攻撃者が実行させたいコードをシステム スタックに配置するといった、スタックバッファオーバーフロー攻撃を防止するために、スタックからコードが実行されることを防止する機能である。ただし、まれに正当なアプリケーションで実行コードをスタックに配置するものがあるため、本機能を有効にするかどうかはプログラム単位で設定できるよう配慮されているものもある。

③ 予測的自己修復（Predictive Self-Healing）

この機能は、CPU、メモリ、PCI の I/O などに関する自己診断の結果をもとに、稼動中に障害の発生したリソースをオフライン化するとともに、障害により停止したサービスを再起動させる機能である。

付録G 諸外国におけるセキュアOSの開発動向

G.１　米国

米国政府によるセキュアOS開発支援は、古くは1960年代から行われており、Honeywell社によるMULTICS（B2クラス）、SCOMP（A1クラス）の開発は有名である。現在でも、Linuxをプラットフォームとする SELinux (Security-Enhanced Linux)の開発は、国家安全保障局(NSA)主導で行われている。SELinuxのセキュリティ・アーキテクチャの中核をなすLOCKというシステムの経費については、開発会社において７年間にわたり詳細な記録が残されており、これに基づき、開発、評価に要した経費が報告されている（参考文献[15]）。

NSAの情報保証研究事務所(IARO：the Information Assurance Research Office)は、Secure Computing Corporation (SCC)社と協力で柔軟性が高い強制アクセス制御のアーキテクチャを共同で開発した。このアーキテクチャは、“Domain and Type Enforcement (DTE)”と呼ばれるメカニズムに基づいており、Logical Co processing Kernel (LOCK)というシステム用に開発されたものである。

NSAとSCCは、“Mach”というオペレーティング・システムをベースに、このアーキテクチャのプロトタイプを構築した。これは、DTMach、あるいはDTOSと呼ばれた。さらにNSAとSCCは、ユタ大学のFlux研究グループと共同で“Fluke research Operating System”へ移植した。この移植中に、このアーキテクチャは、ダイナミックセキュリティポリシーをサポートするよう改善された。

この拡張アーキテクチャは、“Flask”と名付けられている。NSAは現在、Flaskアーキテクチャを、Linuxオペレーティング・システムに移植し、より多くの開発者およびユーザに開放しようとしているが、他の多くのタイプのオペレーティング・システムおよび環境にも適用可能とされている。

A1レベルのLogical Co-processing Kernel (LOCK)（最終的にLinuxに移植）の開発だけでも1987～1992年で、23百万ドル（約24億円）を投じている。また、1994年までにさらに9百万ドル（約9.6億円）を投じ、大幅な改良を加え、メールガード装置に採用されている。したがって、1994年まで７年間にわたり、合計で32百万ドルが投じられたことになる。

G.2　韓国

韓国でのセキュア OS 研究は、90 年代初めに始まり、90 年代後半には活発化していた。韓国のセキュア OS の研究に対する政府支援は、1997 年頃の電子通信研究所(ETRI)におけるセキュア OS による研究から始まるようである。政府は、1997 年からセキュア OS を含む情報保護産業に資金援助を行っていた。IPA の調査（参考文献[16]）によると、韓国政府は、1998～2002 年の５年間で 578 億ウォン（約 60 億円）をセキュア OS 基盤技術と電子署名用 IC データファイル保護技術に投じており、いくつかのセキュア OS 製品を実用化している。その後も政府による支援が継続されている。このような状況の中、2000 年には、ETRI における研究をベースに、Secuve 社が、初の国産商用 OS を商品化した。

2002 年頃までは、Computer Associates 社の e-Trust Access Control がセキュア製品市場を独占していたが、2001 年に TSonNET 社のセキュア OS が、政府機関の認定を受け政府調達されるようになってから国産のセキュア OS 及び関連製品のメーカーがセキュア OS 市場へ参入するようになり、多くのセキュリティ製品が販売されるようになっている（参考文献[17]）。

これには、大学における基礎的研究が盛んであったことも大きな要因と考えられる。大学等の研究機関も政府から資金援助を受け、セキュア OS の研究が活発化し、2001 年には、Hanseo 大学で L4 Linux-MLS が実装された。このあたりから他の大学でも研究が盛んになり、現在までに 30 以上もの大学及び研究所において実装を含めた研究が行われている模様である（参考文献[18]）。

韓国のセキュア OS 製品には、カーネルから書き換えて独自に開発したものもあるが、多くは、既存の Linux、Solaris、HP-UX、Windows 上をプラットフォームとして動作するものが多い。またセキュリティ面の特徴も、米国 NSA の SELinux に類似した機構を取り入れている。
セキュア OS 製品で重要な点は、その評価・認証であるが、韓国情報保護振興院(KISA)等が評価認証を行い、製品の品質を保証している。また、2002 年 8 月に ISO/IEC15408 を同国の評価基準に採用している。

G.３　フランス

フランス国防省は、欧州における著名なセキュア OS 開発・評価会社からなるコンソーシアム（Bertin Technologies 社, Surlog 社, Jaluna 社, Mandrakesoft 社, and Oppida 社）とセキュア OS 開発及び評価契約を締結した。 委託内容は、ISO/IEC15408 による評価で EAL5 を満足する Linux ベースのマルチレベルセキュリティ OS の開発及び評価である。同 OS は、国防省だけでなく、商用アプリケーションの主要な要求を満足することが求められている。

コンソーシアムのパートナーは、OS 設計においては、ハードウエア・パーティショニング技術及び仮想化技術がキーとなるとしている（参考文献[19]）。

同 OS 開発・評価のため、フランス国防省は、2004～2006 年の３年間で 700 万ユーロ（約 9.6 億円）で契約したとされている。

付録H 主なセキュア OS の一覧

製品名	開発元
Compartment Guard for Linux	日本ヒューレット・パッカード株式会社
FreeBSD（5.x 系）	FreeBSD プロジェクト
PitBull	アーガスシステムズグループ社
SELinux	米国 NSA(National Security Agency)ほか
Trusted Solaris	サン・マイクロシステムズ社
Virtual Vault	ヒューレット・パッカード社

用語索引

数字は、各用語を解説しているページ番号を示す。

基礎資料、引用文献及び参考資料

[1] 佐藤慶浩, IT セキュリティソリューション大系 下巻 第 2 編 第５章 第２節 セキュア OS, フジ・テクノシステム, 2004 年.

[2] 原田季栄, セキュアなシステムを作る （3 つの原則に従い OS の機能を強化）, 日経システム構築 2004 年 4 月号 （解説）, No.132, 2004 年.

[3] 中村雄一・上野修一・水上友宏, SELinux 徹底ガイド －セキュア OS によるシステム構築と運用－, 日経 BP, 2004 年.

[4] 米国 Tresys Technology 社 Web サイト (http://www.tresys.com/).

[5] 阿部洋丈他, 静的解析に基づく侵入検知システムの最適化, 情報処理学会論文誌：コンピューティングシステム, Vol.45 No.SIG 3(ACS 5), 2004 年 3 月.

[6] Wagner 他, Intrusion Detection via Static Analysis, 2001 IEEE Symposium on Security and Privacy, pp.156-168, 2001.

[7] 天野 司, Windows はなぜ動くのか, 日経 BP, 2002 年.

[8] 松本 剛, 図解 WindowsOS のしくみ, ディー・アート, 2001 年.

[9] 柿井 弘・中野哲也, よくわかる最新 UNIX の基本と仕組み, 秀和システム, 2002 年.

[10] 三輪信雄・白橋明弘共編, インターネット・セキュリティ教科書（上・下）, IDG ジャパン, 2002 年.

[11] HP-UX 11i システムセキュリティ ホワイトペーパー(PDFHS03-010-01), 日本ヒューレット・パッカード, 2004 年 (http://www.hp.com/jp/hpux).

[12] Stack Buffer Overflow Protection in HP-UX 11i ホワイトペーパー(PDFHS03-025-01), 日本 HP, 2003 年（http://www.hp.com/jp/hpux）.

[13] オペレーティングシステムのセキュリティ機能に関する調査研究プロポーザル, SSR 産学戦略的研究フォーラム, 2004 年.

[14] 韓国における情報セキュリティ政策に関する調査, 独立行政法人情報処理推進機構, 2004 年 12 月.

[15] Richard E. Smith, “Cost Profile of a Highly Assured, Secure Operating System”, Secure Computing Corporation, March 19, 2001.

[16] 独立行政法人情報処理推進機構（IPA）, 韓国における情報セキュリティ政策に関する調査, 2004 年.

[17] KIM, Hyung Chan （金 亨燦）, Trusted Operating System Research & Development in Korea, Security Research Group / Dept. of Info. & Comm. Gwangju Institute of Science and Technology (GIST).

[18] KIM, Hyung Chan （金 亨燦）, Trusted Operating System Research & Development in Korea(II), Security Research Group / Dept. of Info. & Comm. Gwangju Institute of Science and Technology (GIST).

[19] Mandrake 社プレスリリース（2004 年 9 月 23 日付）
http://www.mandrakesoft.com/company/press/pr?n=/pr/corporate/2509