オペレーティングシステムの
アクセスコントロール機能における
セキュリティポリシーモデル

情報処理振興事業協会
セキュリティセンター

目　次

1. OS におけるアクセスコントロール機能

従来の OS においては、特権がコンピュータ全体を支配できる特権利を有する点が問題とされてきた。特権を持つユーザはコンピュータすべての権威を有するために、一旦特権ユーザの権限が不正に奪われると、コンピュータのセキュリティを確保する方法は存在しなかった。セキュア OS においてはこのような一般の OS の問題点を改善するために、強制的アクセスコントロール機能が要求される。セキュア OS でアクセスコントロールを実現するためには以下の 2 つを明確に定義・実装する必要がある[1]。

1. リファレンスモニタ (Reference Monitor)

すべてのデータへのアクセスを正しく監視するために、リファレンスモニタという考え方が導入される[3]。リファレンスモニタとは、コンピュータ上に存在する全データに対するアクセスをコントロールするためのものである。リファレンスモニタは、悪意のあるプロセスによって修正されたり、回避されたりしてはならない。修正や回避されない場合のみ、リファレンスモニタは効果的にアクセスをコントロールすることができる。また、リファレンスモニタは全てのアクセス・リクエストが通過しなければならないシングルポイントでもある。更に、セキュリティ強化におけるリファレンスモニタの重要な役割は、正確に作動しなければならいことである。なぜなら、プロセスの正確な振る舞いの可能性は、プログラムの数、サイズ、および複雑さにつれて減少する。そのため、正確なポリシー強化の最良の保証は、小さく単純で、認識可能なリファレンスモニタが必要となる。

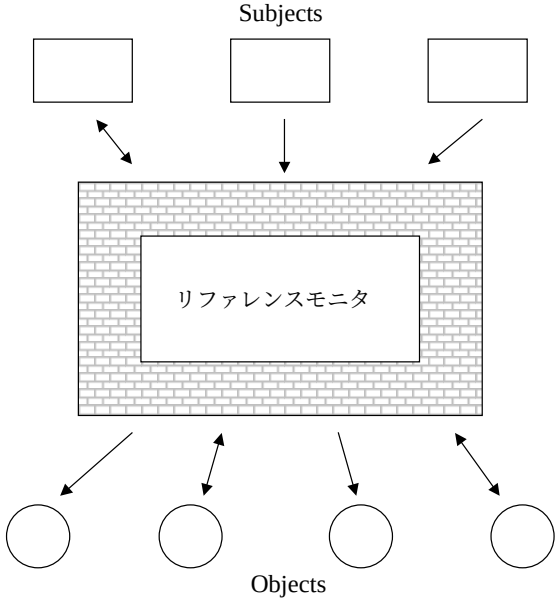


図1. リファレンスモニタ

2. セキュリティポリシーモデル

リファレンスモニタを正確に機能させるためには、「どの主体が、どの対象物を、どのように操作できるのか」を正確にリファレンスモニタに定義する必要がある。リファレンスモニタはこの定義されたセキュリティポリシーに基づき、アクセスの制限を実施する。そのため、定義されたセキュリティポリシーに問題がある場合は、セキュリティを保証することはできない。

トラステッド OS においては、TCSEC において数学的に検証可能なセキュリティポリシーモデルが要求されている。数学的な保証により、初めてセキュリティの保証が可能となる。

いわゆるセキュア OS においては、必ずしも数学的に検証されたセキュリティポリシーモデルが導入されているわけではないが、「どの主体が、どの対象物を、どのように操作できるのか」を明確に定義するための柔軟な枠組みが提供されている。

セキュリティポリシーモデルの詳細について、次節以降において説明する。

2. セキュリティポリシーモデル

アクセスコントロール機能において、セキュリティポリシーとは、「どの主体が、どの対象物をどのように操作できるか」を正確に規定したものをいう。

一般的に、「主体」は「サブジェクト（Subject）」、対象物は「オブジェクト（Object）」と呼ばれている。このサブジェクトとオブジェクトの関係を形式的に定義したものがセキュリティポリシーモデルという。セキュリティポリシーモデルは、多層的セキュリティポリシーと多元的セキュリティポリシーの 2 つに大きく分けられる[1]。

ここでは、この 2 つの分類に従って、それぞれに分類されるセキュリティポリシーモデルを解説する。

[1] ここでは、**エラー! 参照元が見つかりません。**に従って、Multilevel Security Policy を「多層的なセキュリティポリシー」、Multilateral Security Policy を「多元的なセキュリティポリシー」と呼ぶ。

2.1. 多層的なセキュリティポリシーモデル

多層的なセキュリティポリシーモデルは、米国の軍などで適用されるものであり、情報を階層付けて、階層間のアクセスを制限するものである。このモデルはデータに異なる重要度が設定されているような場合に適している。このモデルに分類されるセキュリティポリシーには、Bell-LaPadula、Biba Integrity Model、LOMAC の３つがある。

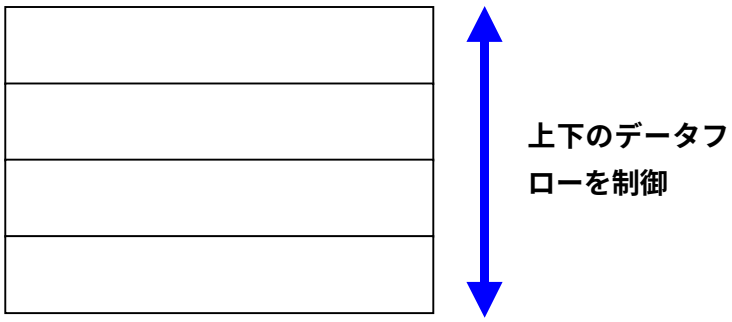


図2.　多層的セキュリティポリシー

Bell-LaPadula モデル

セキュリティモデルとして最も一般的に知られているのは、1973 年に David Bell と Len LaPadula が米国空軍の要請により提案したモデルである[1]。このモデルは米国空軍の機密情報の扱いを数学的な手法でモデル化したものであり、Bell-LaPadula モデルとして知られている。このモデルは多階層セキュリティ（multilevel security）としても知られており、階層で区分された情報間において、「情報が下へ流れない」ことを、その基本条件としたものである。例えば、米軍における文書の機密区分は、図 3 に示すように、４つにラベル付けされていた。そのラベルは、区分外（Unclassified）から始まって、秘（Confidential）、極秘（Secret）、機密（Top Secret）までである。Bell-LaPadula モデルのアクセスコントロールポリシーは単純であり、どのユーザも、自分自身が割り振られたレベルと同等か、それ以下の区分の文書を読むことができる。つまり、「機密」文書を扱うことができるユーザは「極秘」文書も読むことができたが、その逆は許されなかった。Bell-LaPadula モデルの基本は、「情報が下へ流れない」ことであり、情報は上へしか移動しないことである。権限をもつ人が秘密区分を変更しない限り、決して情報が下方へ移動することはない。

図3.　セキュリティ・レベル

Bell-LaPadula モデルでは、これらの情報フローを 2 つの単純な属性で定義している。

- シンプルセキュリティ属性（Simple security property）
  どのサブジェクトも高位のデータを読んではならない。これを NRU（no read up）と呼ぶ。
- スター属性（*-property）
  どのサブジェクトも低位にデータを書き込んではならない。これを NWD（no write down）と呼ぶ。

Bell-LaPadula モデルは、非常に単純な属性のもとで有効に情報の保護が行えるかを示していると同時に、数学的に保護の検証を行えるため、複数の異なる重要度を持つデータを取り扱うシステムを設計するための基礎として数多く利用されている。

Biba Integrity モデル（Low Water-Mark Model）

Bell-LaPadula モデルは、情報の守秘性を確保するために利用することが可能である。すなわち、Bell-LaPadula モデルにおいては、情報の不適切な開示の有無を検証することが可能である。しかしながら、情報の守秘性だけでなく、完全性も重要である。Biba は、情報の完全性の重要さを指摘して、情報の不適切な修正を防ぐためのモデルを 1975 年に提案した。Biba のモデルは、Bell-LaPadula モデルの対となるものである。Bell-LaPadula モデルとは違い、「情報が上へ流れない」ことを基本としている[4]。

図4.　セキュリティ・レベル

Biba は、Bell-LaPadula モデルのセキュリティ・レベルに類似する「インテグリティ・レベル」（完全性レベル）を定義し、以下の単純な 2 つの属性で情報の安全性を保護することが可能であることを示した。

- シンプルインテグリティ属性（Simple integrity property）
  サブジェクトは、下位のデータに限り修正することができる。
- インテグリティ　スター属性（Integrity *-property）
  もしサブジェクトが、同じインテグリティ・レベルのオブジェクトへの読み込み権限を持っていれば、そのサブジェクトは上位のオブジェクトに限り修正することができる。

サブジェクトが下位のオブジェクトを読んだり、上位のオブジェクトに書いたりすることは、高位のオブジェクトが低位のデータに汚染される危険があるので、許可されるべきではない。これは、low water mark モデルとして表現され、あるオブジェクトの完全性は、その作成に寄与した全てのオブジェクトファイルのうち最も低位のものと同じ水準になることを指している。

Biba は、Bell-LaPadula モデルが対象としていない、情報の完全性の問題に取り組んでいるが、逆に Biba モデルは守秘性を対象としていない。現在では、セキュリティシステムにおいて、守秘性とインテグリティを結合しつつ、これらの妥協を許さないモデルを達成することが求められている。

LOMAC モデル

LOMAC は NAI の Tim Fraser によって提案された強制的アクセスコントロールモデルである[5]。このモデルは Biba によって提案されたモデルの問題点を拡張したモデルであり、実際のオペレーティングシステム環境に、Biba モデルを適用するための柔軟性を与える拡張を加えたものである。Biba モデルと同様に情報の完全性を保護するためのモデルであり、データの完全性保護のためにオペレーティングシステム上に実装され、すでに実用されているものである。

Biba モデルには、実際のオペレーティングシステム環境に適用した場合に自己廃止問題（Self-Revocation Problem）という問題が存在することが知られている。この問題は Biba モデルの“降格(demote)”の振る舞いによって、あるサブジェクトが予期しないオブジェクトを修正することで、“降格”してしまう状態で生じる。例えば、高いレベルを持つサブジェクトがあるオブジェクトを作成し、次に低いレベルのオブジェクトを参照する場合に生じる。Biba モデルによれば、上記の一連の操作によってサブジェクトは低いレベルに“降格”させられる。従って、低いレベルにあるサブジェクトは自分で生成したオブジェクトを修正することはできない。作成したオブジェクトは高いレベルのままである。これは保護の観点からすれば一般的な振る舞いである。例えば低いレベルのオブジェクトを参照することによって、サブジェクトは悪意を持ったコードによって汚染される可能性がある。汚染後に生成した高いレベルのオブジェクトに書き込みを行うことで、高いレベルのものに汚染を広げる可能性がある。

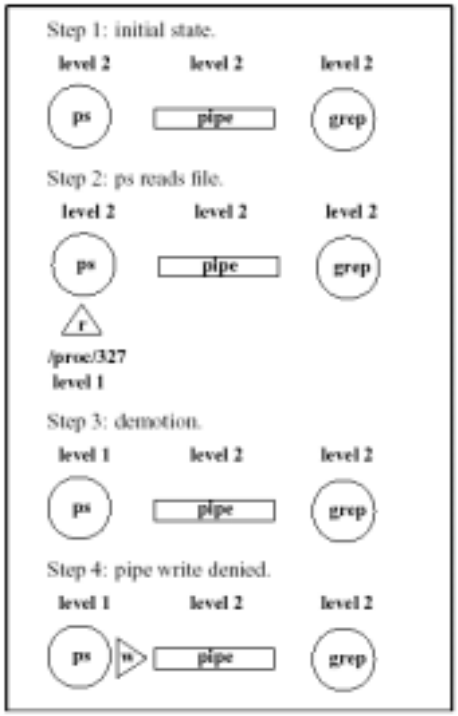


図5.　Pipe における自己廃止問題

このような状況は、避けねばならないが、アプリケーションの立場からは、非常に問題である。オペレーティングシステム環境においては、複数のプログラムが連携して動作する。連携は一連の動作として定義されるため、その過程におけるアクセスの拒否は考慮されない。しかし、Low-Water Mark Model の適用された環境においては、自己廃止問題により、プログラムの連携は非常に制限されてしまう。

Tim Fraser らは、上記の自己廃止問題に取り組み、Biba モデルの改良を行い、LOMAC と呼ぶモデルを開発した。Fraser らは自己廃止問題のほとんどが、図 5 に示すように、名前なしパイプやシェアードメモリなどを含む IPC（Inter Process Communication）において生じていることを突き止め、自己廃止問題を解決するモデルを開発した。Fraser らは“ group ”という概念を Low-Water Mark Model に導入した。“ group ”は連携して動作する一連のプロセスをひとつの job としてカプセル化するための概念である。“ group ”として定義された一連のプロセスにおいては、IPC を通じた読み書きを自由に行うことができる。これにより、自己廃止問題を解決することが可能である。Fraser らはこの修正によっても Biba モデルのセキュリティ強度を低下させないことを数学的に証明している。

2.2. 多元的なセキュリティポリシーモデル

一方、多元的なセキュリティポリシーモデルとは、情報の重要度を順序付けるのではなく、コンパートメントと呼ばれる小さな部屋に区切り、情報へのアクセスをコントロールするモデルである。このモデルにおいては、多層的セキュリティポリシーモデルのように、階層的な情報をコントロールするのではなく、コンパートメント間の横方向の情報フローがコントロールされる。厳密に順序付けのある情報を保護するためには不向きであるが、実世界においては、必ずしも情報が順位付けられている場合ばかりではなく、緩やかに区分される情報を扱うケースも多いため、このようなモデルの必要性がある。緩やかに区分された情報へのコントロールを多層的セキュリティポリシーモデルで記述することは困難である。

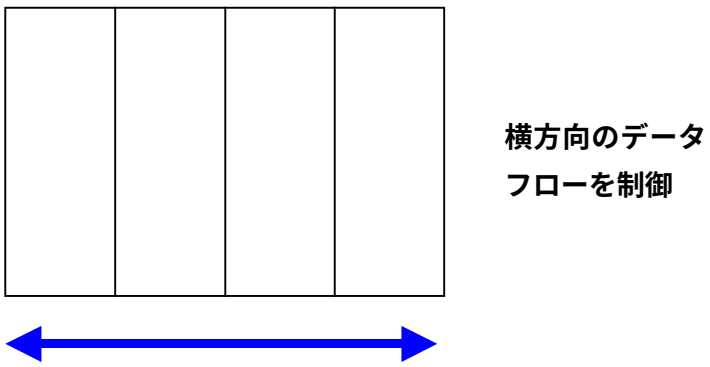


図6.　多元的セキュリティポリシーモデル

このように多元的に情報をコントロールしようとするモデルとしては、Clark-Wilson モデル、Chinese Wall モデル、DTE モデル、RBAC モデルなどが挙げられる。以下ではそれぞれのモデルについて説明する。

Clark-Wilson モデル

1987 年に David Clark と David Wilson は多層的なセキュリティポリシーモデルと多元的なセキュリティポリシーモデルの比較を主題とした論文を提出している[6]。この論文においては、一般の商業活動における重要な問題はデータの完全性を確保することであると指摘している。この指摘は、多元的なセキュリティポリシーを構築するきっかけとなり、以後多元的なセキュリティポリシーモデルが研究された。

Clark-Wilson のセキュリティポリシーモデルにおいては、完全性の保護のためには以下の 2 つ基本的なコンセプトを扱うことが重要であると指摘している。

(1) 定型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ためには、制限された方法でしかデータを操作することができないようにするための手続きを実装することが重要である。この手続きは、サブジェクトがデータにアクセスする前に、アクセスの正当性を検証するものである。Clark と Wilson はこの手続きを「定型化されたトランザクション(well-formed transaction)」と呼んでいる。この手続きは、データが任意にアクセスされないようにする。単純なアクセス権でなく多様で定型化されたトランザクションによって、商業上のセキュリティポリシーは形成されなければならない。

(2) 責務の分離**(separation of duty)**

たった一人の管理者により、すべてのオペレーションが行える状況は、詐欺を引き起こす場合がある。例えば、たった一人によって、注文から、配達、支払いまでを行うことができる場合、支払方法に問題があり、他人が支払い業務を偽装することができるのであれば、注文から配達を偽装して、代金を不正に入手することができる。複数の人でオペレーションを分割することは、このような不正を阻止するために有効である。Clark と Wilson は、「責務の分離(separation of duty)」を提案している。権限を分割することで、例え部分的に権限が奪取されたとしても、全体としてのオペレーションは、矛盾なく実行することが可能である。権限の分離は、商業上のデータの完全性を保護するための基本的な原則である。

上記の 2 つの原則は、商業においてデータの完全性を保護するための大原則であり、Clark と Wilson はこれらの大原則の形式的なモデルを定義している。Clark-Wilson のモデルにおいては、以下の 4 つの要素データが登場する。

- CDI (Constrained Data Items) : 完全性や安全性を必要とするデータ
- IVP (Integrity Verification Procedures) : システム内のすべてのデータがセキ

ュリティの定義に従っているか確認する手続き

- WFT (Well Formed Transactions) : CDI を現在の正しい状態から別の状態に遷移させる
- UDI (Unstrained Data Items) : 完全性や安全性を必要としないデータ

下図においては、CDI が処理される場合は、IVP によってまず正当な処理かどうかが検証される。もし正当な処理である場合は、WFT によって CDI のデータは別の状態に繊維され、再度正当性が検証される。これにより常にデータの完全性が検証され、データの完全性が保たれることとなる。

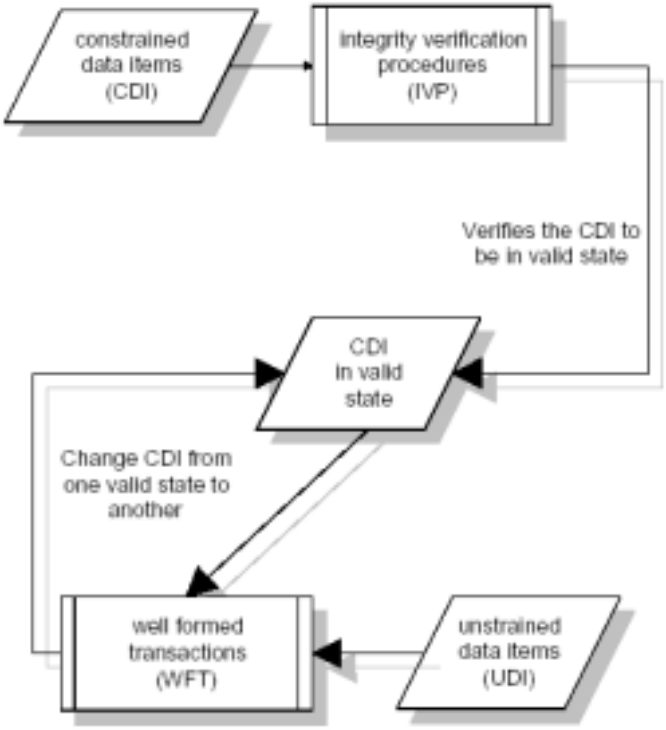


図7.　Clark-Wilson **モデルにおける完全性の検証**

Chinese Wall モデル

Brewer と Nash は、ある特定の商業用途での情報アクセス保護の必要性を説き、そのためのモデルを定義した[7]。このモデルの対象となるのは、関心事が衝突しそうな弁護士、医療、投資家、あるいは会計事務所の人々である。これらの人々が活動する際に生じる情報へのアクセスの問題点を解消するモデルが、Chinese Wall モデルである。このモデルにおいては、「関心事の衝突」を情報保護の基本としている。つまり、競合する会社の情報があった場合に、その両方の企業の情報にアクセス可能な状況は、本来避ける必要がある。競合する会社間での不適切な情報はコントロールされる必要がある。Chinese Wall モデルは、3 つの抽象レベルから構成されることによって始まる。

- **オブジェクト**（individual objects）
  低いレベルでは、ファイルのような要素オブジェクトである。各々のファイルは、１つの企業だけの情報を保有する。
- **企業グループ**（Company Datasets）
  次のレベルでは、全てのオブジェクトは各企業に関して互いにグループ化される。
- **競合クラス**（Conflict Interest Classes）
  次の高いレベルでは、競合企業のオブジェクト・グループの全ては、クラスター化される。

各オブジェクトは、ユニークな企業グループに所属し、各企業グループは、あるユニークな競合クラスに含まれる。ある競合クラスは、１つあるいは複数の企業グループに含まれてもよい。

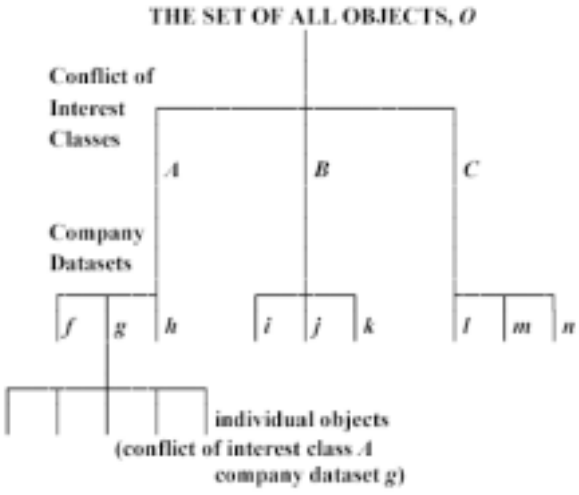


図8.　Chinese Wall **の基本構成**([7]**より引用**)

例として、図 9 に示すように、食品関連企業 A 社と B 社のデータが格納され、さらに、銀行として、C 銀行、D 銀行と E 銀行、さらに航空会社 F 社の情報が格納されている場合を考える。これらの企業の情報は、6 つの企業グループ（１グループに１企業）と 3 つの競合クラスになる。{A 社, B 社}, {C 銀行, D 銀行, E 銀行}, そして {F 社}である。

Chinese Wall Security Policy モデルでのアクセスコントロールポリシーは単純である。これらの情報にアクセスしようとする人物は、初期状態ではどの情報にもアクセスできる。しかし一旦情報にアクセスすると、それ以降は競合クラスに属するどのオブジェクトにもアクセスすることはできない。競合クラスに属さない情報であれば、アクセスは許可される。上記の例においては、A 社の情報にアクセスした人は、競合クラスに属する B 社の情報にはアクセスできなくなる。さらに D 銀行の情報にアクセスすると、競合状態にある C 銀行、E 銀行の情報へはアクセスできなくなる。

Chinese Wall Security Policy モデルは、ほとんどの商用モデルが完全性にフォーカスしているのに対して、守秘性を対象としている。また Chinese Wall Security Policy モデルは英国の証券取引において、その実装が法的に規定されている。

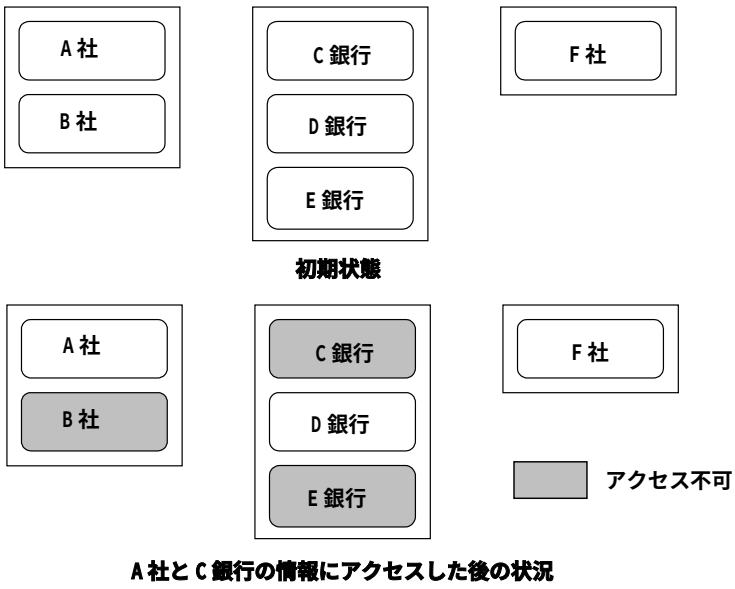
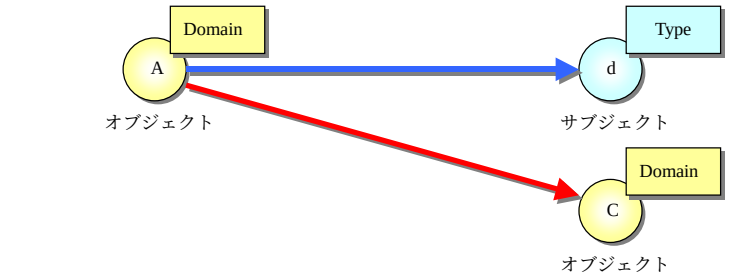


図9.　Chinese Wall **モデルのアクセスコントロール例**

DTE (Domain Type Enforcement) モデル

DTE は、Type Enforcement (TE) の拡張形式である[8][9][10]。TE は Boebert 及び Kain らにより提案されたアクセスコントロールモデルで、初期には LOCK システムというセキュア OS において採用されていたものである。DTE は、テーブル指向のセキュリティポリシーモデルであり、アクセスコントロールのルールをテーブル上に定義し、それに基づいてアクセスコントロールが実行される。DTE においては、システムをサブジェクト（アクティブエンティティ）とオブジェクト（パッシブエンティティ）の集合としてみる。DTE においては “ ドメイン ” と呼ばれる属性が、それぞれのオブジェクト（ファイル、メッセージ、シェアードメモリ）に割り当てられる。アクセスコントロールのためのテーブルとしては、ドメイン定義テーブル（DDT）とドメインインタラクションテーブル（DIT）の 2 つがある。DDT は、サブジェクトとオブジェクトの間の許可されたアクセスモード(read, write, execute,…)を表現するために利用される。また DIT は、ドメイン間のアクセスモード（signal, create, destroy….）が定義される。これらのテーブルを参照して、システム実行時にアクセスのコントロールが行われる。テーブルにおいて認証されないモードでのアクセスは拒否される。



Type

Domain	a	b	c	d	e
A	rwx	rw	r	w	x
B	x	r	r	r	rwx
C	r	r	r	r	-
D	r	r	r	r	r
E	x	x	rx	rwx	r

DDT

Domain

Domain	A	B	C	D	E
A		scd	s	s	s
B	s		s	s	s
C	s	sc		s	d
D	s	s	s		s
E	s	s	s	rwx	

DIT

図10.　DTE のアクセスコントロール例

DTE においては、動作しているプロセスに “ Domain “ 定義を付加し、ファイルなどには ” Type “ 定義を付加する。” Domain “ には、どの ” Type “ のファイルに対して、どのようなアクセス権を持つかが定義される。そのためある ” Domain “ のプログラムがある ”Type“ のファイルに削除するためには、そのプログラムの属する ”Domain“ において、ファイルの属する ” Type “ へのアクセスが許可されている必要がある。

DTE は、NSA の Security-Enhanced Linux などに実装されている。DTE のベースとなっている TE は、米国の特許をもっており、その利用が保護されており、技術的詳細については公開されていない。

RBAC (Role Base Access Control) モデル

セキュリティを考える上で、「どの役割（Role）の人がどのような権限を持つか」は重要である。そのために、ロールという概念が導入された[11]。

アクセスは、コンピュータ資源に何かをする能力である（例えば、利用、変更あるいは閲覧する等)。そして、アクセスコントロールは、何らかの方法で能力が明示的に可能になるか、あるいは制限される手段である。コンピュータによるアクセスコントロールは、誰、あるいは特定のシステム資源にアクセスするプロセスを規定するだけでなく、許可されるアクセスのタイプも規定可能である。

役割に基づいたアクセスコントロールにおけるアクセス決定は、個々のユーザが組織の一部として持つ役割に応じて行われる。ユーザは、割り当てられた役割（例えば、医者、看護婦、患者、事務員など）を引き受ける。そして、その役割を定義する過程は、組織がどのように機能するかの分析が必要になる。

アクセス権は、役割名によってグループ化する。また、資源の利用は、関連する役割を引き受けるために認可された個人に制限される。例えば、病院内では、医者の役割は、診察、治療、薬の処方、手術等を含めることができる。また、研究者の役割は、研究のための匿名の臨床情報を集めることに制限されているかもしれない。アクセスをコントロールする役割の利用は、組織に合わせたセキュリティポリシーを策定・強化し、セキュリティ管理プロセスを合理化するための有効な手段であるといえる。

従来の UNIX システムにおいては、管理者としてのルート権限は、任意のファイルに読み書きし、プログラムを全て実行し、任意のプロセスを停止することができるなど、全てにおいて強力である。

ロールベース・アクセスコントロール（以下 RBAC）は、ルート権限をもつ管理者のモデルの代わりでもある。また、RBAC は、最小の特権のセキュリティ原理とも一致する。それは、どのユーザもそのユーザのジョブ実行のために、必要以上の多くの特権を与えてはならないという考え方である。更に、RBAC により、強大な管理者権限の能力を分離し、各役割に応じた権限を割り当てることが可能になる。これは、様々なセキュリティポリシーを可能にする。そして、アカウントは、セキュリティ管理、ネットワーキング、ファイアウォール、バックアップおよびシステム管理のような特定な目的の管理者のために、生成することができる。もちろん、権限を 1 人に集中させれば、従来のルート権限を持つ管理者同様のアカウントを生成することもできる。

ルート権限をもつ管理者と RBAC を考察する事例としては、1 個のパス・キー（マスター・キー）で建物へ入り、全ての部屋にアクセス可能な会社を想像してほしい。これは、ルート権限をもつ管理者のモデルと似通っている。つまりルート・パスワードを持つ者なら誰でもアクセスすることが可能である。もし、その会社がサーバー室、ネットワーク・パッチ室、およびボイラー室のようなユーティリィティ・エリア用の個別のキ

ーが必要であれば、RBAC モデルに類似する（図 11 参照）。このとき、これらのエリアに責任を持つ従業員は、それらの仕事の役割によって、個別のキーを持っているのである。

RBAC は、セキュア OS だけではなく、データベースなどアプリケーションの分野においても適用されており、アクセスコントロールのモデルとして非常に注目されている。

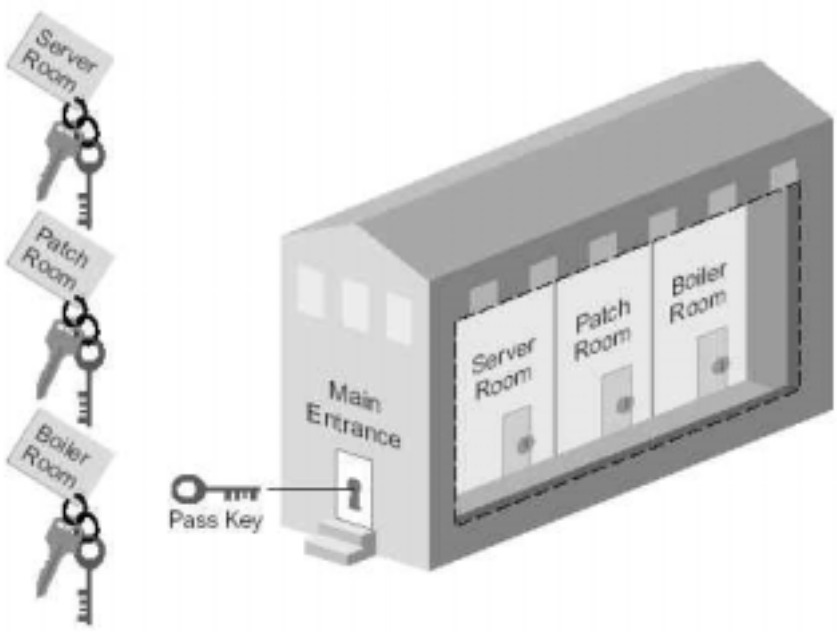


図11.　RBAC

2.3. セキュリティポリシーモデルの位置け

以上､説明してきた個々のセキュリティポリシーモデルの位置付けが明確になるように、ここでは、に示すように、2 つの評価軸上にそれぞれのセキュリティポリシーモデルをマッピングする。の縦軸は、モデルが多層的か多元的かを示す。横軸は、守秘性に適したものか、完全性に適したものかを示す。

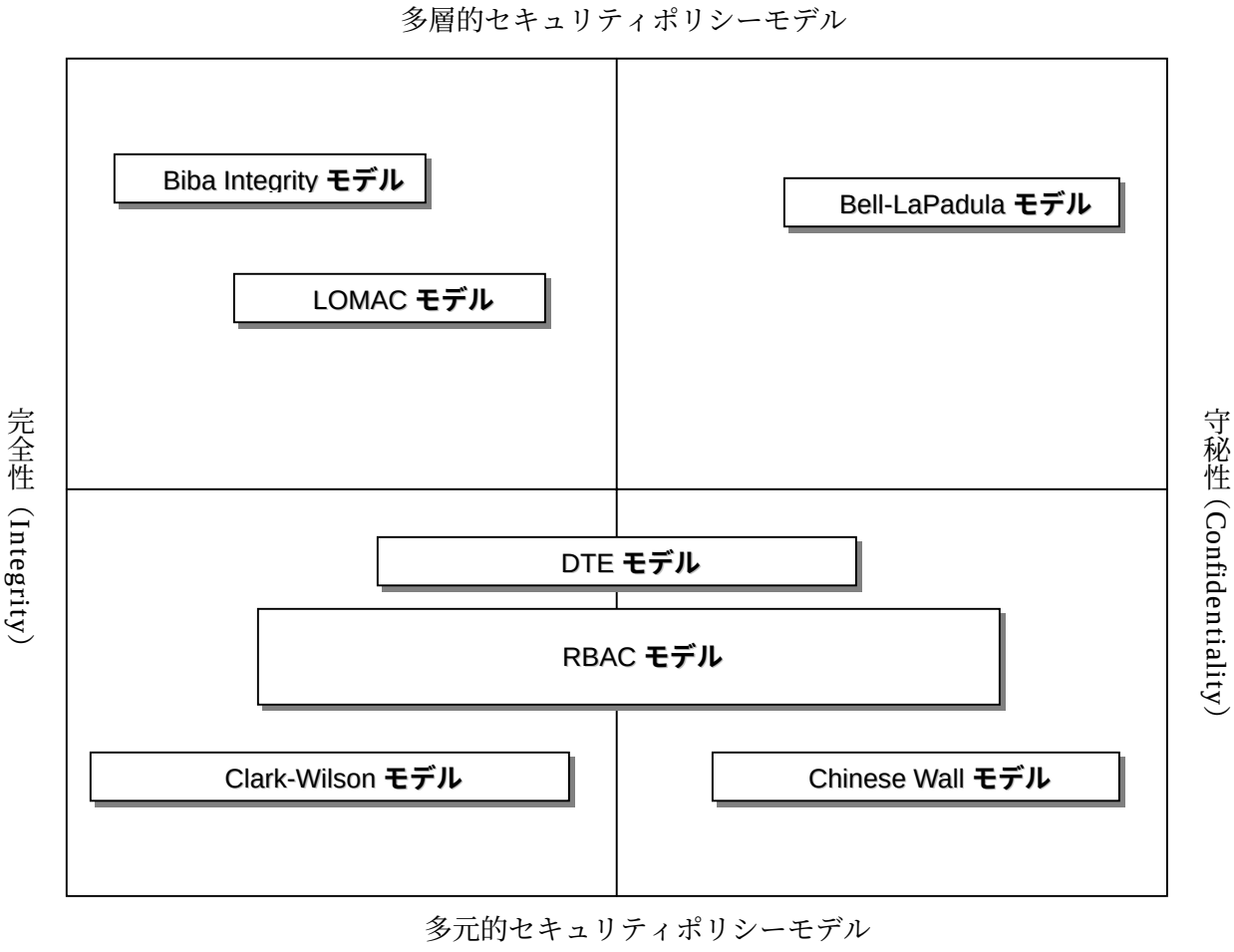


図12.　**セキュリティポリシーモデルの位置付け**

参考文献

[1] Bell D, LaPadula L, “Secure Computer System: Mathematical Foundations and Model”, MITRE report MTR 2547, Nov. 1973.

[2] Charles P. Pfleeger, “Security in computing second edition”, Prentice hall, 1997.

[3] J Anderson, “Computer Security Technology Planning Study”, ESD-TR-73-51, U.S. Air Force Electronic Systems Division, 1973.

[4] K. J. Biba, “Integrity Constraints for Secure Computer Systems”, Mitre Technical Report ESD-TR-76-372, Jul. 1975.

[5] T.Fraser, “LOMAC: Low Water-Mark Integrity Protection for COTS Environments”, in Proceedings of the 2000 IEEE Symposium on Security and Privacy, IEEE, Computer Society Press, pp230-245, 2000.

[6] David D. Clark , and David R. Wilson, “A Comparison of Commercial and Military Computer Security Policies”, Proc. 1987 IEEE Symposium on Security and Privacy, pp. 184-194, Oakland California, Apr. 1987.

[7] David F.C. and Michael J. Nash, The Chinese Wall Security Policy, The Symposium on research in security and privacy, 1-3 May 1989, OAKLAND, CALIFORNIA, pp. 206-214.

[8] Phil Kearns and Serge Hallyn., Deriving Tools to Administer Domain and Type Enforcement., *Proceedings of the 15th Usenix System Administration Conference*, (Dec 2001), pp 151-155.

[9] Kenneth M. Walker, Daniel F. Sterne, M. Lee Badger, Michael J. Petkac, David L Shermann, Karen A. Oostendorp, *Confining Root Programs with Domain and Type Enforcement(DTE)*, Sixth USENIX UNIX Security Symposium, 1996.

[10] Lee Badger, Daniel F. Sterne, David L. Sherman, Kenneth M. Walker and Sheila A. Haghighat, *A Domain and Type Enforcement UNIX Prototype*, Fifth USENIX UNIX Security Symposium Proceedings, Salt Lake City, Utah, June 1995.

[11] Sun Microsystems, INC., White paper “RBAC in the Solaris Operating System”, Apr. 2001.