CSS'99　（1999 年 10 月）

セキュリティ・アーキテクチャに基づくＩＴ設計

THE INFORMATION TECHNOLOGY DESIGN BASED ON SECURITY ARCHITECTURE

佐藤　慶浩
Yoshihiro Sato

日本ヒューレット・パッカード株式会社（〒168-8585 東京都杉並区高井戸東 3-29-21）
Hewlett-Packard Japan, Ltd. (3-29-21, Takaido-higashi, Suginami, Tokyo, 168-8585 Japan)

情報システムの構築の際にセキュリティの問題を解決する場合、技術的コンポーネントのそれぞれに局所的な対策を施すことは問題をかえって複雑にする。これを解決するために、システム全体を定義するセキュリティアーキテクチャに基づいたＩＴ設計が有用であることを紹介する。

When you resolve security issues for building information system, it makes issues more confused that solutions are given to technical components for each. In order to avoid this happens, the information technology design based on security architecture defining overall system is efficient.

１．はじめに

異機種の分散されたコンピューティング環境の情報システム構築においてセキュリティ対策を考える場合、全体のフレームワークを設計して、それに基づいて個々の要素を位置づけ実装していく考え方を CSS'98 で明らかにした。ここでは、その要素の具体的な設計について考えることにする。

２．認証・認可

認証とはアクセスの主体を考えるものであり、認可はその主体がアクセスする客体であるなんらかのデータへのアクセス制御を考えるものである。

したがって、主体と客体を明確にすることから認証と認可の設計は始めなければならない。

単一種類の装置の認証・認可の設計は、装置が主体と客体を定義しており、その実装技法に従って条件を設定するだけでよいが、異機種・分散コンピューティングにおいては、様々な技術を統合する階層における主体と客体の明確化を含めて設計する必要がある。

これを効率的に実施するために、以下のような手順で進めることができる。

２．１　主体の定義

設計の対象とする情報システム全体を一つのブラックボックスと見なして、そのブラックボックスをとりまく主体を役割の観点で種別化する。これは、技術や製品によらず、役割としてだけ整理していく。仮に現実の運用で複数の役割を一人の人、あるいは、一つの部署が担当せざるを得ないとしても、役割として分離できるのであれば、分離して設計する。役割は、その後、なんらかの物理的な人か組織に割り当てるものとして設計する。

ブラックボックスの中に、人がいるように見える場合があるかもしれない。その場合は、その人の部分をブラックボックスの表面に引き出し、外側に出すことによって、あくまでも役割をブラックボックスの外で考える。

また、対象のシステムがそれ以外のシステムと連携しており、そちらのシステムに対しては制御が及ばないことがある。その場合は、それらのシステムの役割を定義し、対象のシステムへの主体として取り扱うことができる。

２．２　客体の定義

ブラックボックス内で処理されるデータをセキュリティの観点で種別化する。セキュリティの観点は、一般的な機密性・完全性・可用性で検討すればよい。この場合も、ブラックボックスを構成する個々の装置にとらわれずに、種別を設計する。

例えば機密性であれば、極秘・機密・公開データのような定義をする。近年は顧客プライバシなどの問題もあるため、単純な段階的レベルによる定義ができな

い場合もある。その場合にはコンパートメント（種別）を用いたグルーピングによる定義も使用する。
コンパートメントとレベルの設計では、個々のコンパートメントに対して、それぞれのレベルを定義する。レベルをコンパートメント共通にする必要は必ずしもない。コンパートメント間は独立であり、レベルは上位が下位を包括するように定義する。
インターネットの匿名性を対象とする場合には、コンパートメントのひとつとして「無認証」を定義すると違和感なく匿名主体を収容できる。

以上の他に、完全性・可用性についての定義を必要に応じて設計する。これらの２つは、一般的に単純な段階的レベルで設計するのが、実装上好ましい。

本来、客体の定義は以上であるが、客体の格納システムの種別を定義すると、システム構築に役立つ。
格納システムの種別は、基本的には客体の定義と同じであるが、客体の定義をより簡略化し、どの客体は、どのシステムに格納するべきかを明確にしておくことで、実際の構築に際してシステムの要件を容易に導くことができるようになる。

２．３　アクセス制御

主体と客体が定義されたので、あとは、すべての主体とすべての客体との間のアクセスの許可・不許可を権限として定義する。
ここで許可された権限アクセス以外のすべてを無権限アクセスとして、後の監視・検出で扱う。

主体と客体を先述の手順で設計した場合、アクセス制御は、２次元の表によって整理することができる。許可・不許可はアクセス手段によって異なるので、それをセル内で設計する。すなわち、作成・書き込み・読み出し・改変・追記・削除などごとに権限を決定する。
アクセス手段のうち、客体が監査証跡に関する場合、追記アクセスを他のアクセスと区別することは重要である。監査証跡の完全性は、監査要素の完全性に直接影響するためだからである。

２．４　認証手段

認証手段についての要件を定義する。
単純に全システム共通の要件を定義することもできるが、主体の重要度に応じて認証手段を定義する方が、効果的にコストを投入することができる。
定義は、知識証明・所有証明・場所証明・動作証明・生体的特徴証明などから、いくつの証明要素を要件とするのかなどで定義する。

２．５　アクセス制御手段

認証手段の要件定義と同様であるが、アクセス制御手段は実装技術に依存することが多く、最終的に制御が実現できればよいので、個々の装置やソフトウェアの要素ごとに定義することになる。

２．６　リスクの洗い出し

以上の手順で、認証・認可についての設計が完了するが、ここで設計したうちのいくつかは、実際の技術実装において分離して実現できない部分があるかもしれない。
これはこの設計が無意味あるいは冗長だったことを意味していない。
設計上分離されていたものが、技術的に分離できなかったことは、技術的なリスクとして認識することができる。多くの場合、技術的に実現できなかったことは、その回避策を運用上検討できる。その運用策を採用するかどうかは、すなわち、そのリスクを受け入れるかを決定することを意味する。
例えば多くの場合、ネットワーク通信装置は識別・認証機構を装備しておらず、アクセス・パスワードによってだけ保護される。これは技術的に識別・認証の機会を失うが、パスワードをデュアル・ロック化することでリスクを軽減できる。そのようにして、リスクの顕在化に役立つものと思われる。

３．監視・検出

次に、制御機構のうち監視・検出要素の設計について考える。

３．１　手順的段階

セキュリティ対策は、予防・保護・監視・対応の４段階で構成される。さらに監視は、能動的なものを検査、受動的なものを検出として区別する。また、監視と対応は、分析によって連結する。
以上を言葉の定義として検めると以下のようになる。

・予防：無権限の行為を未然にさせないようにすること
・保護（あるいは防止）：無権限の行為を不可能か困難にさせること
・検査：保護の脆弱性を能動的に検証すること
・検出：無権限の行為を見つけ出すこと
・分析（あるいは調査）：検出した内容を調査すること
・対応：行為の被害や影響を最小限にするべく対処すること

対応は、監視の結果に対応するものであるが、その具体的手順を事前に計画・訓練しておくことが肝要である。
以下のような手順を設計することができる。

(1) 警報監視
(2) 警報内容分析
(3) 警報内容報告

(4) 対応内容検討
(5) 対応内容承認
(6) 対応作業実施
(7) 対応作業報告
(8) 効果確認
(9) 効果報告

以上は、検出直後になるべく迅速に対処しなければならない事柄である。
その後、さらに以下の実施が必要となる。

(10) 原因究明
(11) 再発防止対策検討

これらは時として困難な作業であるが、当然のことながら完遂する必要のある作業である。

３．２　空間的ドメイン

情報セキュリティを最終的にはデータへの無権限アクセスからの保護として考えた場合、データはシステムに格納され、システムにはネットワークを介してアクセスすることを前提とし、セキュリティ対策のドメインを設計することができる。
すなわち、ネットワーク、システム、データという３層のドメインによって、データを無権限アクセスから守る。このセキュリティ・ドメインのそれぞれにおいて、予防・保護・監視・対応の策を設計する必要がある。
例えば、侵入検出といった場合、ネットワーク領域だけにその実装を頼る場合が見受けられるが、それだけでは、その侵入によって、どのシステムが無権限のアクセスを受けたのかが特定できなくなる恐れがある。
そのような特定ができないと、ネットワークへの侵入によって、その影響を受ける可能性のあるすべてのシステムを停止して調査しなければならなくなる。全体システムとして冗長構成がとられていたとしても、この調査のためには、冗長機器も含めて業務を停止することになる。
したがって、セキュリティ対策はドメインごとに施すことで、多重化により強化されるばかりではなく、対応作業による影響を軽減することにも役立つものである。

３．３　リスクの洗い出し

監視・検出の設計は、手順的段階と空間的ドメインの表により整理することができる。
認証・認可のときと同様に、埋まらないセルや実装が区別されないセルは、リスクとして取り扱うことができる。

４．まとめ

以上のようにセキュリティ・アーキテクチャで定めたフレームワークをもとにＩＴセキュリティの設計をすることで、全体としての網羅性を高めることができる。
仮に、個々の要素の実装技術から組み合わせて全体システムを仕上げたとすると、要素間の隙間として存在するリスクを顕在化させることは容易ではなくなるだろう。
その意味でも、セキュリティ・アーキテクチャに基づくＩＴセキュリティ設計は効果的であると思う。

今回は、セキュリティ・アーキテクチャの中でも比較的整理しやすい要素を取り上げて考察した。
また、この考察の過程において、このアーキテクチャが技術的なフレームワークになっていることから、アクセス制御の許可・不許可の判断の基準となる行為については言及できないことがわかった。このことは、アーキテクチャの目的として十分であるが、業務用途と私的利用の対比を整理する過程で、その基準をフレームワークとして組み込むきっかけが見えてきたので、今後はその統合を試みたいと考えている。