マルチメディア通信と分散処理　94－19
分散システム／インターネット運用技術　14－19
（１９９９．７．１６）

POLICYCOMPUTING™のセキュリティポリシーへの適用

菅野　政孝*　坂田祐司*　小熊　慶一郎*　田中　俊介*　白鳥則郎**

＊　株式会社 ＮＴＴデータ
＊＊　東北大学電気通信研究所

あらまし:
クライアントサーバシステムでは，クライアント(端末)での処理内容が多く，サーバが分散されているため，システムの運用管理コストが大きいという問題がある．これに対処するためマスターポリシー(管理者にとって設定が行いやすい表現形式のポリシー)によって情報システム内の全てのリソース(機器，サーバプロセス，ユーザなど)を一元的に管理するための仕組み POLICYCOMPUTING™ を提案してきた．今回，POLICYCOMPUTING™ の適用例の１つとしてセキュリティポリシーを取り上げ，その適用方法を検討し実装を試みた．その結果，セキュリティのアクセス制御に関するポリシー制御に適用可能であることの見通しが立った．

キーワード:　ディレクトリ，ネットワーク管理，セキュリティポリシー，アクセス制御

A study of POLICYCOMPUTING™ applied to the Security Policy

Masataka Sugano*, Yuji Sakata*, Keiichiro Oguma*, Shunsuke Tanaka*, Norio Shiratori**

* NTT DATA Corporation
** Tohoku University

Abstract:
It is a problem in the Client/Server System that the management cost is heavy, because of complex applications of clients and distributed servers. We have proposed the POLICYCOMPUTING™, that manages all resource (hardware, service-program, user-account, etc.) composing the information system with Master-Policies (policies written in user-friendly languages) at a single point. In this paper, we described a case of POLICYCOMPUTING™ which is applied to the security policies.

Key words:　Directory, Network Management, Security Policy, Access Control

1. はじめに

ホスト系システムと比較してクライアントサーバ系システムではクライアント側の処理内容が多いことや，サーバが分散されているといった特徴がある．このため，クライアント側の運用管理のコストが大きくなるという問題点が指摘されている[1,2]．コストが大きくなる理由としては以下のような点があげられる．

- クライアント端末の設定項目数が多い．
- 端末毎に別々の設定をしなければならない設定項目が多い(例: ホスト名)．
- 機能毎にサーバを分けるため，同一の設定内容を複数のサーバに配布しなければならない(例:ユーザアカウント)．
- サーバを階層的に配置するため，目的のサーバを探し出せる仕組みが必要である(例: Web サーバ)．

また，管理できる人材が不足しており，一部の人に管理業務を集中的に行わせざるを得ないこともコストを増大させる一因である[3]．

運用管理コストを削減するためには多様なアプローチがあるが，我々は「各リソース(機器，サーバプロセス，ユーザなど)の設定に必要な運用管理コスト」に注目しこれらを削減する方法として POLICYCOMPUTING を提案してきた[4]．運用時に資源を管理する必要のある内容については全て本方式が適用可能であるが，筆者らはこの中でイントラネットにおけるセキュリティの確保に着目して本方式の適用を試みた．本論文ではその結果について示す．

第2章では POLICYCOMPUTING について概説する．第 3 章ではイントラネットにおけるセキュリティポリシーの制御に関する運用コスト上の問題点を述べ，第4章でセキュリティ情報の一元管理の有効性を示す．第5章で POLICYCOMPUTING のセキュリティポリシーへの適用方法を，第6章でまとめを記述する．

2. POLICYCOMPUTING™

2.1. 目的

POLICYCOMPUTING の目的は，運用管理コストを削減し，少ない運用管理コストでもきめ細かいシステム管理(資源の効果的な活用，セキュリティの向上 など)ができるようにすることである．

このため，「マスターポリシー(管理者にとって設定が行いやすい表現形式のポリシー)によって情報システム内の全てのリソース(機器，サーバプロセス，ユーザなど)を一元的に管理」し，以下のような点で「各リソースの設定に必要な管理コスト」を削減していく．

- マスターポリシーから個別ポリシーを自動的に作成することにより，管理者は少数のマスターポリシーを入力だけで済む．
- マスターポリシーを管理者にとって理解しやすい表現形式で記述できるようにすることで，システムに精通していない管理者でも十分に管理が行えるようにする．

2.2. 実施内容

POLICYCOMPUTING でマスターポリシーに基づく一元管理を実現する仕組みを図1にしめす．

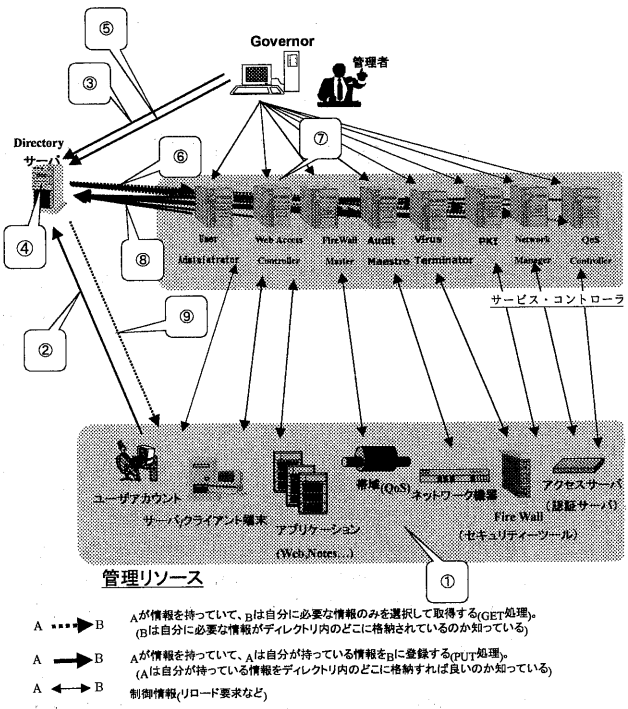

図1 POLICYCOMPUTING™の処理内容

図1の①～⑨までのそれぞれの処理内容は以下のとおりである．

①管理者がリソースを「POLICYCOMPUTING に加えるための最小限の設定」を行う．情報の組み込みにあ

たってはディレクトリサーバを利用することとし，自分のアドレス，ディレクトリサーバ(もしくはポリシーサーバ)のアドレスなどを設定する．

②各リソースがプロファイルを自動的に格納 する．

各々のリソースは起動時に自分が所持している動作状況をプロファイルとしてディレクトリサーバに格納する．また，起動中に動作状況に変化が生じた場合には変化分の情報をディレクトリサーバに格納する．プロファイルとは以下のようなデータである．

[プロファイルの例]

- 属性: ファイルの情報種別
  内容: ファイルXは人事情報
- 属性: 端末にログインしているユーザ
  内容: HostA にユーザYがログインしている

③管理者がプロファイルを格納する．

プロファイルの中には，管理リソースが自動的(機械的)に格納することができないプロファイルがある．例えば，ユーザ・アカウントなどである．自動的(機械的)に格納できないプロファイルは，管理者が手作業で格納する．

④知識データ が登録されている．

知識データとはシステムに依存しない運用ノウハウのような情報である．あらかじめディレクトリサーバに格納しておき，システム管理者がシステム運用中に変更する必要がないようにする．知識データとは以下のようなデータである．

[知識データの例]

- 属性:アプリケーションのポート番号
  内容: ポート番号N番はIP-Phoneである．
- 属性:サービスの品質
  内容: IP-Phoneは100kbpsの帯域があれば高品質である．

⑤管理者がマスターポリシーを格納する．

マスターポリシーも「属性」と「内容」によって構成されている．内容は「A = B」という形式に容易に変更できるような文章である．属性の値によってAおよびBに入る単語の種類を定義しておく．マスターポリシーとは以下のようなデータである．

[マスターポリシーの例]

- 属性: 文書セキュリティ
  内容: 人事情報は管理職のみ閲覧可とする
- 属性: サービスQoSポリシー
  内容: 管理職のコミュニケーション・ツールの通信は高品質にする

⑥マスターポリシー，プロファイル，知識データのうち必要な情報を取得する．

⑦取得した情報を利用して個別ポリシーを自動生成する．

個別ポリシーとはリソースの詳細な設定情報であり，以下のようなデータである．

[個別ポリシーの例]

- 属性: ファイル・アクセス
  内容: ユーザYはxxxにアクセス不可
- 属性: ネットワークQoS
  内容: 端末A-端末B間のポートN番の通信を100kbpsで帯域を確保する

⑧個別ポリシー(設定情報)をディレクトリサーバに格納する．

⑦で作成した個別ポリシーをディレクトリサーバの所定の場所に格納する．

⑨各リソースが個別ポリシー(設定情報)を取得する．

リソースは自分にとって，どの属性の個別ポリシーが必要であるかを知っている(あらかじめ定義されていて，あらかじめリソースに設定しておく)．ディレクトリサーバ内のどの属性の情報を参照するかは，リソース毎に異なる．

3.イントラネットにおけるセキュリティポリシー

3.1. セキュリティ管理におけるポリシー制御の導入

業務の効率化のために組織内に分散した各種情報を相互に参照することが必須になってきている．特にイントラネットのように基本的にクライアントサーバシステムで構成されたネットワークでは組織内の機密情報を始めとする重要な情報が各サーバ間に分散しているため情報の適切なアクセスの必要性が認識されている．

具体的なアクセス制御の例としては以下のようなものがあげられる．

- WWWやメールサーバ（社内，社外向け）へのアクセスの制御
- 各サブ組織が持っているサーバ類のアクセス制御（サブ組織内，あるいは他サブ組織間，組織外）
- 各自が持つファイルのアクセス制御，等々

このようなアクセス制御が必要なシステムではセキュリティの管理を厳格に行うことが急務である．

セキュリティ管理に関しては企業などそれぞれの組織が管理における考え方（セキュリティポリシー）を規定し，これに基づきシステムの管理を行うことが一般的となっている．

分散環境下でセキュリティポリシーを適用するには以下のような仕組みが必要であると考えられる．

- セキュリティポリシーを集中的に管理することが可能な仕組み
- 分散された環境にそのセキュリティポリシーを反映することが可能となる仕組み
- セキュリティ管理者が自分の決めたポリシーが反映されていることが把握できる仕組み

3.2. ポリシー制御導入時のコスト増大の要因

3．1で示した仕組みを導入する場合，最も問題となるのがコストの増大を如何に減少させるかということである．コスト増大の要因を以下に示す．

- リソースが分散されている場合，これらのリソースが格納されているアドレスを全て知っている必要がある．
- 各サブ組織ごとのシステム管理者が別々にセキュリティ制御に関するパラメータの設定を行わなければならない．
- また，管理者がリソースの相違を知っている必要がある．
- 新しいリソースを追加するたびに各管理者が新規にセキュリティパラメータを理解しシステムに追加する必要がある．

例えば，Ｗｅｂサーバと FireWall で同一アクセス制御を行いたい時は，セキュリティパラメータを Web サーバ用からFireWall用に変換する必要がある等が考えられる．

4.セキュリティ情報の一元管理の有効性

3章で示した問題を解決するためにはこれらセキュリティ情報を一元管理することが有効であると考えられる．

以下にセキュリティポリシー及びユーザ情報の一元管理による有効性をしめす．

4.1. セキュリティポリシーの一元管理の有効性

ネットワークセキュリティの対象となる脅威には盗聴，改竄，なりすまし，不正アクセス等があげられるが，対策としては暗号化等のように情報自体に対策を施すものと，ネットワークを通してリソースに不正にアクセスされることを防ぐための対策を施すものとある．今回，ネットワークの管理の観点からアクセス制御に着目した場合の問題は以下のとおりである．

例えばWWWサーバを例に取った場合，ユーザの異動やセキュリティポリシーの変更，情報の追加などに対してセキュリティ管理者からWWWサーバ管理者にたいし変更の依頼を行うが，依頼者と作業者が異なるため確認に稼働がとられたり，相互の行き違いからトラブルが発生することが少なくない．

これらは分散制御の中でサーバが各所に設置されている場合にはなお顕著となる．

従って，これらの問題を解決するには，　i)分散されたサーバ，クライアント等のリソースに対して，集中的な管理を可能としたり，　ii)管理者がリソースの相違，物理的・論理的な位置を意識することなくパラメータを設定することを可能とする，　iii)新しいリソースが追加されても管理上の変更を容易とすることが望ましい．

このためにはサーバなどのリソースの情報を一元的に管理する仕組みを作ることが有効となる．

4.2. ユーザ情報の一元管理の有効性

ネットワークシステムの中では，実際のユーザが持つ氏名が例えばディレクトリサーバ，ドメインのアカウント，ＩＰアドレス等で異なって（違った形式で）管理されている．このため，システム運用者，セキュリティ管理者，そして時には本人がこれらの情報を間違いなく管理するにはかなりの労力を要する．また，組織では人事異動などが頻繁に発生するがこの場合の管理稼働は極めて莫大なものとなる．

従って，これらの問題を解決するには管理元が異なる場合でも同じユーザである限りは一元的に管理する仕組みを作ることが有効となる．

5.POLICYCOMPUTING のセキュリティポリシーへの適用

POLICYCOMPUTING の特長はポリシーを一元化すること，及び管理の対象となる情報を一元管理することである．

以上から 4 章で示したとおりセキュリティポリシーに POLICYCOMPUTING を適用することは極めて有効であると考えられる．

この場合，ｉ）ポリシーをどのように一元化するか，

ｉｉ）管理の対象となるセキュリティポリシーとユーザ情報をどのように一元管理するか，ということが課題となる．

以下にその方法について検討した結果を示す．

5.1. ポリシーの一元化

今回の実装ではセキュリティポリシーを対象として検討してきたが，本来の POLICYCOMPUTING では，セキュリティのみでなくネットワーク運用時のポリシー，あるいはネットワーク利用時のサービス品質（ＱｏＳ）に関するポリシー等も含め統一した考え方で管理したい．

従来ＱｏＳについても POLICYCOMPUTING で対応することを検討してきており[4]，ここでの検討結果を踏まえセキュリティポリシーに関してもディレクトリサーバを利用し他のポリシーと統一的に扱い，２章に示したメカニズムで管理していくこととする．

（図 2）

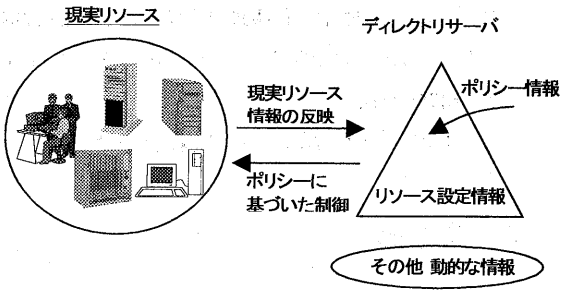

図2　セキュリティポリシー情報の管理

5.2. 情報の一元管理

３．２節で示したポリシー制御導入時のコスト増大の回避するには，分散されたリソースの情報を把握していること，あるいは新しいリソースが追加されても問題無く管理情報を更新できること等を柔軟に且つ拡張性をもって行うことが必要になる．

一方，組織構造は一般的に階層化されており，各階層毎に管理者がいてその管理者は自分の所掌する範囲を管理しておけば全体的に管理ができるといった仕組みになっている．

今回は，このアナロジーによりポリシー情報についても階層化された構造により一元管理することを提案する．

ポリシーコンピューティングによりポリシーを集中管理する概念図を図３（次頁）に示す．

5.3. システムの特徴

５．１，５．２節に示した仕組みを実現することにより，本システムでは以下のような特徴を持つ．

例としてＷｅｂサーバの管理を取り上げた場合を示す．

（1） POLICYCOMPUTING によりセキュリティーポリシーは一元化してディレクトリサーバに格納される．これにより情報セキュリティポリシーについてはこの情報だけを管理すれば良い．また，これを行うことが出来るのはセキュリティ管理者のみである．

（2）Ｗｅｂサーバ管理者は情報と情報種別を関連付けておき，後は情報の管理を行えば良い．

これによって管理者の所掌範囲が明確となり，相互の連携不足から生じるミスを削減することが出来るとともに，責任範囲も明確となる．

今回実現したシステムのイメージを図４に示す．

6．まとめ

筆者らは，大規模なクライアントサーバシステムは運用管理コストが大きいという問題点があることから「各リソースの設定に必要な管理コスト」を削減するという点に着目し，マスターポリシーによる一元管理システム POLICYCOMPUTING を提案してきた．本論文では POLICYCOMPUTING を適用する事例としてセキュリティーポリシーの制御をとりあげ，適用にあたっての課題とその対策を示した．

今後は，セキュリティポリシーの制御に関する POLICYCOMPUTING の有効性を検証する予定である．

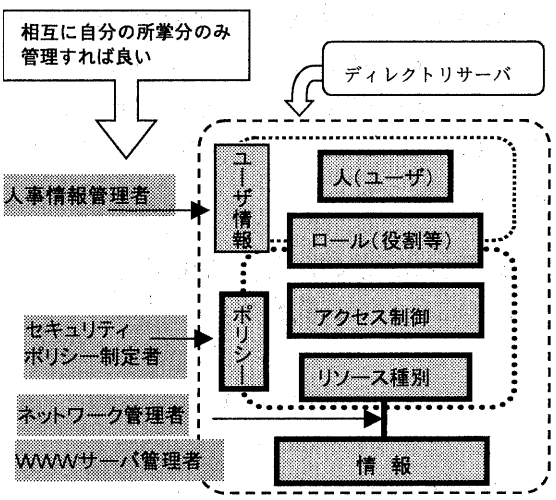

図3 階層的ポリシー管理

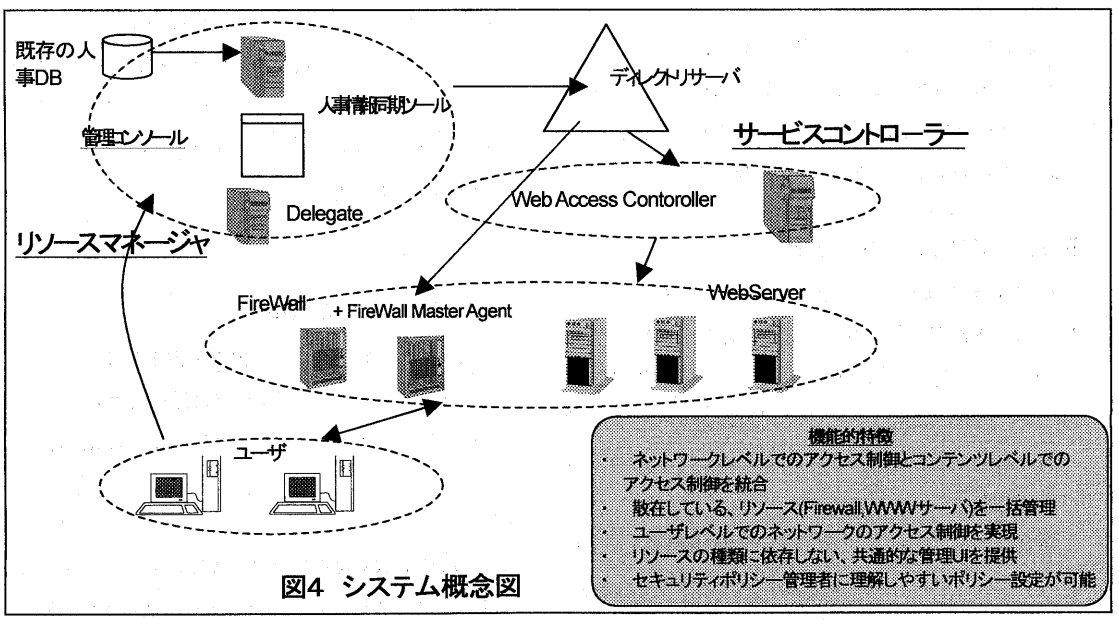

図4 システム概念図

[参考文献]

[1] 大鐘,TCP/IP と OSI ネットワーク管理,SRC,Apr.1993

[2] K. Dec, etc, Management Solutions for the Distributed Computing Revolution, Garter Group Strategic Analysis Report, Jul.1996

[3] 山口,インターネットの今後,情報処理学会連続セミナー'98 第３回,Oct.1998

[4] 田中，菅野，他：情報ネットワークシステムのポリシー制御 POLICYCOMPUTING™ に関する一検討，情処研報 Vol99,No.18,pp.121-126 (Feb.1999)