2009-IOT-4 (31)
2009 / 3 / 5

持ち込み PC を対象としたネットワーク利用許可権限の委譲を可能にするアクセス制御メカニズムの実現

馬渕 充啓† 小沢 健史† 高田 真吾† 豊岡 拓† 松井 慧悟†
佐藤 聡† 新城 靖† 加藤 和彦†

† 筑波大学大学院システム情報工学研究科 〒 305-8573 茨城県つくば市天王台 1-1-1
E-mail: †mmabu@osss.cd.tsukuba.ac.jp, †{ozawa, takada, hiraku, pachi}@softlab.cs.tsukuba.ac.jp,
†akira@cc.tsukuba.ac.jp, †{yas, kato}@cs.tsukuba.ac.jp

あらまし 組織外部の利用者が持ち込んだ PC を用いて組織のネットワークを利用することを許可する場合，利用者の利便性を向上させることと管理者による管理コストを低減することを同時に実現することは非常に難しい．従来の手法では，管理者は，利用者を特定するために利用者登録を行う必要がある．これらの管理作業は管理者しか行うことができないため，管理者が不在の場合に利用許可を得ることができないという問題がある．我々は，この問題を解決するために，利用者に対して利用許可を発行するための管理権限を，管理者から組織内部の利用者へ委譲することを可能にするアクセス制御メカニズム（CaNector）を提案する．提案手法では，管理権限委譲を実現するためにケーパビリティに基づくアクセス制御を用いている．CaNector に対する全ての操作は Web ブラウザから行うことが可能であり，利用者は特別なソフトウェアのインストール等を行うことなく持ち込み PC を用いてネットワークを利用することが可能である．この論文では，CaNector を筑波大学システム情報工学研究科コンピュータサイエンス専攻所属のソフトウェア研究室においてテスト運用した結果について報告する．

キーワード ネットワーク運用技術，アクセス制御，権限委譲，ケーパビリティ

Implementation of an Access Control Mechanism that Enables Delegating Rights of Network Usage Permissions for Guest PCs

Mitsuhiro MABUCHI†, Tsuyoshi OZAWA†, Shingo TAKADA†, Hiraku TOYOOKA†, Keigo MATSUI†, Akira SATO†, Yasushi SHINJO†, and Kazuhiko KATO†

† Graduate School of Systems and Information Engineering, University of Tsukuba Tennoudai 1–1–1, Tsukuba, Ibaraki, 305-8573 Japan
E-mail: †mmabu@osss.cd.tsukuba.ac.jp, †{ozawa, takada, hiraku, pachi}@softlab.cs.tsukuba.ac.jp,
†akira@cc.tsukuba.ac.jp, †{yas, kato}@cs.tsukuba.ac.jp

Abstract When guests access their PCs to an organization's network, it is a challenging task for network administrators to achieve user-friendliness while reducing administrators' efforts. In conventional systems, an administrator needs to perform some administrative operations including registration of a name, address, and email address to identify guest users. Since these administrative operations are performed only by administrators, there is a problem that guests cannot receive network usage permissions if administrators are not present. To solve this problem, we propose CaNector, an access control mechanism that enables delegating rights of network usage permissions for guest PCs. In this mechanism, we use a capability-based access control model to implement delegation of rights. CaNector enables host and guest users to handle capabilities with Web browsers and host and guest users don't need to install any special softwares to their PCs . In this paper, we report experimental results in a laboratory in Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba.

Key words Network Operation Technology, Access Control, Delegation of Rights, Capabilities

1. まえがき

企業や大学等の大規模な組織では，多くの場合，利用者を認証することによりネットワークに対するアクセス制御を行っている．このとき，利用者は，自分の権限の範囲内でのみネットワークを利用することができる．たとえば，HTTP プロトコルを用いた通信でのみネットワークを利用することができる．

このような従来の手法では，組織外部の利用者が持ち込み PC を用いて組織のネットワークを利用することを許可する場合，管理者は利用者を特定するために利用者登録を行う必要がある．このような場合，利用者の利便性を向上させることと管理者による管理コストを低減することを同時に実現することは非常に難しい．たとえば，大学では，共同研究者や学会開催等により組織外部の利用者が多数訪れ，持ち込んだ PC を用いて一時的にネットワークを利用することがある．その度に，外部利用者を登録するということは，管理者にとって大きな負担となる．また，これらの管理作業は管理者しか行うことができないため，管理者が不在である場合に利用許可を得ることができないという問題がある．

本論文では，上述した問題を解決するために，利用許可を発行する管理権限を管理者から組織内部の利用者に委譲可能にするアクセス制御メカニズム（CaNector）の実現について述べる．CaNector の特徴は，ネットワークのアクセス制御において利用者認証ではなく，ケーパビリティを用いていることにある [1] [6] [4] [7] [10] [13]．CaNector を用いることで，DNS（Domain Name Service）[8] [9] のように管理権限を分散管理することが可能になる．

組織外部の利用者に対しては，組織のネットワーク利用を許可するとしても組織内部のサーバにアクセスできないようにするなどの制限を加える必要がある．そのため，CaNector では，ケーパビリティの属性として宛先 IP アドレス，宛先ポート番号，有効期限，そして，使用回数等を設定可能にする．また，悪質な利用に関しては，事後に利用者を特定できるようにする必要がある．そこで，CaNector では，ケーパビリティの利用ログをとる機能も提供する．

実験では，CaNector を使用したネットワーク利用許可にかかる処理時間を計測した．また，CaNector は，筑波大学システム情報工学研究科コンピュータサイエンス専攻ソフトウェア研究室で２００８年１２月から約３ヶ月間に渡りテスト運用を行っている．それらの結果を利用して，CaNector の評価を行った．

本論文は以下の章で構成される．まず，2. では，CaNector の設計について述べる．3. では，CaNector の実装について述べる．4. では，CaNector を用いた実験とそのテスト運用について述べる．5. では，本論文のまとめと今後の課題について述べる．

2. CaNector の設計

2.1 想定する環境

CaNector の想定する環境では，**管理者**がおり，組織のネットワークを管理している．また，その組織のネットワークに対して元々アクセス権限を保持することが可能な組織内部の利用者を**内部利用者**と呼ぶ．そして，そのネットワークに対して元々アクセスする権限を保持することができない組織外部の利用者を**外部利用者**と呼ぶ．内部利用者と外部利用者の両方を指す場合は，単に**利用者**と呼ぶ．このような組織に複数の外部利用者が不定期に訪れ，一時的にネットワークを利用することが頻繁にある．たとえば，研究室の共同研究者や学会開催に伴う参加者の来訪により，その時に限ってネットワークを利用することである．

2.2 想定環境における従来手法の問題

2.1 で述べたような環境において，従来手法を用いた場合について考察する．

利用者認証と利用者記録を行うゲートウェイシステムとして佐賀大学で開発された Opengate がある [12]．Opengate では，外部利用者に対してネットワーク利用許可を出す場合，管理者が一時的なアカウントを用意しそれを外部利用者に渡す運用を行っている．Opengate を用いた場合，管理者がその場にいないとアカウントを受け取ることができないという問題がある．ジョージア工科大学で開発された LAWN では，教員はネットワークを利用するための一時的なアカウントを発行する権限を持っている [5]．しかし，LAWN を用いた場合も，教員がその場にいないとアカウントを発行できないという Opengate と同じ問題がある．また，Opengate や LAWN で発行されたアカウントは，発行された瞬間から有効になるため事前に配布しておくという運用形態をとることはできない．

2.3 要求要件

前節で述べた問題を解決し目的を達成するために，以下の要求を満たすアクセス制御メカニズムを設計する．

（1） ネットワーク利用許可の権限を管理者から，内部利用者に対して委譲可能にする．

（2） ネットワークに対するアクセスを制限した利用を許可することを可能にする．

（3） 利用許可を事前に行い，その許可を事前に利用者に配布可能にする．

（4） 複数の利用者に対して，一括で利用を許可することを可能にする．

（5） 利用者が特別な設定やソフトウェアのインストールなしに利用可能にする．

（6） 利用者を事後に特定するために，利用ログを取得可能にする．

要求 1 を満たすために，CaNector では，利用者認証に基づくアクセス制御ではなくケーパビリティに基づくアクセス制御を用いる．ケーパビリティとは，アクセス対象であるオブジェクトの識別子とそのオブジェクトに対して可能な操作から構成されるものである [1] [6] [4] [7] [10] [13]．CaNector で扱うケーパビリティには，ケーパビリティ管理権限（ケーパビリティの作成，編集，そして，削除等の管理作業を行うことができる権限）とネットワーク利用権限（ネットワークを利用することができる権限）の２種類がある．本論文では，以降ケーパビリ

ティ管理権限のことを管理権限と呼ぶ．管理権限を内部利用者に分散することにより，管理者，あるいは，教員が不在の場合にでも対応可能にする．

要求２と要求３を満たすために，CaNector では，ケーパビリティに対して属性として宛先 IP アドレス，宛先ポート番号，そして，有効期限を設定可能にする．宛先 IP アドレスを制限することで，組織内部のサーバ等にアクセスできないようにすることが可能になる．また，宛先ポート番号を制限することで，使用可能な通信プロトコルを制限することが可能になる．有効期限は，開始時刻と終了時刻の２つの要素から構成し，未来に有効になる権限を作成することを可能にする．これにより，事前に作成したケーパビリティを配布しておくことで，作業可能な時間帯に作業を行うことを可能にする．

要求４を満たすために，CaNector では，同じ権限のケーパビリティを一度に複数作成することを可能にする．これにより，複数の利用者に対して，同じ作業を繰り返すことなく利用を許可することが可能にする．

要求５を満たすために，CaNector では，ケーパビリティの利用や管理を行うインタフェースとして Web ブラウザを用いる．Web ブラウザは，ネットワークを利用する利用者にとって使い慣れた物であり，多くの PC に最初からインストールされているため特別なソフトウェアをインストールすることなく使用することができる．また，ケーパビリティの利用方法として，QR コードも利用可能にする．これは，スマートフォン等の入力インタフェースが貧弱なモバイル機器でケーパビリティを入力する手間を削減するためである．

要求６を満たすために，CaNector では，ケーパビリティの利用に関する記録を IP アドレスと MAC アドレス，そして，利用された日時とともにデータベースに記録する．ケーパビリティの利用とは，ケーパビリティを用いたネットワーク利用，ケーパビリティの作成，編集，そして，削除等である．また，CaNector では，データベースに保存したログを管理画面で閲覧可能にする．

3. CaNector の実現

3.1 CaNector の構成

図 1 に CaNector の構成を示す．CaNector は，ゲートウェイ・スイッチとサービスサーバで構成される．ゲートウェイ・スイッチは，パケットフィルタリング機能を用いて，あるパケットがそのスイッチを通ることができるかどうかを検査することで通信路の開放閉鎖を行う．サービスサーバは，ケーパビリティ管理サービス，DHCP サービス，そして，DNS サービスを LAN 内のクライアントに提供する．LAN 内のクライアントがサービスサーバにアクセス可能となるようにゲートウェイ・スイッチを設定する．また，デフォルトでは，LAN の外にはアクセスできないようにゲートウェイ・スイッチを設定する．サービスサーバとクライアント間の通信が盗聴されケーパビリティが漏洩することを防ぐために，HTTP については HTTPS を用いて通信を暗号化している．DNS と DHCP に関しては，暗号化していない．

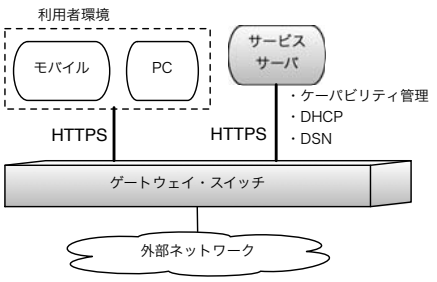

図 1　CaNector の構成．
Fig. 1　The architecture of our system.

CaNector の特徴として，サービスサーバとゲートウェイ・スイッチが完全に分離していることが挙げられる．そのため，現在 CaNector では，スイッチ専用のアプライアンス（Extream Network 社製の eXtreme Summit X150）を用いて通信路の開放閉鎖を行っているが，他の機種や iptables [2] 等のパケットフィルタリングを行う汎用 OS で動作するプログラムに変更することも可能である．

3.2 CaNector におけるケーパビリティ

CaNector で扱うケーパビリティの本体は，以下の要素から構成される．

- 識別子：このケーパビリティを識別するために用いる．
- 宛先 IP アドレス：通信可能な宛先 IP アドレスを表す．
- 宛先ポート番号：通信可能な宛先ポート番号を表す．
- 有効期限：ケーパビリティの使用可能な期限を表す．開始時刻と終了時刻の２要素から構成される．
- 使用回数：ケーパビリティの使用可能な回数を表す．
- 上位のケーパビリティの識別子：このケーパビリティを作成する際に用いられた元のケーパビリティを識別するために用いる．

ケーパビリティは，乱数により作成した識別子によって識別される．乱数を用いる理由は，識別子の偽造を防ぐためである [3] [4] [7]．利用者がケーパビリティを利用する際は，識別子を CaNector に与える．識別子を受け取った CaNector は，それを用いてケーパビリティ本体をデータベース内から探す．

あるケーパビリティを元にして作成されるケーパビリティは，元のケーパビリティよりも権限が弱くなる．このように，元のものよりも権限が弱くなったケーパビリティのことを，本論文では，**弱いケーパビリティ**と呼ぶ．弱いケーパビリティを作成すると，ケーパビリティは DNS の管理構造の様に階層構造を形成する [8] [9]．図 2 にケーパビリティの階層構造の例を示す．

弱いケーパビリティは，管理者，および，管理権限ありのケーパビリティを保持する内部利用者の両方によって作成される．管理者によって作成されたケーパビリティは，階層の最も上に位置する（図 2 の Capability A）．本論文では，このようなケーパビリティを**ルート・ケーパビリティ**と呼ぶ．管理権限ありのケーパビリティを保持する内部利用者によって作成される全てのケーパビリティは，元のケーパビリティよりも弱くなる．たとえば，Capability B，C，そして，E は，それぞれ別の

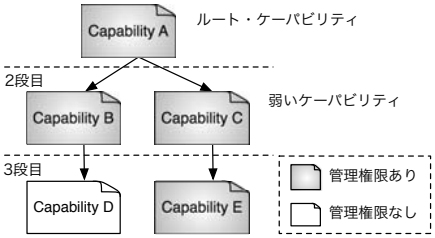

図 2　ケーパビリティの階層構造.
Fig. 2　The hierarchy of capabilities.

ケーパビリティを元に作成されているためそれらよりも権限が弱くなっている．このように，ケーパビリティに階層を取り入れることで，現実社会に存在する管理者，教員，学生，そして，外部利用者のような権限の階層構造と一致したアクセス権限を作成し各利用者に配布することができる．たとえば，大学等での研究室内部ネットワークについて考える．まず，管理者が教員に対して無制限の Capability A を作成し渡す．次に，教員がこの Capability A に有効期限を付加した Capability B，C，そして，E を作成し学生達に渡す．さらに，Capability B を保持した学生がそれに使用回数制限を付加し管理権限をなしにした Capability D を作成し，共同研究者である外部利用者に渡す．また，Capability C を保持した学生がさらに有効期限を狭めた Capability E を作成し別の学生に渡す．このようにして，ケーパビリティ作成の権限を制限して下位層の利用者に委譲することができるため，管理作業を管理者や他の内部利用者間で分担することが可能になる．管理権限なしのケーパビリティは，ケーパビリティの作成を行うことはできないため，階層の葉に位置する．

3.3　ケーパビリティの利用方法

CaNector では，ケーパビリティの利用方法として以下の 2 種類を用意する．

- 利用者が，Web ブラウザでケーパビリティの識別子をポータルページに対して手で入力.
- 利用者が，ケーパビリティの識別子を含む URL にアクセス.

Web ブラウザを用いる場合，利用者は，CaNector のケーパビリティ入力ページにケーパビリティの識別子を入力し ”connect” ボタンをクリックする．すると，CaNector に識別が送信され，CaNector は受け取った識別子を用いてケーパビリティ本体を取り出しその検証を行う．取り出されたケーパビリティに上位のケーパビリティがある場合，再帰的にルート・ケーパビリティにまでさかのぼって全てのケーパビリティを検証する．全てのケーパビリティが存在し，かつ，権限が正しい場合，CaNector は，スイッチにその利用者の持ち込み PC の許可エントリを追加する．その後，ブラウザには，操作が成功したことと使用したケーパビリティの権限情報が表示される．

URL にアクセスを行う場合，URL に含まれた識別子がパラメータとして CaNector に送信される．CaNector は，その識別子を用いてケーパビリティ本体を取りだしその検証を行う．後の CaNector の動作は，上述した場合と同じである．また，この URL をエンコードした QR コードを利用することで，URL を手で入力する手間を省くことが可能になる．

スイッチに追加された許可エントリは，cron を用いて許可された期限が過ぎたら削除される．

3.4　ケーパビリティの管理

ここでは，管理者，および，管理権限ありのケーパビリティを保持する内部利用者によるケーパビリティの管理方法について述べる．ケーパビリティの管理操作として以下のものがある．

- **作成**：ケーパビリティを作成する．管理者に作成される場合，ルート・ケーパビリティが作成される．管理権限ありのケーパビリティを保持する内部利用者に作成される場合，元のケーパビリティの下に位置するケーパビリティが作成される．ケーパビリティが作成される時，同時にこのケーパビリティの識別子を含んだ URL をエンコードした QR コードも作成される．また，同じ権限のケーパビリティであれば，一度の操作で指定個数作成できる．
- **編集**：ケーパビリティの権限を編集する．ただし，元のケーパビリティの権限を越えた編集はできない．たとえば，元のケーパビリティの使用回数が１０回であるとき，その下に位置するケーパビリティの使用回数を１０回より大きくすることはできない.
- **削除**：ケーパビリティを削除する．また，そのケーパビリティよりも下位層に位置する全てのケーパビリティが削除される.

上述した管理操作について，CaNector のスクリーンショットを用いて詳しく説明する．図 3 には，管理者がケーパビリティを管理する際のスクリーンショットを示す．管理者の場合，全てのルート・ケーパビリティとそれらから派生したケーパビリティが表示される．図 4 には，管理権限ありのケーパビリティを保持する内部利用者がケーパビリティを管理する際のスクリーンショットを示す．この管理画面にアクセスするためには，管理権限ありのケーパビリティの識別子を入力する必要がある．管理権限ありのケーパビリティを保持する内部利用者の場合，入力された識別子を用いて取得したケーパビリティと，それよりも下位層に位置するケーパビリティ全てが表示される．図 4 の上部に表示されているケーパビリティが，入力された識別子を用いて取得したケーパビリティである．また，入力された識別子を用いて取得されたケーパビリティに対しては操作（編集，削除等）を行うことができないようになっている．

各ページの下部にある ”ケーパビリティの作成” リンクをクリックすると，ケーパビリティの権限を入力するページ（図 5）が表示される．まず，管理権限ありのケーパビリティを作成する場合は，チェックボックスにチェックをする．チェックがない場合，管理権限なしのケーパビリティが作成される．権限を入力後，ページの下部にある ”作成” リンクをクリックすることでケーパビリティの作成が完了する．入力された権限と元のケーパビリティの権限を比較し，権限が弱くなっていれば作成を許可する．しかし，強くなっている場合は，作成を拒否し再度権限の入力を求める．また，同じ権限のケーパビリティを一

図 3　管理者用管理ページのスクリーンショット.
Fig. 3　The screenshot of the manager's page.

図 4　利用者用管理ページのスクリーンショット.
Fig. 4　The screenshot of a host user's page.

図 5　ケーパビリティ作成ページのスクリーンショット.
Fig. 5　The screenshot of the page for creating capabilities.

括で作成したい場合，作成個数を指定のフィールドに入力する．

ケーパビリティに付加された権限を編集する場合，ケーパビリティの右側に表示されている ”編集” リンクをクリックする．すると，権限を入力するページ（図 5 とほぼ同じ）が表示されるので，そこに新しい権限を入力し編集の ”確定” リンクをクリックすることで権限の編集が終了する．ここでも，入力された権限が元のケーパビリティの権限より強くなっていないかどうかを検査する．また，削除する場合，各ケーパビリティの右側にある ”削除” リンクをクリックすると削除確認が表示され，OK を選択すると削除される．

3.5　ケーパビリティの利用ログ

CaNector では，ケーパビリティの利用ログをデータベースに保存する．保存されたログは，ケーパビリティ管理画面で閲覧可能である．

図 6　ログ表示ページのスクリーンショット.
Fig. 6　The screenshot of the page for displaying logs.

ログの表示画面を図 6 に示す．ログは，日時を用いてソートされた順で表示される．ケーパビリティに対する操作（create, destroy，edit）の場合，ログには操作対象のケーパビリティが表示される．ケーパビリティを用いてネットワーク接続の操作（connect，disconnect）を行う場合，ログにはその PC の IP アドレスと MAC アドレスが表示される．

4.　実験とテスト運用

4.1　マイクロベンチマーク

利用者がケーパビリティの識別子を CaNector に入力し，スイッチの操作が完了するまでの時間が長い場合，利用者にとっては大きな負担となる．3.2 で述べたようにケーパビリティには階層があり，ケーパビリティの検証やその権限の検証は再帰的に行われる．そこで，ルート・ケーパビリティから 4 段目のケーパビリティまでの 5 種類のケーパビリティについて，以下の 4 つの場合に分けて処理時間の計測を行った．

- **制限なし**：全てのケーパビリティに制限がない．
- **有効期限付き**：全てのケーパビリティに有効期限がある．
- **使用回数制限付き**：全てのケーパビリティに使用回数制限がある．
- **有効期限／使用回数制限付き**：全てのケーパビリティに有効期限と使用回数制限の両方がある．

この実験で用いたクライアント・マシンと Web サーバの環境を表 1 に示す．今回使用したハードウェア・スイッチは，eXxtreme Summit 150-24t である．また，要求を行うプログラムとして wget を用いた．

計測した結果を図 7 に示す．横軸はケーパビリティの階層を表しており，縦軸は処理時間を表している．処理時間の単位は秒で，これらの処理時間は１０回同じ実験を繰り返したものの平均時間である．また，Telnet でスイッチの操作を行う処理時間は，約 1.2 秒だった．

CaNector の処理時間は 2 秒前後であり，ケーパビリティの階層が深くなるにつれ処理時間は大きくなる傾向があることがわかる．処理時間のほとんどは，Telnet でのスイッチ操作と HTTPS を用いた通信である．そのため，CaNector 自体の処理時間はそれほど大きくはない．

表 1　実験環境.
Table 1　Environments for the experiment..

	クライアント	サービスサーバ
CPU	Pentium Core 2 Duo 2.2GHz	Pentium D 2.8GHz
メモリ	2.0GB	512MB
OS	Mac Ubuntu 8.04 (Linux Kernel 2.6.24)	Cent OS 5.2 (Linux Kernel - Xen 2.6.18)
LAN	Gigabit Ethernet	Gigabit Ethernet

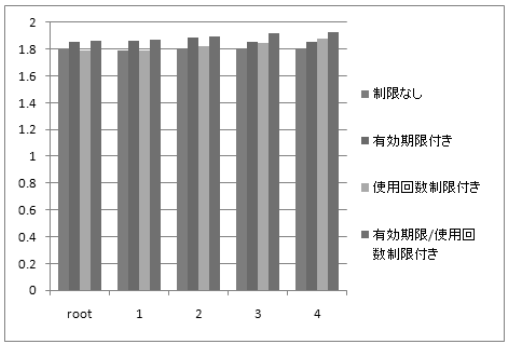

図 7　CaNector を用いたアクセス制御にかかる時間.
Fig. 7　Processing times of access control using CaNectors.

表 2　利用者の評価.
Table 2　The evalution by users.

評価項目	平均点
ネットワークの利用し易さ	3.4
ケーパビリティ管理機能の使い易さ	2.7

4.2　テスト運用

CaNector は，現在，筑波大学システム情報工学研究科コンピュータサイエンス専攻のソフトウェア研究室でテスト運用を行っている．主に無線 LAN に持ち込み PC やスマートフォン等のモバイル機器を接続して利用している．同時使用台数は 10 台程度で，２００８年１２月から約３ヶ月間にわたって運用している．

4.3　調 査 結 果

テスト運用期間中に CaNector を使用した 7 名の利用者に対してアンケート調査を行い，以下の 2 項目に関して 4 段階（4 が最高で，1 が最低）で評価してもらった．

（1） ネットワークの利用し易さ．

（2） ケーパビリティ管理機能の使い易さ．

各選択項目の点数の平均を表 2 に示す．評価項目 1 に関しては，一定の評価を得ることができた．しかし，評価項目 2 に関しては，改善の余地があることが分かった．コメントとしては，ケーパビリティ作成の際に，JavaScript 等を用いて権限を簡単に入力できるようなインタフェースを用意するとよい等があった．

5. ま と め

本論文では，ネットワーク利用許可発行権限を管理者から内部利用者に委譲可能なアクセス制御メカニズム（CaNector）について述べた．CaNector を用いることで，管理権限を分散管理することが可能になるため，管理者がいなくても他の利用者に対して利用を許可することが可能になる．

現在，CaNector は２００８年１２月から約３ヶ月間のテスト運用を行っているが，実用性を示すためにはより長く，そして，より多くの利用者に利用してもらい意見を聴く必要がある．今後の課題としては，利用可能場所を拡大し多くの人が利用可能な環境を構築することと，利用者の意見を取り入れることによるシステムの改善である．

文　献

[1] M. Accetta, R. Baron, W. Bolosky, D. Rashid, A. Tevanian, and M. Young, Mach: A New Kernel Foundation for UNIX Development, Proceeding of USENIX Summer Conference, pp.93-112, 1986.

[2] O. Andreasson, Iptables Tutorial 1.2.0, 2005.

[3] M. Anderson, R. D. Pose, and C. S. Wallace, A Password-Capability System, The Computer Journal, 29(1):1-8, 1986.

[4] R. Geambasu, M. Balazinska, S. D. Gribble, and H. M. Levy, HomeViews: Peer-to-Peer Middleware for Personal Data Sharing Applications, Proceedings of the 2007 ACM SIGMOD international conference on Management of data, pp.235-246, 2007.

[5] Georgia Institute of Technology, LAWN: Local Area Walkup/Wireless Network, http://www.lawn.gatech.edu/index.html, 2008.

[6] H. M. Levy, Capability-Based Computer Systems, Digital Press, 1984.

[7] M. Mabuchi, Y. Shinjo, A. Sato, and K. Kato, An Access Control Model for Web-Services that Supports Delegation and Creation of Authority, Proceeding of Seventh International Conference on Networking, pp.213-222, 2008.

[8] P. Mockapetris, *Domain Names - Concepts and Facilities*, RFC1034(standard), Nov. 1987.

[9] P. Mockapetris, *Domain Names - Implementation and Specification*, RFC1035(standard), Nov. 1987.

[10] S. J. Mullender, G. Rossum, A. S. Tenenbaum, R. van Renesse, and H. van Staveren, Amoeba: A Distributed Operating System for the 1990s, IEEE Computer, Vol.23, pp.44-53, 1990.

[11] A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts, Wiley, 2008.

[12] 渡辺義明，渡辺建次，江藤博文，只木進一，利用と管理が容易で適用範囲の広い利用者認証ゲートウェイシステムの開発，情報処理学会論文誌，vol.42，No.12，pp.2802-2809，2001.

[13] W. Wulf, E. Cohen, W. Corwin, A. Jones, R. Levin, C. Pierson, and F. Pollack, HYDRA: The Kernel of a Multiprocessor Operating System, Communication of the ACM, Vol.17, No.6, pp.337-345, 1974.