2007－CSEC－37 (7)
2007／5／25

ユーザによる設定を可能とする Proxy型ネットワークアクセス制御方式の提案

関口 聖美† 黒羽 秀一† 齋藤 孝道‡
†明治大学大学院 ‡明治大学
〒214-8571 神奈川県川崎市多摩区東三田 1-1-1
{skgchi, kuroba, saito}@cs.meiji.ac.jp

あらまし 複数のユーザが利用するWebアプリケーションで，ユーザごとのディレクトリやファイルを直接的に利用するシステムでは，ユーザが直接，アクセス制御の設定を行いたいという要求がある．そこで，それらファイル等を用いたコンテンツ作成や運用の際，認証されたユーザがアクセス制御の設定を柔軟に行うことを可能とするProxy型のアクセス制御方式の提案と実装を行う．

キーワード アクセス制御 権限委譲 コンテンツ

Proposal of Network Access Control Scheme based on Proxy : User can Set an Access Control

Kiyomi SEKIGUCHI† Shuichi KUROBA† Takamichi SAITO‡
†Graduate School of Meiji University ‡Meiji University
1-1-1, Higashimita, Tama-ku, Kawasaki-shi, Kanagawa 214-8571, Japan
{skgchi, kuroba, saito}@cs.meiji.ac.jp

Abstract In Web Application that two or more user use, there are some systems that the user access their directories and files. In those systems, there are some requests that the user want to set an access control by themself. So, we propose and create an Access Control Scheme based on Proxy. The user who was certified can set an access control by using this scheme when the user create and use with their directories and files.

Keyword Access Control, Delegation of Authority, Contents

1 はじめに

XOOPS[1]やWiki[2]のように，管理者ではない一般ユーザが，Webサーバ内のファイルやディレクトリを編集して，コンテンツを作成するようなWebアプリケーションが利用されつつある．このようなWebアプリケーションにおいて，Webサーバ内に格納されているファイルやディレクトリにアクセス制御を設定する場合，管理者が一元的にアクセス制御を設定する．例えば，XOOPSのアクセス制御を設定できるユーザは，管理者権限を所有しているユーザだけであり，管理者権限を所有していない一般ユーザは，Webサーバ内のファイルやディレクトリに対してアクセス制御を設定することはできない．しかし，管理者が一元的にアクセス制御を設定すると，一般ユーザが希望するアクセス制御ポリシーの設定は，現実的ではない．また，この方式を利用する場合，ユーザ数やWebサーバ内のファイルやディレクトリ数の増加，およびアクセス制御ポリシーの肥大化に伴い，管理者の負担は増大する．

そこで，Webサーバ内に格納されているファイルやディレクトリに対して，管理者ではない一般ユーザによるアクセス制御の設定を，インターネット越しに可能とし，さらに，ユーザ間

でアクセス制御が設定可能な権限の委譲を可能とする．本論文では，上述の機能を持った，プロキシ型のアクセス制御方式の提案と実装を示す．

2 ファイルシステムにおけるアクセス制御の実現方式

管理者権限を持たないユーザが，自分の所有するファイルやディレクトリのアクセス制御の設定を実現する方式として，Linuxではkernel2.6[3]から実装されたACL (Access Control List) 機能を利用する方法がある．この機能は，ユーザがどのグループに所属しているかに関係なく，ファイルやディレクトリの所有者が，任意のユーザに任意のアクセス権限を与えることができるものである．この機能を用いて，ファイルやディレクトリの所有者であるユーザが，任意のユーザに，アクセス権限ではなく，アクセス制御を設定できる権限を与えたい場合，ACL機能ではこれを実現することはできない．例えば，ユーザAliceがユーザBobに，Aliceが所有するファイルに対するアクセス権限を与える (図1中の①)．そして，Aliceは，Bobが第三者であるユーザCarolにアクセス権限を与えることを許可する場合 (図1中の②)，Bobがそのファイルのアクセス権限をCarolに与えることはできない．よって，AliceがCarolにアクセス権限を与えなければならない (図1中の③)．

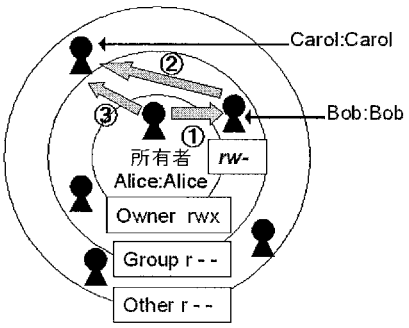

図 1: ACL機能の使用例

ファイルやディレクトリの所有者が，他のユーザにアクセス権限を与え，アクセス権限を与えたユーザは，権限を与えられたユーザが第三者のユーザにアクセス権限を与えることを許可する．このとき，ACL機能を用いて，アクセス制御が設定可能な権限を与えることはできないため，所有者がユーザ毎にアクセス権限を設定しなければならず，所有者の負担が増大する．

3 システム構成

図2に，提案システムの構成例を示す．システムの構成要素は，クライアント，提案システム，Webサーバの3つである．クライアントとWebサーバ間はHTTP通信をしており，この2つの間にアクセス制御機能を持ったリバースプロキシとして，提案システムを配置する．図中の実線は，ACM(後述) を経由するクライアントとWebサーバ間のHTTP通信を表し，破線はクライアントとEUI(後述) 間のHTTP通信を表している．以下に，システムの各構成主体について示す：

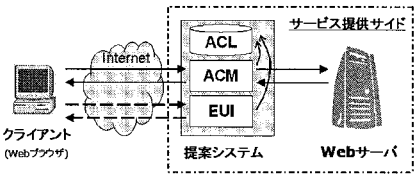

図 2: 提案システムの構成例

クライアント

Internet Explorer[4]やNetscape[5]等のWebブラウザである．クライアント数は1台以上である．

提案システム

Webブラウザから送信されたHTTPリクエストを基に，Webサーバ上のコンテンツに対するアクセス制御を行う (この処理を行う提案システム上のプロセスを，ACMと呼ぶ)．アクセス制御を行う際に参照する，コンテンツに対するアクセス制御ポリシーが記述されたリスト (以降，ACLと呼ぶ) を格納している．ACLの実現には，RDBMS (Relational DataBase Management System) を利用した．また，ユーザがインターネット越しにACLを編集するため，ACL編集用のユーザインタフェース (以降，EUIと呼ぶ) を提供する．

ACMは，Fedora Core 4 (kernel 2.6.11) 上に，Java (J2SE SDK, v1.4.2_12) を用いて実装し，ACLはMySQL 5.0.22[6] を用いた．EUIは，JSP (Java Server Pages) によって実装し，そのサーブレットコンテナとWebサーバには，Tomcat 5.0.28[7] とApache 2.2.2[8] をそれぞれ利用した．

Webサーバ

Apache-2.2.2が動作している．サーバ内には，

ユーザのコンテンツであるファイルやディレクトリが保存されている．また，Basic 認証 [9] を設定してある．

4 提案システム

本章では，提案システムの説明をする上で必要となる用語の定義を述べ，次に，以下に示す2つの要件を満たす本システムの機能や動作について述べる．

(要件 1) 管理者権限のないユーザによる，アクセス制御ポリシーの設定や変更ができること

(要件 2) アクセス制御が設定できる権限を，ユーザ間で委譲できること

4.1 用語の定義

提案システムでは，ユーザがアクセス制御ポリシーの設定または変更をすることが可能である．このとき，ユーザに与えられた，アクセス制御ポリシーの設定または変更ができる権限のことを，「アクセス制御ポリシーが設定可能な権限」と呼ぶ．この権限を付与されているユーザは，指定されたリソースに対するアクセス制御の設定やアクセス制御ポリシーの変更をすることができる．また，提案システムでは，「アクセス制御ポリシーが設定可能な権限」をユーザ間で委譲することが可能である．以降，これを「権限を委譲する」と表現する．権限を委譲されたユーザは，「アクセス制御ポリシーが設定可能な権限」を得るだけでなく，第三者のユーザに，この権限を委譲することができる．

4.2 提案システムの機能と動作

提案システムは，以下の3つの機能がある：

1. アクセス制御を行う機能
2. ACL の作成や編集を行う機能
3. 権限を委譲する機能

アクセス制御は，ACM と ACL を用いて行う．詳しい説明は，4.3 副節で述べる．ACL の作成や編集，権限の委譲は，EUI と ACL を用いて行う．これらの説明は，4.4 副節と 4.5 副節にて述べる．

4.3 アクセス制御

本節では，まず ACL の構造を説明し，次に，ACL を用いた ACM でのアクセス制御について説明する．

4.3.1 ACL の構造

提案システムの ACL は，MySQL のテーブルの集合で表される．まず，Web サーバ上のディレクトリと同じ名前のテーブルを1つずつ用意する*．そして，そのテーブルのカラムに，テーブル名に対応するディレクトリ配下のリソースのアクセス制御ポリシーを記述する．図 3 に，テーブルのフィールドを示し，以下に各フィールドの役割を示す：

file	allow	deny	delegate	owner

図 3: テーブルのフィールド

file: ファイル名または，ディレクトリ名を記述する．

allow: file フィールドに記述されたリソースに対してアクセスを許可するユーザを記述する．アクセス権限は，「読み込み」と「書き込み」の2つがあり，それぞれ r と w のフラグ表記で指定する．例えば，「Carol:rw」のように記述する．

deny: file フィールドに記述されたリソースに対してアクセスを拒否するユーザを記述する．上述の allow フィールドの場合と同様，このフィールドも r と w のフラグ表記を用いる．例えば，「Carol:-w」のように記述する．

delegate: file フィールドに記述されたリソースのアクセス制御ポリシーが設定可能な権限を，他のユーザに委譲する場合に利用するフィールドであり，ここには権限を委譲されるユーザを記述する．権限は2種類あり，フラグ O(owner) とフラグ A(append) のどちらかを指定しなければならない．これらのフラグの詳細については，4.5 副節で説明する．例えば，「Bob:O」のように記述する．また，オプションとして，権限を委譲できる階層数を指定することが可能である†．これについての詳しい説明

*ただし，ルートディレクトリのアクセス制御ポリシーを記述するためのテーブル名は，「Top_dir」である．

†委譲できる人数ではなく，委譲できる階層数を記述する．

も 4.5 副節で述べる．例えば，「Bob:O3」と記述する．

owner: file フィールドに記述されたリソースの所有者を記述する．これは，このリソースに対するアクセス制御の権限を委譲されたユーザが，その所有権を奪取し，本来の所有者がそのリソースに対するアクセス制御ポリシーの変更ができなくなることを防ぐために利用する．

allow，deny，delegate の各フィールドに記述するユーザ名は Basic 認証におけるユーザ名を利用しており，1 つのフィールドに，複数のユーザ名を記述する場合は，カンマで区切る．また，allow と deny フィールドに記述される All という表現は，すべてのユーザが対象であることを示す．そして，All は，allow と deny フィールドのどちらかに記述しなければならない．さらに，allow と deny フィールドのアクセス権限のフラグ表記のうち，「-」と表記されている部分は，そのフラグのアクセス権限が無効であることを示す．

例えば，Web サーバ内に，図 4 に示すようなファイルおよびディレクトリが格納されている場合，ACL の記述例は図 5 のようになる．

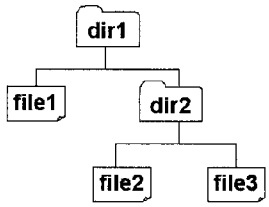

図 4: ディレクトリ階層例

table: Top_dir

file	allow	deny	delegate	owner
dir1	All:rw			Alice

table: dir1

file	allow	deny	delegate	owner
file1	All:rw	Carol:rw		Alice
dir2	All:rw		Bob:O	Alice

table: dir2

file	allow	deny	delegate	owner
file2	All:rw			Alice
file3	All:rw			Alice

図 5: ACL のテーブル例

4.3.2 ACM の動作

ここでは，ACM の処理を交えて，提案システムでのアクセス制御について説明する．ACM は，クライアントと Web サーバ間の HTTP 通信を中継しつつ，クライアントから送信される Basic 認証におけるユーザ名を基に，アクセス制御を行う．

図 6 に ACM を経由する，クライアントと Web サーバ間の HTTP 通信の流れを示し，以下にその詳細を示す．図中の (1) ～ (9) が，詳細説明の番号に対応している．前提として，Basic 認証の認証情報は Web サーバとクライアント間で予め共有されているものとする：

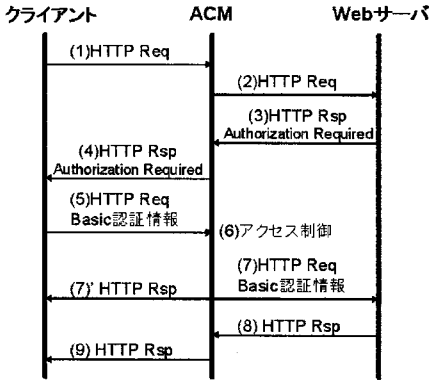

図 6: ACM を経由する HTTP 通信の流れ

(1)(2) クライアントは ACM にファイルの取得要求を送信し，ACM はそれを Web サーバに転送する．

(3)(4) Web サーバは，"Authorization Required" を応答として送信し，ACM はそれをクライアントに転送する．

(5) クライアントは，認証情報がヘッダに含まれた HTTP リクエストを ACM に再送信する．

(6) ACM は，クライアントから送信された認証情報と ACL を用いて，アクセス制御を行う．まず，HTTP リクエストに含まれるファイルパスのルートディレクトリに対応する ACL のアクセス制御ポリシーを確認する．allow と deny フィールドを確認し，アクセス権限を判定する．同様に，ファイルパスの次のファイルまたはディレクトリに対応する ACL のアクセス制御ポリシーを確認していく．これを，ファイルパスの最後のリソースを確認するまで，または，クライアントの読み書きのアクセスを拒否すると判断するまで繰り返す．

(7) ACM が (6) の処理で，アクセスを許可すると判断した場合，ACM は，(5) の処理でクライアントから ACM に送信された HTTP リクエストを Web サーバに転送する．
(7)' ACM が (6) の処理で，アクセスを拒否すると判断した場合，ACM はアクセス拒否を表す通報をクライアントに送信する．
(8)(9) Web サーバは，HTTP リクエストに対するレスポンスを送信し，ACM はそれをクライアントに転送する．

4.4 ACL の作成・編集

ACL の作成と編集は，ユーザインタフェースである EUI を経由して行う．Web サーバ上にディレクトリまたは，ファイルを作成するたびに，ユーザは EUI を通して，ACL を編集する．

ACL にアクセス制御ポリシーが設定されていない状態を，図 7 に示す．ルートディレクトリのアクセス制御ポリシーを記述するための Top_dir という名前のテーブルだけが存在し，そのテーブルにアクセス制御ポリシーは記述されていない．

table: Top_dir

file	allow	deny	delegate	owner

図 7: ACL の初期状態

次に，ACL の具体的な編集方法について説明する．4.3.1 副々節で述べた通り，allow と deny フィールドのどちらかに，必ず「All」と記述しなければならない．そこで,「All」を allow フィールドに記載する場合と deny フィールドに記載する場合の ACL の設定方法について示す．

allow フィールドに All と記述する場合

allow フィールドに「All」と記述する場合，「All」と共に設定できるアクセス権限は「rw」のみとした‡．そのため，allow フィールドには，「All:rw」と記述する．さらに，deny フィールドに，アクセスを拒否するユーザ名とアクセス権限を記述する．deny フィールドに記述できるアクセス権限は,「rw」と「-w」とした．例えば，図 8 のように設定する．これは，dir1 に対するユーザ Carol の書き込みのアクセスのみを拒否し，その他のユーザのアクセスは許可するということを意味している．

‡アクセス権限「r-」と「-w」は設定不可である．

table: ——

file	allow	deny	delegate	owner
dir1	All:rw	Carol:-w		Alice

図 8: allow に All:rw と記述した場合の ACL 例

deny フィールドに All と記述する場合

deny フィールドに「All」と記述する場合,「All」と共に設定できるアクセス権限は「rw」と「-w」とした．deny フィールドに「All:rw」と記述する場合，allow フィールドに，アクセスを許可するユーザ名とアクセス権限を記述する．このとき，allow フィールドに記述できるアクセス権限は,「rw」と「r-」である．例えば，図 9 のように設定する．これは，dir1 に対するユーザ Bob の読み書きのアクセスを許可し，その他のユーザのアクセスは拒否するということを意味している．

table: ——

file	allow	deny	delegate	owner
dir1	Bob:rw	All:rw		Alice

図 9: deny に All:rw と記述した場合の ACL 例

deny フィールドに,「All:-w」と記述する場合，allow フィールドにアクセスを許可するユーザ名とアクセス権限を記述する．さらに，deny フィールドに読み書きのアクセスを拒否するユーザ名とアクセス権限を記述する．このとき，ユーザ名と共に記述できるアクセス権限は，allow, deny フィールド共に,「rw」のみである．例えば，図 10 のように設定する．これは，dir1 に対するユーザ Bob の読み書きのアクセスを許可し，ユーザ Carol の読み書きもアクセスを拒否する．その他のユーザは，書き込みのアクセスのみ拒否するということを意味している．

table: ——

file	allow	deny	delegate	owner
dir1	Bob:rw	All:-w, Carol:rw		Alice

図 10: deny に All:-w と記述した場合の ACL 例

4.5 アクセス制御の権限委譲

アクセス制御ポリシーが設定可能な権限の委譲は，EUI を経由して，ACL の delegate フィールドを編集することで行う．権限の委譲を行うため，権限を委譲するユーザが，委譲されるユーザ名と権限 (O 権限または A 権限) を，delegate

フィールドに記述する．権限を委譲されたユーザは，自分の与えられた権限内で，以下を行うことができる：

- あるディレクトリやファイルにおいて，O 権限を付与されたユーザ§
  - 設定済みのアクセス制御ポリシーの追記，削除
  - 他のユーザへの O 権限または A 権限の委譲
  - ディレクトリまたはファイルの新規作成
- あるディレクトリやファイルにおいて，A 権限を付与されたユーザ
  - 設定済みのアクセス制御ポリシーの追記
  - 他のユーザへの A 権限の委譲
  - ディレクトリまたはファイルの新規作成

「設定済みのアクセス制御ポリシーの追記」は，既に設定されているアクセス制御ポリシーに，新たにアクセス制御ポリシーを追加することができることを意味している．「設定済みのアクセス制御ポリシーの削除」は，既に設定されているアクセス制御ポリシーを削除することができることを意味している．「他のユーザへの権限の委譲」は，権限を委譲されたリソース配下にあるリソースに対する，アクセス制御が設定可能な権限を，他のユーザへ委譲することができることを意味している．権限を委譲された場合，その権限は，権限を委譲されたリソースの配下にある全てのリソースに適用される．また，他のユーザに権限を委譲しても，自らの権限がなくなることはない．「ディレクトリまたはファイルの新規作成」は，予め Web サーバ内に格納されているリソースの配下に，新たにディレクトリやファイルを作成できることを意味している．これを行うと，ACL に，作成したリソースに対応するテーブルやカラムが追加され，このリソースの上位のリソースのアクセス制御ポリシーが，このリソースのアクセス制御ポリシーとして引き継がれる．

4.3.1 副々節で述べた通り，オプションとして，アクセス制御が設定可能な権限を委譲するとき，権限を委譲できる階層数を指定することができる．例えば，アクセス制御が設定可能な権限をユーザ Alice からユーザ Bob に委譲し，Bob がユーザ Carol に委譲した場合，Alice から見ると自分から Bob へ，Bob から Carol へと，2 回の委譲が起こったので，階層数は 2 となる．delegate フィールドに記述する O または A 権限の後に，階層数を指定する．権限の後ろに数字が記述されていない場合は，無限に委譲できることを意味している．例えば，ユーザ Alice が，あるリソースの権限をユーザ Bob に，A 権限を付与して委譲し，Alice は，Bob がその他のユーザに A 権限を委譲することを拒否したいとき，delegate フィールドに「Bob:A0」と記述する．このようにして，権限がリソースの所有者が知らないところまで委譲されることを，阻止することができる．

提案システムでは，このようにして，アクセス制御が設定可能な権限の委譲を，ユーザ間で行うことができる．

§そのディレクトリやファイルの所有者も含む．

5 まとめ

提案システムでは，Web サーバ内に格納されているディレクトリやファイルに対して，管理者ではない一般ユーザが，インターネット越しにアクセス制御ポリシーの設定を行うことができる．また，ユーザ間で，アクセス制御ポリシーが設定可能な権限の委譲をすることが可能である．本論文では，これらを機能を備えたプロキシ型のアクセス制御方式の提案と実装を示した．提案システムを用いることで，管理者を必要としない，アクセス制御ポリシーの設定が可能となる．また，管理者が Web サーバ内のディレクトリやファイルのアクセス制御を全て設定しなくてもよいので，管理者の負担を軽減させることができる．今後の課題として，EUI の拡張がある．

参考文献

[1] http://xoopscube.org/
[2] http://pukiwiki.sourceforge.jp/
[3] http://wwww.kernel.org/
[4] http://www.microsoft.com/
[5] http://www.netscape.com/
[6] http://www.mysql.com/
[7] http://tomcat.apache.org/
[8] http://www.apache.org/
[9] http://httpd.apache.org/docs/1.3/howto/auth.html