信頼の輪モデルに基づいた個人認証手法の運用

正岡 元, 菊池 豊

高知工科大学大学院 工学研究科 基盤工学専攻 情報システム工学コース

概要

信頼の輪は、個人と個人との信頼関係の合成によって構成される個人認証のモデルである。本研究では、この信頼の輪モデルに基づいてホストの利用権限をユーザに委譲する個人認証手法を提案した。

この手法により、従来ホストの管理者に集中していたユーザ管理コストを分散させることが可能となる。今回この手法を実装することにより、この手法が運用可能であることを示すと共に、従来の手法と比較してどの程度の効果が期待できるのか検証を行った。

Employing A Person Authentication Method Based on the Web of Trust Model

MASAOKA Hajime, KIKUCHI Yutaka

Information Systems Engineering Course, Department of Engineering,
Graduate School of Engineering, Kochi University of Technology

Abstract

The web of trust is a model that composed of trust relations among users. This research proposes a person authentication method using authority delegation based on the web of trust.

This proposal introduces the web of trust model to the conventional administration method. The method is based on trust of users who already have permissions. Moreover the method gives a function that restricts users' permissions according to the trust.

We have implemented the method to hosts so that we will show the effectiveness of the proposed way as compared with the conventional authentication method.

1 はじめに

信頼の輪は、個人と個人との信頼関係が複数存在する際に、それらの合成によって構築される個人認証のモデルである。本研究では、この信頼の輪モデルを基にしてホストにおける利用権限をユーザに委譲することによって個人認証を行う手法を提案した [2]。この認証手法を、信頼の輪認証と呼ぶこととする。

従来のユーザ管理手法では、ユーザ管理のコストはホストの管理者 (以降、管理者と表記) に集中していた。提案手法を用いることでユーザ管理コストを管理者以外のユーザに分散させることにより、管理者の負担するユーザ管理コストを低減することが可能となる。

今回この手法を運用し、この手法が運用可能であることを示すと共に、従来の手法と比較してどの程度の効果が得られるのか検証を行った。

2 信頼の構造

本研究における認証手法は信頼関係をベースにしている。本節では、この信頼の構造について述べる。

2.1 信頼の定義

本手法では、ユーザが別のユーザを信頼することで自らが持っている権限を別のユーザに与えることができる。この権限を与える行為を、権限の委譲と呼ぶ。この時委譲される権限は信頼する側が元々持っている権限と信頼度とに基づく。

信頼度とは、信頼の程度を 0%から 100%の間で数値化したものである。信頼度が高いユーザは、信頼度が低いユーザより多くの権限を得る。例えば信頼度 100% のユーザは、信頼したユーザの持つ全ての権限を得る。また、信頼度 0%のユーザは権限を全く得ることができない。

2.2 信頼の結合

ユーザ同士は信頼関係を作ることができる。複数の信頼関係の結合により、より複雑な構造を持つ。この構造を信頼の網構造と呼ぶ。信頼の網構造によって、直接知らないユーザを間接的に信頼することができる。

網構造を図 1 に示す。図中の矢印は根元の人物が先の人物を信頼している状態を示す。この網構造を持つ信頼の集まりを、信頼の輪と呼ぶ。信頼の輪の代表例として OpenPGP[1][1]における鍵リングがあげられる。

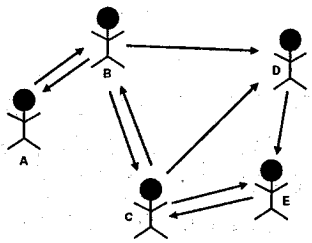

図 1: 信頼の網構造

3 権限の委譲手法

まず認証に対して求められる事柄を第 3.1 節にまとめ、既存の認証手法の問題点を第 3.2 節にあげ、その改善手法を第 3.3 節で提案する。

3.1 必要な認証システム

ホストの利用権限には、コマンドの実行権限やファイルの読み書きの権限、印刷の権限、扱えるファイルの大きさなど、さまざまなものがある。そしてそれらの権限は各ホストごとに設定する必要がある。そのため、複数のホストに新たなユーザを登録する際には、管理者はあらかじめ各ホストの権限に関する情報を調べておく必要がある。たとえば、各ホストにおいて利用できるソフトウェア、存在するファイルの種類、スキャナやプリンタなど権限に関係するデバイスの有無、そして登録されるユーザが求める権限、などである。

管理者はこれらの情報を元に、各ホストにおいてそれぞれ与える権限を決定する。この時、不要な権限を与えないように注意することが望ましい。

[1] http://www.openpgp.org/

次に、決定した権限を各ホストにおいてユーザに適用する。このユーザ登録に必要な手順を以下に整理しておく。

- 情報の収集
  - 各ホストにあるソフトウェア
  - 各ホストにあるファイルの種類
  - 各ホストに接続されたデバイスの種類
  - ユーザの求める権限
- 与える権限の決定
  - 不要な権限を与えない
- 各ホストに権限を設定

3.2 既存の認証手法の問題点

既存の認証手法において、第 3.1 節で述べた点を実現する際に問題となる点を示す。ここでは既存の認証手法として Unix を取り上げる。

Unix のユーザは、大きく特権ユーザと一般ユーザとの 2 つに分けられる。特権ユーザはホスト上の全ての権限を持つ。一般ユーザの利用権限の制限は、ファイルのオーナー、グループ、その他、という 3 つの単位に対する読み書きや実行の許可、不許可によって行う。

この様な Unix での権限の制御機能では、各ユーザに対して異なる権限を適用するには複雑なグループ管理が必要になり、管理コストが増大する。さらにこれらの権限を各ホストにおいてユーザに適用するためには、ホストの台数に応じた管理コストを必要とする。

NIS を用いることで、マシンの増加による管理コストの増大は解決できる。しかし NIS では、Unix での権限の制御機能が持つ欠点を回避することはできない。

そのため、不必要な権限を与えずに必要な権限のみを複数のホストにおいて適用することは困難である。

3.3 改善手法の提案

第 3.2 節で述べた、管理者がユーザの権限に関する管理の全てを行う手法では要求を満たせない。我々はこれを改善する信頼の輪認証と呼ぶ手法を提案した [2]。

ここに、ホストを利用できるユーザ A がすでにいると仮定する。本方式は、ユーザ A がユーザ B を信頼している場合に、この信頼関係に基づいて、ユーザ A の持つ権限をユーザ B に委譲する手法である。この時、委譲する権限はユーザ B の信頼度によって制限する。

ここで、委譲する権限は、OS におけるコマンドの実行権限である。管理者は全てのコマンドの実行権限を持っている。本手法による権限の委譲は Unix コマンドとして実装し、このコマンドの実行権限も委譲することができる。

管理者は OS の持つコマンドをグループ化し、そのコマンドグループ単位で権限をユーザに委譲する。これは、そのユーザが扱うことのできる権限の範囲を明確にするためである。コマンドをグループ化する例を図 2 に示す。

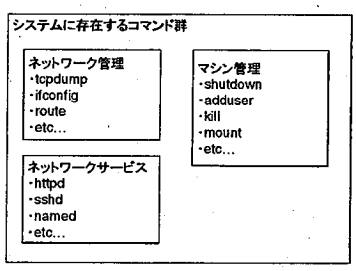

図 2: コマンドのグループ化の例

また、ある信頼度を持つユーザに対してどれだけの権限を与えるか、という情報もあらかじめ管理者が用意する。信頼度と与えられる権限との関係を、マシン管理コマンドのグループを例にして図 3 に示す。

4 提案の実装

本節では今回運用した実装について述べる。過去の実装では基本的な機能のみを実装し、ローカルな環境において動作可能であることを示した。今回は、運用に必要な機能を追加し、複数のホストにおいて運用を行った。

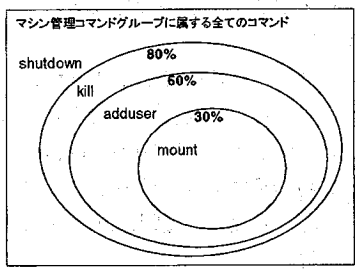

図 3: 信頼度と権限との関係

4.1 仕様

第 3 節に述べた認証手法を実現するための仕様を述べる。

このプログラムは信頼関係に基づいて権限の委譲を行う。また、委譲する権限は信頼度に基づいて決定する。そのために、ユーザ間の信頼関係とその信頼度をデータとして持つ。権限の委譲には sudo[2]を用いる。sudo は多くの Unix または Unix 互換 OS(以下、Unix 系 OS と表記) で動作するアプリケーションであり、一般ユーザに対して管理者の権限を適切に与えるツールとして利用されている。sudo は一つ一つのコマンドの実行権限を、ユーザごとに設定することができる。

あらかじめ管理者が設定しておく情報は以下の通りである。これらの設定は専用のファイルに記述しておく。

- コマンドグループの設定
- 信頼度に応じて与える権限

ユーザが他のユーザに権限を委譲する際は、適切な信頼度を設定し、そのユーザとの信頼関係としてファイルに記述する。

プログラムは、ファイルに記述されているユーザの信頼関係を調べ、適切な権限が適用された sudo の設定ファイルを出力する。プログラムが実行されるとファイルから信頼関係を読み込み、誰が誰を信頼しているのか確認する。次にプログラム実行前の sudo の設定ファイルを元に、信頼する側の持つ権限を確認し、その権限を信頼される側のユーザに対して適用する。与える権限は信頼度に応じて縮小する。この動作メカニズムを図 4 に示す。図中の番号に従って、以下のように動作する。

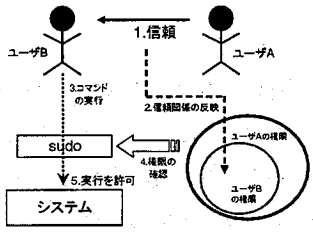

図 4: 動作メカニズム

1. ユーザ A がユーザ B を信頼していることをファイルに記述する
2. 本プログラムが信頼関係を sudo の設定ファイルに反映する
3. ユーザがコマンドを実行しようと試みる
4. sudo が設定ファイルを読み、権限を確認する
5. 与えられた権限に基づき、コマンドは実行される

1 がユーザ A によって行われた後で、管理者によって 2 が実行される。これは管理者自身が手動で実行するか、あるいは cron の設定により自動的に実行されるようにする。

4.2 設計

本節では、プログラムの設計について述べる。

今回実装したプログラム (以下、新実装と表記) の持つ機能を以下に示す。本手法を提案した際に検証用に作成したプログラム (以下、旧実装と表記) においてすでにあった機能 (以前からある機能) と、今回新たに実装した機能とに分類して示す。

- 以前からある機能
  - 各ユーザの信頼関係を読み取る
  - 信頼関係に基づく新たな権限の割り当て
  - sudo の設定ファイルとして書き出す

[2]http://www.courtesan.com/sudo/

- 今回新たに実装した機能
  - 信頼度の設定
  - 権限の委譲の流れを表示する
  - 指定したユーザの権限を剥奪する

プログラムを実行すると、まず管理者の信頼関係を記述したファイルを読み込み、管理者が信頼するユーザを確認する。次に管理者が信頼する各ユーザのホームディレクトリからファイルを読み込み、そのユーザが信頼するユーザを確認する。以降、再帰的に信頼するユーザを確認し、信頼の輪のトポロジーを作成する。最後に、ユーザが権限を得るためのsudoの設定ファイルを書き出す。

信頼関係は各ユーザのホームディレクトリにあるファイルに記述する。ファイルには、そのユーザが誰をどの程度信頼しているか(信頼度)を記述してある。あるユーザXが、ユーザYを完全に(100%)信頼しており、ユーザZをある程度(50%)信頼している状態を記述した例を表1に示す。

Y	100
Z	50

表1: 信頼関係の記述

新実装における新たな機能は主に3つある。まず、信頼度の実装である。この機能により不要な権限を与える必要が無くなり、より安全に権限を委譲することが可能となる。

次に、誰が誰を信頼しているか、ある権限は誰によって与えられたか、という情報を表示できる。これにより、管理者の意図しない権限委譲の発見が容易になる。

また、意図せず委譲された権限を、明示的に与えないことにする設定が可能である。これは信頼の記述とは逆に、信頼しないことの記述によって行う。例えば、ユーザAがユーザBを信頼しないと設定した場合、ユーザBはユーザAの持つ権限を全く得ることができない。

4.3 実装

本プログラムの実装について述べる。実装したOSはFreeBSDである。実装言語はrubyであり、プログラム全体の構成は以下の通りである。

- pga.rb
  プログラム本体(400行のrubyスクリプト)
- cmdgroup.conf
  コマンドグループの設定ファイル
- ~/.trust
  各ユーザごとの信頼関係の設定ファイル

5 考察

第4.3節に述べた実装は、基本的な仕様が正しく動作することを確認した。各ユーザの信頼に基づいて権限が与えられ、管理者が作業を行うことなくユーザは必要な権限を新たに得ることができた。

本節では、今回運用した内容について考察する。本手法は実際の運用に耐えうるのか。本手法の目的である、管理者の負担するユーザ管理コストの低減が実現可能なのかどうか。本手法および実装の安全性に問題は無いか。これらの事柄についての考察を以下に述べる。

5.1 管理コスト

本手法では、ユーザ登録に際して管理者は必要な権限を考慮する必要は無く、単にユーザを追加するのみでよい。必要な権限は、すでに権限を持っているユーザが新たなユーザに対して与える。そのため、ユーザ登録に際しての管理者の負担は低減されたといえる。

しかし、権限の追加は一般ユーザによって行われるものの、ユーザの登録は管理者が行う必要がある。これは権限の委譲にsudoを利用しているためである。sudoはすでに存在するユーザにコマンドの実行を許可するため、登録されていないユーザに対する権限の委譲はできない。

本手法では、従来のUnix系OSにおける手法に比較すれば管理コストが低減されているものの、単にユーザを追加するだけとはいえ複数のホストへのユーザ登録は大変である。更なる管理コストの低減には、ユーザ登録の仕組みそのものを分散管理する手法が必要である。

5.2 安全性

本手法では、一般ユーザによって権限が委譲される。そのため管理者の意図しない権限の委譲が行われてしまう可能性がある。旧実装では意図しない権限の委譲を発見することが困難であった。また、発見してもそれを止められなかった。

新実装では、意図しない権限の委譲を容易に発見することができる。さらに、明示的に権限を与えない機能により、万一意図しない権限の委譲が行われてしまった場合にも安全を確保することが可能である。しかし、定期的な信頼関係の検査や必要に応じた権限の剥奪など、管理コストが増加する可能性は残る。

別の問題点として信頼度と権限との関連付けがある。ある信頼度を持つユーザに対してどのような権限を与えるかを、あらかじめ管理者が決定しておく必要がある。ここで、この関連付けが適切に行われていないと、不要な権限の委譲が行われる可能性がある。

5.3 実装言語

今回の実装の目的は、本手法の運用可能性を確認することと、従来手法に対する管理コストの比較であるため、実装言語にはrubyを利用した。ただし、多くのUnix系OSではrubyは標準配布物ではない。そのため本プログラムだけでなく、rubyのインストールも必要となる。ユーザ認証はOSの基本的な機能であるため、本手法の運用はOSをインストールした後早い段階で開始する必要がある。そのため、OS標準の環境で動作する実装が望ましいといえる。

5.4 実装の範囲

ホストを利用するユーザに適切な権限を与えるためには、各ユーザのログイン時にsudoの設定ファイルを再生成することが望ましい。しかし、ホストには、コンソール、XDM、リモートログインなど様々なログイン方法がある。これらはそれぞれ違う手段で認証を行うため、これら全てにログイン時に設定ファイルを再生成する仕組みを付加する必要がある。しかし今回はこれらの認証手法に対する変更は加えず、sudoの設定ファイルを生成する部分のみを実装した。これは提案した手法を実現するのに最低限必要な部分であるとともに、この部分だけでも提案を評価することが可能であるためである。

6 まとめ

本研究では、個人間の信頼関係に基づいて構築された信頼の輪の概念を用いることにより、適切なホストの利用権限を与える手法を示した。信頼の輪は、個人と個人との信頼関係が複数存在する時、それらの合成によって構築される個人認証のモデルである。このモデルによって、直接知らない相手を、間接的に信頼することができる。

この手法を実装し実際に運用することによって、提案手法が運用可能であることを示した。さらに、本手法によって実際に管理コストの低減が可能であることを示した。

この手法を用いることで、従来管理者に集中していていたユーザ管理のコストを分散することが可能となる。さらに従来の手法では困難であった、ユーザに与える権限を細かく制御することも可能となる。本手法は、本来不要である権限を割り当ててしまう従来の手法に比べ安全にユーザ管理を行うことができる。

今後、OSのインストール直後から本手法が運用できるように、Unix系OSにおいて標準的に利用できる環境のみで動作できるように改変を行う予定である。

参考文献

[1] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP Message Format, November 1998. RFC2440.

[2] 正岡元, 菊池豊. 信頼の輪モデルに基づいたシステム利用権限の委譲による個人認証手法. 情報処理学会研究報告. 分散システム運用技術研究会, May 2003.