社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2005－OS－99（25）
2005／5／27

オブジェクトを隠蔽可能なファイルシステムの実現

田中 大樹　　小林 良岳　　中山 健　　前川 守

電気通信大学 大学院情報システム学研究科 情報システム設計学専攻

E-mail: {htanaka,yoshi,ken,maekawa}@maekawa.is.uec.ac.jp

概 要

セキュリティ上の問題として，ファイルやディレクトリなどのオブジェクトの存在が見えてしまう事が問題となる場合がある．そこで本稿では，任意のオブジェクトの存在を隠蔽する事を可能にするファイルシステムの提案を行う．このファイルシステムは，オブジェクトの内容流出を防ぐ機構も持つ．

A File System that Supports Concealed Object

Hiroki TANAKA　　Yoshitake KOBAYASHI
Ken NAKAYAMA　　Mamoru MAEKAWA

Department of Information Systems Science, Graduate School of Information Systems,
University of Electro-Communications

Abstract

It becomes a security issue that other users can check object existence. In this paper, we describe a file system that supports concealed objects, such as files and directories. This file system also has a mechanism which prevents a content outflow for the concealed object.

1　背景

ファイルの改竄や盗難といった，セキュリティ上の問題は古くから存在する．IT 化が進む昨今にあっては，それらはより重大な問題になっている．そのため，認証や暗号化といった防衛機構が用意され，不正を未然に防ごうという試みが一般化しているが，それら防御機構が突破されるケースも多く存在する．

この防御機構を突破されたケースの中には，インターネットを経由して，情報が引き出されるという問題も含まれる．この代表例としては，ユーザが外部からアクセスされるべきでないファイルを誤って Web サーバなどに配置してしまい，これにより機密が外部に漏れてしまうものが有り，このケースは多く報告されている．

ユーザが判断を誤る理由の一つには，OS の利便性や操作性が，大衆化に追いついていない事が挙げられる．故に，これらを防止するより強力かつ柔軟で明快なファイル保護機構が求められている．

そこで，本研究では，ファイルやディレクトリなどのオブジェクトの存在を隠蔽，並びにデータの流出防止，そしてユーザが明快に利用可能な保護機構を実現することを目的とする．

2　関連研究

ACL(Access Control List) は，商用 UNIX などでは古くから存在する機構であり，最近では Linux カーネル 2.6 に標準採用された．これを用いることで，各ファイルに対して，UNIX 標準のオーナー/グループ/第三者による，パーミッション指定よりも柔軟であり，より自由なアクセス権制御を行う事が可能である．アクセス権制御は可能であるが，ファイルの存在は制御の対象ではない．

chroot(change root directory) は，任意のディレクトリをルートディレクトリとして設定するもので，そこより上層にあるディレクトリを隠し，アクセスを遮断する．ディレクトリ単位での制御であり，下層にライブラリやその他一般使用の為の実行ファイル群を必要とする．

jail[1] は，chroot の発展形で，ユーザディレクトリ間に破ることのできない壁を作り，お互い不可侵な環境を作成する．これはディレクトリだけでなくネットワーク設定やプロセスにも対応する．chroot と同じくディレクトリ単位であり，やはり下層にライブラリ群などが必要になる．

Wrappers[2] は，OS レベルのラッパーであり，アプリケーションの動作制御を目的としている．アプリケーションレベルの Proxy などは脆く，それ

でいて複雑化してきている．欠陥のある Proxy が一つあると全体に悪影響が出るため，ラッパーを用いて動作仕様を定義している．ただしラッパーは，システムコールを素通しするという問題がある．

3 システム概要

2 節で述べた関連研究のうち，ACL はアクセス権の設定のみであり，また chroot/jail では，ディレクトリ単位でしか設定できないなど，決して柔軟であるとは言えない．さらに，どちらもデータの流出を防ぐ機能は持っていない．

データがファイルとして (例えばディスクに) 存在するということが問題であり，重要であればあるほど，存在することを他者に認知されてはならない (図 1)．既存の UNIX のパーミッションでは，ディレクトリ単位では，内部のファイルを閲覧不可能な状態にする事は可能だが，ディレクトリ名は見えてしまう，またアプリケーションによっては固定のパスを必要とする場合もある．このため，ファイル単位での隠蔽を可能にする必要性が生まれる．同時に，ユーザが誤って機密情報が含まれるファイルを外部からアクセス可能な場所に置いてしまった場合でも，そのファイルが実際にネットワークなどへ流れる事を水際で防止する機構が必要である．

そこで本研究では，ファイルやディレクトリといったオブジェクトの存在そのものを隠蔽し，かつ柔軟なアクセス制御をユーザに提供する機構の作成を行う．

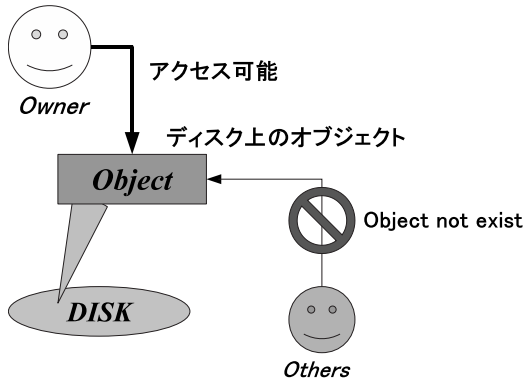

図 1: 隠蔽イメージ

3.1 存在の隠蔽

許可が無い限り存在そのものを他のユーザから隠蔽する．名前が見えているだけであっても，そこから内容の類推が可能であるため，それを防ぐためにも”Object not exist”の状態を目指す．また，第三者に管理者権限を乗っ取られる可能性や，管理者自体が，あらゆる時点で信頼できる存在であるとは限らないので，管理者に対しても隠蔽を行う必要がある．

3.2 入出力制御

ファイルが内部の第三者ユーザからは見えない状態であっても，それだけでは安全であるとは言い切れない．特に昨今はネットワークからアクセスにより情報が盗み出されるケースが多い．そこでネットワークに流しても良いファイルであるか否かなどの設定を行えるようにする．これにより，ユーザが誤った操作を行った場合でも，情報の流出を防ぐ事が可能となる．

4 設計と実装

オブジェクトをユーザから隠蔽するための手段として，本研究ではシステムコールを制御することで，これを実現する．また情報の流出を抑制するために，デバイスドライバに対しても制御を行い，安全性の向上を目指す．

4.1 SystemCall Wrapper

ラッパーを構築・設置し，アプリケーションとファイルシステム間で発せられるシステムコール (特に，READ 命令など) の制御を行う．これにより，許可無きユーザからのオブジェクト参照を防ぎ，存在の隠蔽を実現する．デフォルト設定はオーナー以外は不可視とし，ユーザは必要に応じて開放を行う事が可能である．

4.2 DeviceDriver Wrapper

デバイスドライバの呼び出しを制御し，FD やネットワークといった外部へのデータの流れを制御する．ユーザが誤って公開ディレクトリにファイルを置いた場合などでも，通信を遮断することで，情報の流出を防ぐことが可能となる．オブジェクト単位で，デバイスドライバの使用の可否を設定し，柔軟な制御を実現する．

5 まとめ

本研究では，柔軟に設定可能なラッパーを用いてシステムコールとデバイスドライバのデータフローに抑制を加えることで，情報を保護することを提案した．

その一方で，設定すべき事柄が増え，操作が複雑になることは，人為的なミスを誘発する原因になりかねない．しかしながら，操作が簡単になれば，それは悪意ある第三者に対しても，門戸を大きく開くことになりかねない．どちらに舵を切るかで手法や様相が異なるため，十分な検討が必要である．

参考文献

[1] Jail Chroot Project, http://www.jmcresearch.com/projects/jail/

[2] Jeremy Epstein, Linda Thomas, Eric Monteith: Using Operating System Wrappers to Increase the Resiliency of Commercial Firewalls, Proceedings of 16th Annual Computer Security Applications Conference, pp.236-245, Dec, 2000.