セキュアOSにおける動的アクセス制御の実装

香取知浩　杵渕雄樹　神田渉　湯村悠　中島達夫
早稲田大学大学院　基幹理工学研究科　情報理工学専攻

1.本研究の背景と目的

インターネットの普及により，今日ではネットワーク経由でソフトウェアをダウンロードして利用する機会が増えている．これらのソフトウェアには便利なものも多いが，ウィルスやトロイの木馬のように危険なものも数多く存在している．

一方，システムのセキュリティを高めるため，セキュアOS[2]と呼ばれるオペレーティングシステムが登場している．これらのOSは厳密に定義されたセキュリティポリシによりアプリケーションに対して厳格なアクセス制御を行う．しかし，アプリケーションが安全に動作するようなポリシを定義するためには，アプリケーションの挙動を仔細に理解している必要がある．このため，ダウンロードしてきた挙動の不明なアプリケーションをセキュアOSの上で安全に動作させることは非常に難しい．

本研究の目的は，セキュアOS上へのアプリケーションダウンロードを想定し，アプリケーションに対して動的なアクセス制御を行うシステムを構築することである．

2. 設計

動的アクセス制御機構を設計するに当たり，以下の三つの点を要求事項として考慮した．

第一に，カーネルのアクセス制御機構に対する変更は極力避けるという点である．システムのアクセス制御は，セキュアOSによる強制アクセス制御を基礎としている．この基礎部分の信頼性を低下させないために．カーネルへの変更は必要最小限にとどめることが望ましい．

第二に，最小限のコード変更でアプリケーションが動作可能であるという点である．こうすることで，システム上で既存のアプリケーションを簡単に動作させることができる．また，アプリケーションの開発者は権限取得について意識することなくコーディングを行うことができる．

第三に，アプリケーションに権限を付与してよいかの判断基準をさまざまな方法で実装可能にするという点である．判断の基準としては，アプリケーションの信頼度，アクセスの対象と種類，ユーザの意思に基づく判断などさまざまな手法が考えられる．どのような方法が最適かは，システムの利用形態によりそれぞれ異なる．さまざまな環境に対応するために，実装の形式を柔軟に変更できることが必要である．

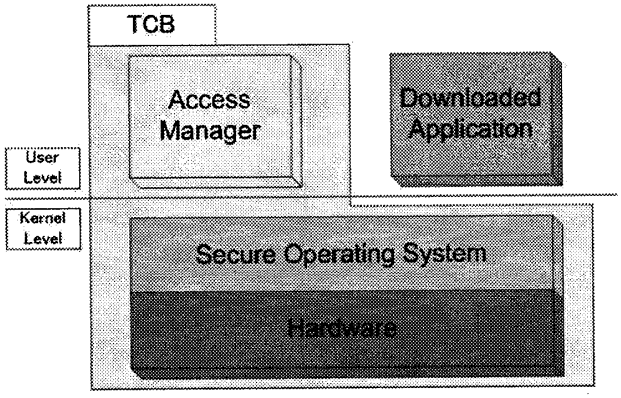

図2.1　システムアーキテクチャ

図2.1に示すのが，動的アクセス制御システムのシステムアーキテクチャである．

アクセス制御を執行する側の機構として，アクセスマネージャとセキュアOSが存在する．ダウンロードされたアプリケーションは，セキュアOSのセキュリティポリシによる束縛を受け，システムのリソースに直接アクセスすることができない．アプリケーションがリソースへのアクセスを必要とする時は，次の手順でアクセスマネージャからアクセス権を取得する．

1．アクセスマネージャにアクセス要求を行う．

2．アプリケーションのリソースに対するアクセスを認めるならば，アクセスマネージャは自身の権限に基づいてリソースへのアクセス権を取得する．

Implementing Dynamic Access Control System on Security-Focused Operating System
Tomohiro KATORI, Yuki KINEBUCHI, Wataru KANDA, Yu YUMURA and Tatsuo NAKAJIMA, Dept. of Computer Science, Graduate School of Fundamental Science and Engineering, Waseda University

3．アクセスマネージャが取得したリソースへのアクセス権を，アプリケーションに受け渡す．

これにより，アプリケーションからリソースへのアクセスが可能となる．

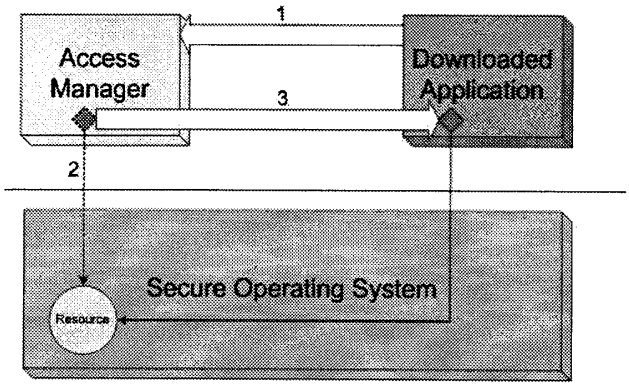

図2.2　アクセス権取得の流れ

アクセスマネージャはセキュア OS 上で動作するユーザレベルのプロセスである．このため，アプリケーションの要求を審査する方法として，GUI などのインターフェースを用いたユーザへの問い合わせや，ネットワークを介したセキュリティ情報の取得などを容易に実装することができる．また，アクセスマネージャもセキュア OS のポリシによる制限を受けるので，アクセスマネージャ自らに与えられないアクセス権をアプリケーションに付与することはできない．

3. 実装

今回の研究では，利用するセキュア OS の基礎として AppArmor[1]を用いた．AppArmor はパス名を使った単純なポリシ記述に基づくファイルアクセス制御を行うことができるが，ネットワークやシグナル送信などのリソースに関しては制御することができない．そこで，IP アドレスとポート番号に基づくアクセス制御機能，シグナル番号とプロセスが実行するファイルに基づくシグナル制御を実装した．

次に，セキュア OS の上で動作するアクセスマネージャを実装した．アクセスマネージャはプロセス間通信を経てアプリケーションからの権限要求を受け取り，要求したアプリケーションの名前や要求するファイルの名前，IP アドレス，アクセス権の種別などのパラメータを解析する．その解析したパラメータを予め登録した要求審査モジュールに渡し，審査結果を受け取る．システム開発者はこの要求審査モジュールを実装することで，GUI を用いた審査・ネットワーク経由での認証処理を経た審査など，システムの形態にあわせたアクセス要求審査を実現することができる．

さらに，アプリケーションからアクセスマネージャにアクセス要求を送信し，アクセス権を受け取るためのライブラリ関数を実装した．セキュア OS のアクセスコントロールにより，アプリケーションには必要最低限のアクセス権しか認められない．このため，システムコールなどを介してアクセス要求を行っても，アプリケーションが保持しないアクセス権を取得できないことが保証される．アプリケーションが予め与えられた権限を越えてアクセスを行うときは，このライブラリ関数を経てアクセスマネージャからアクセス権を取得する必要がある．

3. 実例

動的アクセスマネージャを利用したシステムの実例として，タッチパネルつきディスプレイ（ALGO SYSTEM SMART DISPLAY）上に組み込み機器を想定したシステムを構築した．このシステムでは，アプリケーションからのアクセス要求を受け取ったアクセスマネージャがユーザへの問いかけを行い，ユーザがタッチパネルで選択することでアクセス権を変更することができる．

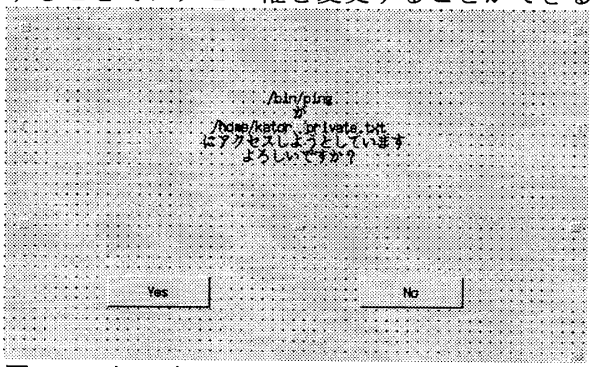

図2.3　タッチスクリーンによるアクセス権変更

4. まとめ

本研究では，セキュア OS とアクセスマネージャを組み合わせることにより，アプリケーションに動的にアクセス権を付与する動的アクセス制御の枠組みを提案し，実装した．実例として，ユーザへの問いかけを行うシステムを構築し，動的アクセス制御の有効性を確認した．

参考文献

[1] Leona Beatrice Campbell, Jana Jaeger. Nobell AppArmor 2.0 Administration Guide. Nobell Inc, 2006

[2] 内閣官房情報セキュリティセンター．　OS のセキュリティ機能等に関する調査研究．みずほ情報総研株式会社，　2005