モバイルエージェント・セキュリティ

3S－9

牧野 聡[1]　大越 匡[1]　徳田 英幸[1,2]
[1]慶應義塾大学 政策・メディア研究科 [2]慶應義塾大学 環境情報学部

1 概要

本論文では，まずモバイルエージェントの必要性を明らかにし，そしてモバイルエージェント・セキュリティの必要性とその具体例，現状での問題点について考察する．

2 モバイルエージェントの必要性

エージェント指向コンピューティングとモバイルエージェントの必要性を明らかにする．

2.1 エージェント指向コンピューティング

エージェント指向コンピューティングとは，オブジェクト指向コンピューティングの発展型としてとらえることができる[1]．ここでは，ソフトウェアに対する擬人化の第一段階として，属性とメソッド，そしてメッセージ駆動という概念が採用された．

そしてエージェント指向コンピューティングにおいては，擬人化の概念がさらに推し進められ，ユーザとのインタフェースやその挙動アルゴリズムについても人間の外見や思考に似たものとなっている．

オブジェクト指向コンピューティングがソフトウェアに対する擬人化の要求から生まれたとすれば，オブジェクト指向コンピューティングからエージェント指向コンピューティングへの進化は必然である．

2.2 モバイルエージェント

モバイルエージェントは，エージェント指向コンピューティングのもつ特質のうち移動性について特に着目したものであり，次世代の分散システムとして注目を集めている[2]．自律的に移動先ホストを選択した上でそこに移動し，移動先において処理を継続する計算主体として定義される．

モバイルエージェントを利用することにより，通信回数の低減，ネットワーク・プログラミングの隠蔽，移動先計算資源に対する直接アクセスなどの利点が実現される．

3 モバイルエージェント・セキュリティ概論

本論文で述べるモバイルエージェント・セキュリティ問題の概要について述べる．

3.1 問題意識

様々な特質を持つモバイルエージェントであるが，一般への普及は進んでいない．その最大の原因として，セキュリティの欠如が挙げられる．具体的には，

- ホストは受け入れたエージェントがどのような動作を行うか予測できない
- エージェントは移動先ホストにおいてどのような処理が行われるか予測できない

以上2点の問題が存在するため，ユーザは安心してエージェントを送り出したり，エージェントに対して計算資源を提供したりすることができない．

2.2項において述べたモバイルエージェントの利点を一般に広めるためにも，セキュリティの確保が最重要の課題であると考える．

3.2 対策の難しさ

モバイルエージェント・セキュリティ問題の解決が遅れている理由として，従来のセキュリティ対策における前提が通用しないという点が挙げられる．例えば，

- エージェントの位置は一定ではない
- 内部状態も一定ではない
- 責任の所在が分散する
  (ダウンロード元，所有者，移動元など)

モバイルエージェントは以上のような性質を持つため，従来のセキュリティ対策を適用することは困難である．

4 攻撃・防御策の具体例

モバイルエージェントに関するセキュリティアタックと，これに対する防御策の具体例について以下に述べる．

4.1 考えられる攻撃例

モバイルエージェントを使用する場合に考えられるセキュリティ・アタックを，対ホスト・対エージェントに分けて列挙する．

ホストに対する攻撃

- 機密データを盗み出す
- ユーザ権限を獲得する
- データを改竄する
- DoS (Denial of Service) 攻撃を行う

エージェントに対する攻撃

- 機密データを盗み出す
- IDデータを盗み出し，なりすましを行う
- データを改竄する
- 動作内容を改竄し，ウィルス化する
- 直接操作 (破棄・非活性化・送信など) する

4.2 防御策の具体例

各種の攻撃に対する防御策を，攻撃・被攻撃主体ごとに論じる．

一般的な対策

一般的防御策としては，エージェントの動作に際して認証を行うこと，転送経路及びホスト上においてエージェントデータに対して暗号化を行うこと，エージェントデータが改竄あるいは複製された場合にそれらを検出することなどが考えられている．

エージェントからの攻撃に対するホストの防御

この種の攻撃に対する防御策として，以下のものが考えられている．

- エージェントプログラム変換

Mobile-agent Security
Satoshi Makino[1], Tadashi Okoshi[1] and Hideyuki Tokuda[1,2]
[1]Graduate School of Media and Governance, Keio University
[2]Faculty of Environmental Information, Keio University

計算資源へのアクセス箇所を排除あるいは監視.

・ 機能制限付き仮想機械
資源へのアクセス能力のない仮想機械を用意.

・ 証明付きコード
処理内容を添付し，実際の処理と比較.

ホストからの攻撃に対するエージェントの防御

ホストからエージェントへの攻撃に関しては，認証機構によって信用されたホストにのみ移動して実行を行うという対策が提案されている．しかし，抜本的な方法はいまだに見つかっていない.

エージェント相互間の攻撃に対する防御

この問題に関しては，以下の対策が考えられる.

・ 実行空間の隔離
エージェントごとに異なる空間を用意.

・ 実体の隠蔽
何らかの中継媒体を通じてのみ相互間で通信.

5 主要プロダクトにおける防御策

現在公開されている各エージェントシステムのセキュリティ機構について述べる.

5.1 Voyager ORB

Voyager ORB[3]では，JDK1.1ベースのセキュリティ機構 (`SecurityManager`) が実装されている．しかし，セキュリティポリシの変更のためにはプログラムの再コンパイルを必要とするため，初心者には設定が難しい.

5.2 Aglets

Aglets[4]では，Java 2互換のポリシ記述ファイル (ACL) を用いたセキュリティ機構が用意されている．このファイルを書き換えるだけで，資源ごとの柔軟なアクセス制御が可能となっている．しかし，3.2項において述べたような理由から，エージェントのコードに対する署名についてはサポートされていない.

5.3 耐タンパ・移動エージェント

「対タンパ・移動エージェントの調査研究」プロジェクト[5]は，エージェントの計算結果に対する改竄の検出を目的としている．エージェントコードに対し複数の方法を用いてobfuscation (難読化) を行い，それぞれのエージェントが得た計算結果を持ち寄って多数決を行うことにより計算結果の信頼性を高めている．この方式の問題点としては，完全なobfuscationは不可能であるため依然としてエージェントは改竄の危険にさらされること，計算結果しかチェックされないためエージェントはホストに対して攻撃可能であることなどが挙げられる.

5.4 Ajanta

Ajanta[6]は，すべての計算資源に対してProxy (中継媒体) を付与している．エージェントは計算資源に対して直接ではなくProxyを経由してアクセスするため，安全性が実現されている．同時にJava言語における `ClassLoader`・`ThreadGroup` の両機構を用いて，サーバ上での各エージェントの実行空間に対する隔離を行っている.

6 考察

3.2項や5節から明らかになった問題点と，今後の課題について以下に述べる.

6.1 現状の問題点

問題点として以下の2点が挙げられる.

ホストからの攻撃に対する防御

まず，エージェントはホストからのアクセスや攻撃に対して無防備であるという点が挙げられる．この理由は，ホストはエージェントのコード・内部状態ともに自由にアクセス可能である (そうでなければエージェントを実行できない) ためである．また，ホストの機能を制限するAPIを設けても，悪意のあるホストはこれを守らないため無意味であるという理由も挙げられる.

エージェント間でのアクセスコントロール

エージェントに対するホストのアクセスコントロールはJava言語のセキュリティ機構を用いてすでに実現されている．また逆に，ホストに対するエージェントのアクセスコントロールは前項で述べたとおり原理的に不可能である．一方，エージェント相互間におけるアクセスコントロールは実現可能であるにもかかわらず，5節で取り上げたシステムではいずれも実装されていない．ホストの安全性を仮定できるような状況下では，このようなアクセスコントロールも必要であると考える.

6.2 今後の課題

おわりに，モバイルエージェント・セキュリティの実現に向けて考慮されるべき課題について考察する.

統合されたソリューション

対タンパ・移動エージェントでは，ホストに対する攻撃についてはあまり考慮されていない．またAjantaはエージェント間でのアクセスコントロール機能を持たないなど，各システムに一長一短があるといった状況である．これらのシステムはモバイルエージェント・セキュリティの部分問題を解決しているにすぎず，モバイルエージェントの普及のためにはこれらの解決を統合してユーザに提供することが望まれる.

シンプルな構成

一方，統合されたシステムの構成はできるだけシンプルなものであることが望ましい．なぜなら，複雑すぎるシステムはユーザに対して非親和的であり，モバイルエージェントの普及に対して妨げとなり得る．また，システムの構成要素に第三者の所有するホストが含まれる場合，そのホストをどのようにして信用・認証するかという問題点が派生する.

参考文献

[1] 本位田・大須賀，『オブジェクト指向からエージェント指向へ』，ソフトバンク，1998

[2] 佐藤他，『モーバイルエージェント』，日本ソフトウェア科学会チュートリアル，1999

[3] ObjectSpace Inc., *Voyager ORB*, <http://www.objectspace.com>.

[4] IBM Corp., *Aglets*, <http://www.trl.ibm.co.jp>.

[5] IPA，『対タンパ・移動エージェントの調査研究』，<http://www.ipa.go.jp>

[6] Karnik et al., Ajanta, <http://www.cs.umn.edu>.