3ZC-04

巡回エージェントによるサーバーのセキュリティ機能の向上*

星野 良将† 佐藤 弘章† 能登 正人†

神奈川大学 工学部‡

1 はじめに

現在に至るまでインターネットは便利性のみが前面に打ち出されてきて，そのユーザー数はうなぎ上りであったのに対し，インターネットの危険性(ネットワークセキュリティ)についてはそれほど注目されることはあまりなかった．そのため，最近のネットワークセキュリティは穴だらけであり，インターネットの知識を持っている人が操作すれば，簡単に企業などのネットワークに侵入が可能な状況を作り出してしまっている [6].

本稿では，エージェントを Web 上に巡回させ，サーバーのセキュリティチェックを行い，サーバーのセキュリティ機能を向上させる方法を提案する．

2 現在のサーバーセキュリティ

最近ではインターネットセキュリティへの関心がだいぶ高まってきている．実際に，一部の企業では，ファイアウォール(防御壁，ネットワーク監視)を導入するなど非常に高度なセキュリティを導入をする傾向が見られる半面，全く無関心，無知な人が多いのもまた事実である．クラッカー達はセキュリティの甘いサーバーを足掛かりにして，より高度なセキュリティを持つサーバーに攻撃を仕掛ける．これは，そうする事によって不正アクセスが検知されても足掛かりにしたサーバーまでしかたどり着けず足がつかなくなるからである．それ故，一部の過剰ともとれるセキュリティの導入よりも，インターネットの危険性を知り，ネットワーク全体でのセキュリティ意識と機能の向上が必要である [7].

*Improvement of Security for Server using Patrol Agents

†Yoshimasa HOSHINO, Hiroaki SATO and Masato NOTO

‡Faculty of Engineering, Kanagawa University

3 巡回エージェント

次のような性質を持つシステムをエージェントと呼ぶことが，コンセンサスとなりつつある．

1. 自律性：プログラム自身が自己完結していて，他からのアクションや刺激などが無くても実行を維持・継続させている状態．
2. 協調性：エージェント同士が通信し合う機能を持つという事である．(社会性などという言い方をしている人もいる．)
3. 反応性：外界の事象が発生した場合，それに反応して動作が可能である．
4. 自主性：エージェント自身に，何らかの解決すべき問題が与えられている場合，他から強制されなくてもその問題の解決に向うように，自分自身のプログラムを動作させることを言う．

このような性質を持ちなおかつ WEB 上を自由に巡回するシステムを本稿では巡回エージェントと呼ぶ [8].

4 サーバーチェック

4.1 使用するエージェント

本研究では目的に応じた 2 つのエージェントを使用する [1, 2, 3, 4]. エージェントの簡単な動きを図 1 に示す．

1. チェックエージェント：Web 上を巡回し，サーバーの探索およびセキュリティチェックを行う．チェック項目に対して何らかの対策がたててある場合，IP アドレスを記憶し情報取得エージェントに伝達する．

2. 情報取得エージェント：チェックエージェントによりセキュリティが良しとされたサーバーのIPアドレスを譲り受け，サーバーに赴きソースコードからセキュリティ情報を取得する．

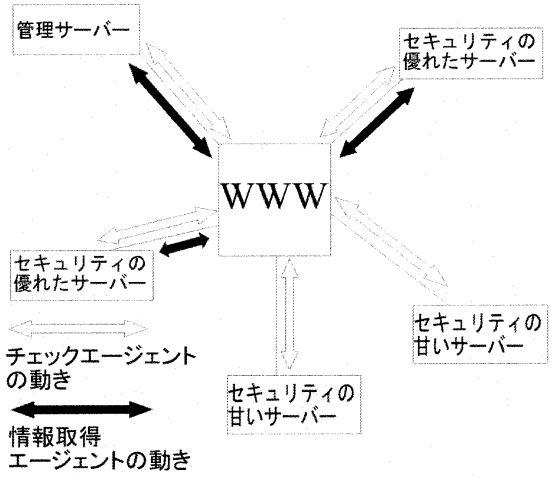

図 1: 巡回エージェントの動き

4.2 チェック項目

サーバーのセキュリティチェックは参考文献 [7] にあるものを参考にし，必要であると思われる項目をまとめた．

1. インターネットに接続されているコンピュータで危険なサービスが動いているかどうかの調査
2. 実際に攻撃しての調査
3. 管理者権限の奪取が行えるか否かのチェック
4. すでに攻撃されているか否かのチェック
5. サービス停止攻撃が行えるか否かのチェック
6. 学内，社内の情報が奪えるか否かのチェック
7. ネットワークの盗聴が行えるかチェック

4.3 サーバーチェックの流れ

1. サーバーの種類をチェックする．(NT，BSD，Java，Linux 等)
2. 管理しているサーバーと同じ種類のサーバーであったらチェックを開始する．サーバーのチェックは上記の項目で行う．
3. セキュリティ状況を調べて，管理するサーバーにないセキュリティコードまたはバージョンアップ等の改良されたセキュリティコードを発見したらIPアドレスを取得し，情報取得エージェントに送る．
4. 情報取得エージェントは送られて来たIPアドレスに赴き，ソースからセキュリティコードを取得し，管理するサーバーに持ち帰る。
5. 持ち帰られたセキュリティコードを管理するサーバーのソースに書き込み，新たに取得したセキュリティをサーバーに組み込む．
6. 書き込みが終了したら再び同じ作業に戻る。

5 おわりに

本稿では，サーバーのチェックからセキュリティのソースコードの取得までを行った．しかし，セキュリティに対して評価が行われていないため最適なセキュリティが得られるとは言い難い．今後はエージェントに評価の概念を持たせ最適と思われるセキュリティを更新出来るようにする必要がある。

参考文献

[1] 岩井俊弥：JAVA モバイルエージェント，SRC(1998).

[2] ジョセフ・オニール：独習JAVA，翔泳社(1999).

[3] David Flanagan：JAVAプログラムクイックリファレンス，オーム社(1998).

[4] Robert Lafore：Javaで学ぶアルゴリズムとデータ構造，SOFTBANK(1999).

[5] 谷口 功：通信プロトコルしくみ，日本実業出版者(1998).

[6] http://softplaza.biglobe.ne.jp/text/security/

[7] http://www.netagent.nd.to/index.html

[8] http://www.isdnet.co.jp/product/