サーバの動作監視によるサイバー攻撃防御システム

矢野尾　一男　　中江　政行　　小川　隆一
NEC インターネットシステム研究所†

1. はじめに

近年のインターネットの普及に伴い、インターネットは重要な社会インフラとなりつつある。その一方で、サーバシステムの脆弱性（ソフトウェアの障害・設定ミス等）を悪用する、いわゆるサイバー攻撃がセキュリティ上の大きな問題となっている。

サイバー攻撃を検知する目的で不正侵入検知システム(IDS)の導入が進んでいるが、顧客リストの流出のような深刻な被害に対処するには、攻撃を検知するのみでは不十分であり、これを未然に防止できる必要がある。

本稿では、サーバの動作を監視することによって、脆弱性に対する攻撃を未然に防ぐサイバー攻撃防御システムを提案し、その監視方法と監視ポリシー作成方法について報告する。

2. 課題

脆弱性に対する攻撃は、理想的にはサーバの脆弱性自体を無くすことによって未然に防ぐことができるが、現実にはソフトウェアの障害を完全に無くすことは困難である。また、サーバソフトウェア作成者とサーバ運用者は一般に同一ではないので、サーバ運用者がサーバソフトウェアの仕様を誤解して不適切な設定をしてしまう可能性は排除できない。

したがって、攻撃を防御するためには、サーバソフトウェアとは独立して、サーバ運用者が期待しているとおりにサーバが動作しているかどうかを監視し、違反動作を阻止する仕組みが必要となる。

これは、OS のアクセス制御機能によりある程度実現でき、SELinux 等アクセス制御機能を大幅に拡張した OS やミドルウェアが提案されている。

しかし、従来のアクセス制御技術では、「ファイルを書き出す」「ファイルを実行する」といった個々の違反動作は阻止できるが、「書き出したファイルを実行する」といった一連の動作に関連する違反動作は阻止できない。ソフトウェアの動作監視のためには、このようなソフトウェアの内部状態に依存したアクセス制御が必要であると考えられる。

3. 状態依存アクセス制御

前節の課題に対処するため、以下に述べる状態依存アクセス制御モデルを提案する。

状態依存アクセス制御のルールを、$\pm(s,o,a,p)$ と表す。これは、アクセス制御の主体 s の対象 o に対する操作 a の可否(±)が、前提条件 p によって規定されることを意味する。p は、操作の履歴パターンを正規表現で記述したものであり、操作 a が実行されるまでの s から o への操作の履歴が、これに合致する場合に限り、操作 a に対してアクセス制御 $\pm(s,o,a,p)$ が適用される。例えば、

$+(s,o,a_1,a_2a_3)$

は、s から o に対する操作 a_1 が、その直前に a_2 と a_3 が連続して実行されている場合のみ許可されることを表す。

$-(s,o,a_2,a_4.*|a_1)$

は、s から o に対する操作 a_2 が、それ以前に a_4 が実行されているか、その直前に a_1 が実行されている場合のみ拒否されることを表す。

上記のアクセス制御モデルは、以下の実行系により実現できる。まず、それぞれの対象 o ごとに、許可される操作のみ状態遷移が可能な有限状態遷移機械(FSM)を生成する。例えば、上記の２つのルールからは図 1 のような FSM が生成される(a は a_1, a_2, a_3, a_4 以外の全ての操作を示す)。実行時は、アクセス(s', o', a')を受けるごとに、o' に対応する FSM に a' を入力し、状態遷移不可の場合に限りその操作を拒否する。

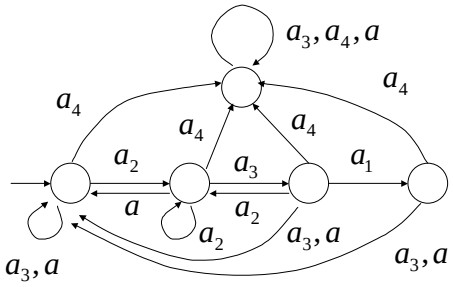

図 1　FSM の例

An Intrusion Prevention System based on Server Monitoring
† Kazuo YANOO, Masayuki NAKAE, Ryuichi OGAWA, Internet System Research Laboratories, NEC

4. 提案システム

状態依存アクセス制御のルールを記述することは、通常のアクセス制御のルールを記述する場合よりも一層プログラムの動作に関する知識が必要となり、サーバ運用者が直接これを記述することは困難であると考えられる。

そこで、状態依存アクセス制御を実現する監視ミドルウェアと、アクセス制御のルール（監視ルールと呼ぶ）を生成する監視ルール生成ツールから成るサイバー攻撃防御システムを提案する（図 2）。

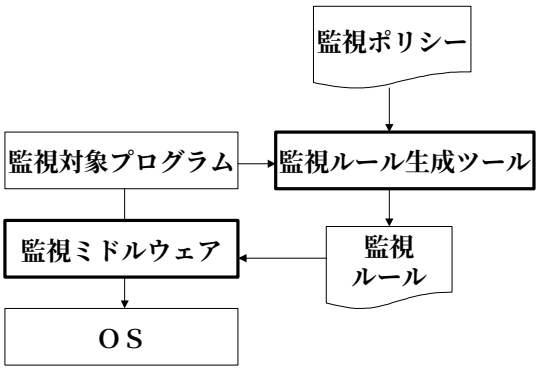

図 2　システム構成

4.1. 監視ミドルウェア

監視ミドルウェアは、監視対象プログラムから OS へのシステムコールの呼び出し可否を状態依存アクセス制御により決定する。

アクセス制御の対象として、ファイル名やＩＰアドレスを指定することにより、特定リソースに対する状態依存アクセス制御を記述できる。また、対象を指定せずに、システムコールの呼び出しそのものの可否を指定することもできる。

4.2. 監視ルール生成ツール

本システムでは、サーバ運用者は、サーバに期待する動作を監視ポリシーとして指定する。監視ポリシーは、「ファイルを書き出した後、それを実行してはならない」といった抽象度の高い時相論理で記述される。

監視ルール生成ツールは、システムの詳細に基づいて監視ポリシーを監視ルールにコンパイルすると同時に、監視対象プログラムを解析して、監視ポリシーによって禁止されたアクセスが生じる可能性がないか整合性検査をする。その可能性がある場合は、サーバ運用者に監視ポリシーの修正案を提示する（図 3）。

この機能は、厳格すぎる監視ポリシーを設定することによって生じる誤検知(False Positive)を防ぐためのものである。静的チェックの能力は限られるが、試験運用では発見しにくい例外的処理に由来する誤検知の排除に効果がある。

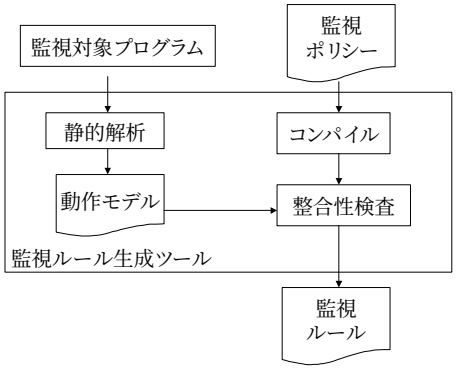

図 3　監視ルール生成ツール

5. 先行研究との比較

提案システムに類似した研究として、仕様ベース IDS が挙げられる。状態依存アクセス制御は、[1]で提案されているセキュリティオートマトンを拡張したものである。本提案の拡張により、脆弱性に対する攻撃を防御するために十分な記述力が得られると考える。[2]は本提案よりも複雑なアクセス制御の記述が可能であるが、実行系の実行性能の低下と、ルールの記述ミスの発見が困難である点で問題がある。

また、[3]は本提案と同様に、監視ルールを生成する仕組みを持つが、本提案では、誤検知を低減させるための整合性検査機能によって、運用性を向上させている。

6. まとめ

本稿では、監視ポリシーに基づきソフトウェアの脆弱性を突く攻撃を防御するシステムを提案した。

本方式は、CGI 等のためサイトごとに監視ポリシーが大きく異なる Web サーバシステムの保護に特に向いていると考えられる[4]。今後は、Web サーバを対象として、実行性能と防御性能を評価していく予定である。

参考文献

[1] F. Schneider, Enforceable Security Policy, ACM Transactions on Information and System Security, Vol.3, Issue 1, Feb. 2000

[2] R. Sekar 他, Synthesizing Fast Intrusion Prevention/Detection Systems from High-Level Specification, 8th USENIX Security Symposium, 1999

[3] D. Evans 他, Flexible policy-directed code safety, IEEE Security and Privacy 1999

[4] 中江他, 動作監視に基づく Web サーバ防御システム, 情処研究会報告 CSEC-19-3, 2002