プログラムの動作解析によるアクセスポリシー生成方式

矢野尾　一男　　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、サーバの脆弱性（バグ・設定ミス等）を悪用して不正なコードを実行させる攻撃が情報セキュリティ上の大きな脅威となっている。

この問題は、サーバに対して適切なアクセス制御をかけて、動作を制限することによって対処することができるが、不正な動作のみ制限するアクセスポリシーを作成することが困難であるという課題がある。

本稿では、プログラムからのリソースアクセスの呼び出し位置を利用して、アクセスポリシーを生成する方式を提案し、その評価結果について報告する。

2. 背景と課題

最近、プログラム毎にファイル等リソースへのアクセスを制限できる OS やミドルウェアが実用化されており、適切なアクセスポリシー（リソースへのアクセスの可否を表すポリシー）を設定すれば、ソフトウェアの脆弱性を原因とした不正アクセスの多くを防止することができる。

しかし、正しいアクセスポリシーを作成するためには、プログラムが実行時にアクセスするリソースを全て知る必要があり、人手によってこれを正しく記述するのは困難である[1][2]。

そこで、不正アクセスを受けない状態でプログラムのリソースアクセスを一定期間観測し、アクセスポリシーを自動生成する方法が提案されている[2]。

しかし、プログラムのリソースアクセスを一定期間観測しても、一般に、そのプログラムが行ない得るすべてのリソースアクセスを観測することは困難である。したがって、観測したリソースアクセスを汎化して、より一般的なアクセスポリシーを生成する必要がある。

例えば、/tmp/tmp123 と/tmp/tmp456 へのファイル書き込みが観測された場合、テンポラリファイル書き込みと考えられるので、/tmp/tmp*へのファイル書き込みを許可するアクセスポリシーを生成するのが望ましい。一方、/etc/passwd と /etc/host のファイル読み込みが観測された場合、/etc/*のファイル読み込みを許可するのは必ずしも正しくない。なぜならば、テンポラリファイルの場合と異なり、この２つのファイル読み込みは全く別の意味を持つ場合が多く、その場合は汎化すべきではないからである。

3. 提案方式

上述のとおり、汎化処理を単にアクセス先のファイル名のパターンのみから行うのは限界がある。そこで、筆者らは、プログラム中の同じ場所から呼び出されたリソースアクセスは、同じ目的を持つことが多いという性質を利用したアクセスポリシー生成方式を提案する。

本方式では、プログラムのリソースアクセスを一定期間観測し、観測されたリソースアクセスがプログラム中のどの位置から呼び出されたかに基づいてグループ化し、それぞれのグループ内で汎化処理を行う。そして、その結果得られるリソースアクセスを許可し、それ以外を禁止するアクセスポリシーを生成する(図 1)。

このような、プログラム共通の性質を用いることにより、「/tmp 以下にはテンポラリファイルが書き出される」といった ad hoc な知識に依存しない汎用性の高い汎化処理を実現できる。

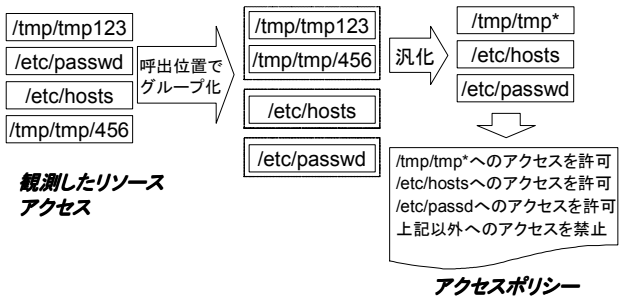

図 1 提案するポリシー生成方式

4. 試作ツールの概要

Linux OS 上で、提案方式を実装したアクセスポリシー生成ツールを試作した(図 2)。

アクセスポリシー生成の対象となるプログラムは、アクセス制御モジュールのデバッグ子プロセスとして起動され、プログラムから OS へのシステムコールは ptrace システムコールを利用して毎回トラップされる。

アクセス収集モジュールは、システムコール呼び出し時のレジスタとスタックの内容から呼

A Method for Generating Access Policies by Dynamic Program Analysis
† Kazuo YANOO, Ryuichi OGAWA,
Internet Systems Research Laboratories, NEC

出位置を特定し、その呼出位置と、システムコールの種類・引数とをアクセス履歴に記憶する。

ここで、システムコール呼び出し時のプログラムカウンタの値のみでは、共有ライブラリ経由でシステムコールが呼び出された場合、同じ位置から呼び出されたものと見なされてしまう。そこで、スタックの内容を調べ、同じコールスタックを持つシステムコール呼び出しのみ、同じ位置から呼び出されたものと見なす。

ポリシー生成モジュールは、収集されたアクセス履歴を読み込み、呼出位置によってグループ化し、それぞれのグループの中で汎化する。

対象とするシステムコールは、ファイル入出力に関わるものに限定し、汎化処理は、アクセス先のファイルパスを一旦 '/'と'.'を区切り文字としてトークン化し、そのトークンの最大共通部分文字列を算出する方法を用いた。

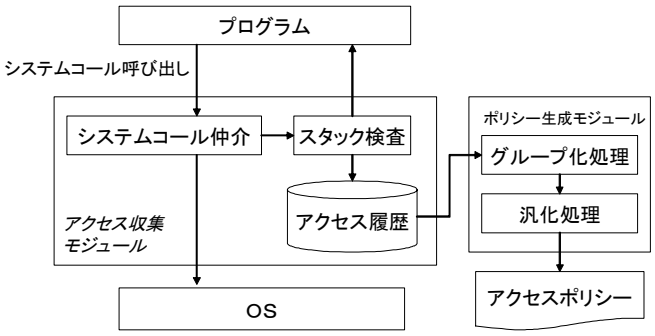

図 2　システム構成図

5. 性能評価

実際に運用されている Apache サーバを対象にして、試作ツールの評価実験を行った。

運用中に収集したアクセスログから、連続した 100 個のリクエストを無作為に抽出して Apache に送信し、open システムコールのアクセスポリシーを生成した。

また、本方式とファイル名のパターンを用いる手法とを比較するため、呼出位置の情報は用いずに、拡張子によってグループ化して汎化する方式でもアクセスポリシーを生成した。Linux では拡張子でファイル種別を表すことが多いため、妥当なグループ化処理と考えられる。

両方式で生成したアクセスポリシーの一部（open が許可されるファイル）を表 1 に示す。

次に、アクセスログに含まれる 10000 個のリクエストを Apache に送信して open システムコールを全て収集し、両方式で生成されたアクセスポリシーの精度を評価した。評価指標として、誤検知率と適合率の 2 つを算出した。

誤検知率は、実際に生じたアクセスの中で、アクセスポリシーで禁止されているものの割合を表す。この数値は、システムの誤動作を引き起こす過剰防御(False Positive)の発生確率に相当する。

適合率は、アクセスポリシーで許可されているアクセスの中で、実際に生じたものの割合を表す。この数値が小さいほど、検知もれ(False Negative)の可能性が高くなる。

適合率が両者とも低いが、その理由として、10000 個のリクエストでは Apache の全てのファイルアクセスを発生しきれていないこと等が挙げられる。

一方、誤検知率の評価結果は本方式の結果が明らかに優れている。検知もれを抑えながら、誤検知も減らせているという点で、より精度の高いアクセスポリシーを生成できていることが分かる。

表 1 生成されたポリシーの一部

本方式	拡張子方式
/etc/hosts	/etc/hosts
/etc/resolv.conf	/*.conf
/www/conf/httpd.conf	/*lib/*.so.*
/www/lib/*	/www/htdocs/*.html
/www/htdocs/*	/www/htdocs/*/*.gif

表 2　各方式の評価結果

	本方式	拡張子方式
誤検知率	1/268 (0.4%)	17/268(6.3%)
適合率	227/1174 (19.3%)	212/1189(17.8%)

6. まとめ

本稿では、リソースアクセスの呼び出し位置を利用して、アクセスポリシーを生成する方式を提案した。そして、評価実験により、呼び出し位置の情報のみからでも、リソースアクセスを有効に分類でき、精度の高いアクセスポリシーを生成できることを確認した。

今後は、呼び出し位置以外の情報と組みあわせた、さらに精度の高いポリシー生成方式を開発する予定である。

参考文献

[1] 矢野尾,中江,小川, "サーバの動作監視によるサイバー攻撃防御システム", 第 65 回情報処理学会全国大会, 2003

[2] N. Provos "Improving Host Security with System Call Policies", 12th USENIX Security Symposium, Aug. 2003