M-087

ふるまいに基づくサーバ攻撃防御方式
A Behavior-based Process Confinement Method for Server Applications

中江 政行* 小川 隆一*
Masayuki Nakae Ryuichi Ogawa

1. はじめに

CGI を用いた Web アプリケーションなど，一般的なサーバアプリケーション（サーバ AP）では，複数プロセスの連携動作によりサービスが提供される．このようなサーバ AP において，マスタープロセスだけでなく，関連する任意プロセスの脆弱性を突いた不正侵入が脅威となる．こうした脅威に対抗するには，関係する全てのプロセスを保護する必要がある．

そのために，従来からユーザプロセスのシステムコール発行を監視する方式が提案されているが[1]，サービスの公開先やプロセス実行順序などに関するサーバ AP に依存した仕様（以下 AP 仕様）への対応が不十分であり，誤検知なく防御することが困難であった．そこで，サーバ AP によるシステムコール発行履歴を「AP のふるまい」として用いる攻撃防御方式を提案する．本方式は，サーバ AP の正常動作を表す文脈依存ポリシー（CSP）と，アクセス要求ごとのシステムコール発行履歴とを照合して，不正動作をリアルタイムに検知・遮断する．CSP はプロセスごとのシステムコール発行可否だけでなく，アクセス元ドメインとファイルアクセスとの関係などを指定することで，AP 仕様を詳細に記述できる．本方式により，複雑な AP 仕様をもつサーバ AP に対して，より高い防御精度を実現できる．

本稿では CSP とサーバ AP のふるまいモデル（ビヘイビアツリー）の定義，ビヘイビアツリーを用いた CSP 照合アルゴリズムを示す．また，本方式に基づく攻撃防御ミドルウェア「S-Tracer」の実装と評価について述べる．

2. サーバ攻撃防御の要件

2. 1. サーバ AP の動作

サーバ AP は一般に複数のサービスを外部ネットワークに提供するよう構成されており，クライアントからのアクセス要求に応じて，(1)サービスが選択され，(2)サービスごとに定められた手続きを行い，(3)クライアントに応答を返す．

まず(1)について，典型的なインターネットサーバでは，一般公開されるサービスのほか，保守用の特権サービスが用意されており，そのアクセス要求時には，アクセス元 IP アドレスが所定の管理ドメイン内にあるか否かにより，その提供可否がチェックされる．

次に(2)について，例えば CGI による Web アプリケーションでは，選択されたサービスに応じた CGI プロセスが新たに起動され，様々なネットワーク I/O やファイル I/O が実行される．さらにシェルなどの外部コマンド呼び出しにより，副次的なプロセス生成が行われることも多い．

2.2. 脅威

前述のような動作を行うサーバ AP にとって，以下のような攻撃が具体的な脅威として想定される．

- マスタープロセスの脆弱性により，特権チェックがバイパスされて，管理ドメイン外から特権サービスが起動される．
- 一般公開サービスを構成する任意のサブプロセスの脆弱性により，特権サービスを構成するプロセスの起動や，特権データへの不正アクセスが可能となる．

2.3. 防御の要件

これらの攻撃による被害を未然防止するためには，以下の 5 つの要件が必要と考える．

(a) 正常動作知識：AP 仕様に基づいて，サーバ AP 動作の正当性を判定できること．
(b) 識別可能性：不正な特権サービス起動などを検知するために，任意のプロセス動作について，それを引き起こしたアクセス要求を識別できること．
(c) 追跡可能性：プロセス実行過程を追跡して，不正なプロセス起動やデータアクセスを検知できること．
(d) 網羅性：任意のプロセスの動作監視が行えること．
(e) パフォーマンス：平常時の可用性を損なわないこと．

3. 提案方式

3.1. 文脈依存ポリシー（CSP）

CSP はサーバ AP の正常動作を表すための記述モデルであり，以下 3 要素の組 (IC, SC, PB) の集合と定義する．

- アイデンティティコンテキスト**(IC)**：アクセス要求元 IP アドレスの集合．
- サービスコンテキスト**(SC)**：特定のサービスを提供するプロセスの実行系列．
- プロセス動作**(PB)**：許可されるシステムコールイベントの集合．

それぞれの要素は，IP アドレスのビット列，プロセス名系列，システムコール名とパラメータ系列に関する正規表現として指定される．

CSP は，アクセスが許可されるネットワークドメインとサービスとの関係，サービスとネットワーク I/O・ファイル I/O 動作との関係を用いて AP 仕様を記述することができるため，前述の要件(a)を満たす．

3.2. ビヘイビアツリー

一般にサーバ AP は同時に複数のアクセス要求に対して，並行してサービス提供処理を行うが，前述の要件(b)・(c)を満たすためには，個々のアクセス要求に関連するイベント系列を選択的に抽出する必要がある．

そのために，イベント履歴の木構造表現であるビヘイビアツリーを設計した．ビヘイビアツリーは，プロセス生成イベント（fork, execve など）に相当するプロセスノードと，I/O 動作イベント（accept, read, write など）に相当する I/O ノードから成る順序木であり，以下の性質を満たすよう，イベントが観測されるたびに更新される（図 1 破線部参照）．このとき，システムコール名やパラメータ

*NEC インターネットシステム研究所
*NEC Internet Systems Res. Labs., NEC Corp.

なども属性として保持されるものとする．

(性質 1) プロセスノードのみ子ノードをもつ．

(性質 2) 任意のあるプロセス P が発生させたシステムコールイベント e は，P に相当するプロセスノードの子ノードとして左から発生順に並べられる．

サーバ AP のマスタープロセスは，複数のアクセス要求を受け付ける場合でも，アクセス要求 r を受信した後，直ちにサービス提供用サブプロセス π(r)を生成することから，さらに以下のような位相的性質が導かれる（図 2）．

(性質 3) r に起因して発生する任意のイベントに相当するノードは，π(r)に相当するプロセスノードを根とする部分木 T(r)を構成する．

(性質 4) r の受信に関する I/O ノードと T(r)とは，マスタープロセスの子として常に隣り合う．

3.3. CSP 照合方式

(1) イベント系列の抽出: 新たなイベント(カレントイベント)が観測される度に，ビヘイビアツリーを図 1 太線部のような順序で探索して，受信イベントに相当する I/O ノード accept(r)と，プロセスノード πi の系列 σ = (π0, π1, π2, ...)を求める．上記性質 3・4 より，カレントイベントを引き起こしたアクセス要求 r について，accept(r)が r の受信イベントに相当し，σ が r に関連するプロセス実行系列に相当するので，2.3 節の要件(b)・(c)を満たすことができる．

(2) 照合手続き: 抽出されたイベント系列と，CSP の各要素 (IC, SC, PB)について，(i) IC と accept(r)のソース IP アドレス属性，(ii) SC と各 πi ∈ σ のプロセス名属性の系列とが照合される．いずれにもマッチした場合，PB が現在の AP 状態における正常動作として選択され，カレントイベントと照合される．カレントイベントが PB にマッチしない場合，または抽出された系列がいかなる(IC, SC)にもマッチしない場合，カレントイベントを異常動作とみなす．

(3) 計算量: イベント系列の抽出にかかる計算量は木の深さに比例するので，木のサイズ N に対して O(log N)となる．イベント系列のサイズも木の深さに比例するので，その照合にかかる計算量は O(log N)である．よって CSP 照合全体の計算量は O(log N)である．木のサイズ N はサーバ AP 負荷にほぼ比例することから，高負荷状態においても効率よく照合処理をおこなえることがわかる．

4. 実装と評価

4.1. 実装

3 章で述べた方式に基づく攻撃防御ミドルウェア「S-Tracer」を実装した．S-Tracer は Linux Security Module (LSM)に準拠したカーネルモジュールであり，サーバ AP に一切手を加えることなく，ユーザ空間内の任意プロセスによるシステムコール発行を監視・追跡することで，2.3 節に示した要件(d)を満たすようにした．

所定の CSP に反するシステムコール発行が観測された際には，その実行前にブロックすることで被害の未然防止を行うことができる．

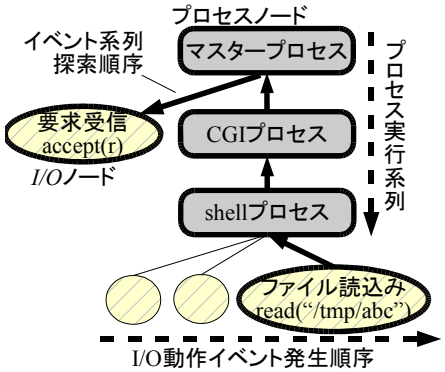

図 *1* ビヘイビアツリー

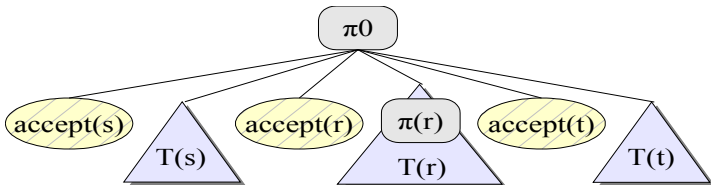

図 *2* ビヘイビアツリーの位相的性質

4.2. 防御性能評価

S-Tracer の防御性能を評価するために，入力未チェック脆弱性をもつ 2 つの Perl CGI から成る実験用 WebAP を構築した．一方の CGI は外部 NW に公開された個人情報登録サービスを，他方の CGI は特定管理ドメインから個人情報を参照するための保守サービスを想定している．

それぞれの CGI に対して外部 NW 側から，(1)パスワードファイル（/etc/passwd）の不正参照・改ざん，(2)個人情報ファイルの不正参照・改ざん，(3)不正な通信を行うプログラムの設置・起動といった攻撃実験を行い，すべて未然に防止できることを確認した．

4.3. パフォーマンス評価

S-Tracer のパフォーマンス評価として，以下のようなサーバに対して負荷測定実験を行った．

- HW: Pentium4-2GHz, 1GB メモリ
- OS: Linux 2.4.18 (lsm-patch-3)
- サーバ AP: Apache-1.3.27

クライアントからは自作ツールを用いて，実運用されている Web サーバのアクセスログを基に，以下のような HTTP トラフィックを発生させた．

- 実験時間・総アクセス回数: 24 時間・4,350,932 回.
- アクセス頻度(回/分): 平均 3021.5, 最大 6943, 最小 299.
- 平均応答バイト数: 4266 バイト

その結果，S-Tracer の非組込時と組込時における平均 CPU 負荷がそれぞれ 2.77%, 3.30%であり，0.5%の負荷増大に抑えられることを確認した．また，平均スループットはいずれも 209.79KB/秒であり，サーバ性能が低下しないことを確認した．これらの結果から，2.3 節に挙げた要件(e)を満たすことを確認できた．

5. さいごに

サーバ脆弱性を突く攻撃の防御方式として，サーバ AP の正常動作記述である CSP と AP のふるまいモデルであるビヘイビアツリーの設計，およびイベント系列と CSP の効率的な照合方式について概略を述べた．

また，本方式に基づく攻撃防御ミドルウェアである S-Tracer の実装について述べ，評価実験を通じて，実用的な防御性能・パフォーマンスをもつことを確かめた．

参考文献

[1] Chari, S. et al., "BlueBoX: A Policy-driven, Host-Based Intrusion Detection System", ACM TISSEC, v.6, n.2, 2003.