TOMOYO Linuxの2つの実装方式の性能評価

武田　健太郎[†]　半田　哲夫[††]　原田　季栄[†]

1. TOMOYO Linuxとは

筆者らはセキュアOSの1種であるTOMOYO Linux[1)2)]を開発し，GPLのオープンソースソフトウェアとして公開している．セキュアOSはセキュリティ機能を強化したOSであり，Linux向けのセキュアOSには，RedHat系ディストリビューションで採用されているSELinux[3)4)]，SuSE LinuxやUbuntuに採用されているAppArmor[5)]などが存在する．

著名なセキュアOSの中で，TOMOYO Linuxは使いやすいことで知られている．その最大の特徴は，セキュアOSのアクセス制御の設定であるポリシーをプロセスの挙動から学習できる点にある．TOMOYO Linuxを学習モードに設定した状態では，プロセスがアクセスした資源（ファイル，ネットワーク，シグナルなど）をカーネルが監視し，その挙動を自動的にポリシーに追加していく．この自動学習機能を用いることで，システム管理者は容易にシステムの挙動を把握し，保護したいアプリケーションのポリシーを簡単に作成することができる．

いくつかのセキュアOSについて，ファイルに対するアクセス制御機能の性能測定結果が文献[6)]に示されている．しかし，セキュアOSごとにセキュリティ強化の考え方と実装方式が異なることや，設定によって性能が大きく変化することから，セキュアOSの性能を正確に評価，考察することは難しい．

本発表では，TOMOYO Linuxの2つの実装方式を対象としてアクセス制御機能の性能について評価し，実装まで踏み込んだ考察を行う．また，LiveCDによるデモを実施し，TOMOYO Linuxがセキュリティ強化だけでなく，システムの挙動を理解する助けとなることを示す．

2. TOMOYO Linuxの2つの実装方式

TOMOYO Linuxには実装方式の異なる2種類が存在し，1.x系統と2.x系統というバージョン体系で区別されている．2.x系統は，Linuxの2.6カーネルに標準で搭載されているセキュリティフレームワークであるLSMを利用して実装されている．1.x系統はLSMを使用しておらず，種々のシステムコールに独自のアクセス許可チェック関数の呼び出しを追加することで，セキュアOSとしての機能を実現している．

LSMの実体は各種システムコールに挿入されているフックで，フック関数から呼び出される処理だけを実装することでセキュアOSを実現できるようになっている．このため，LSMを利用すれば日々変化するLinuxカーネル本体の変更に追従するのが容易である．逆に1.x系統のように独自にフックを挿入すると，カーネルの変更への追従コストは大きくなるが，設計の自由度は高く機能的に充実したセキュアOSを実現できるという特徴がある．TOMOYO Linuxの場合，2.x系統は機能面では1.x系統のサブセットとなっている．

3. TOMOYO Linuxの性能評価

独自フックを挿入しているTOMOYO Linux 1.5.1とLSMを利用しているTOMOYO Linux 2.1.0を評価対象として，Linuxカーネルのシステムコールレベルでの性能を測定できるLMBenchを用いて性能評価を行った結果を図1に示す．

セキュアOSの主たる機能であるファイルに対するアクセス制御機能に着目すると，TOMOYO Linuxはファイルのオープン操作に100%程度の特に大きなオーバヘッドが生じている．また，ファイルの作成と削除については，ファイルサイズが大きいほどTOMOYO Linuxのオーバヘッドは小さい．プロセス関連のベンチマーク項目では，プロセスの実行を伴う項目（fork+execve, fork+/bin/sh -c）で，ファイル操作ほどではないものの，5%程度のオーバヘッドが生じている．

これらのオーバヘッドは，TOMOYO Linuxのアクセス許可のチェック時に行われる文字列比較処理によるものである．

† 株式会社NTTデータ
†† NTTデータ先端技術株式会社

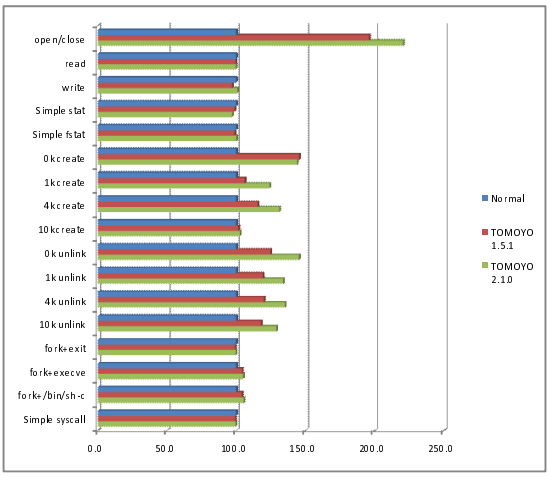

図 1 LMBench による性能測定結果

ファイルのオープン時には，パス名の比較でパターンマッチングを含む文字列比較が伴うため，大きなオーバヘッドが発生している．同じくパス名の比較が行われるファイルの作成，削除に関しても同様であるが，サイズの大きいファイルの作成，削除の方がオーバヘッドが小さい傾向にあるのは，TOMOYO Linux のアクセス制御の考え方に起因している．

TOMOYO Linux は，プロセスが資源に対してアクセスする「入口」を制限する，という考え方でアクセス制御を実装している．ファイルアクセスに関していえば，TOMOYO Linux はファイルのオープン時にのみチェックを行い，ファイルの内容の読み書きはチェックしないため，ベンチマーク項目の read/write ではオーバヘッドが生じていない．このため，作成・削除を行うファイルサイズが大きいほうが，ファイルのオープンにかかる時間の占める割合が小さくなり，オーバヘッドが小さくなる．セキュア OS の中にはファイルのオープン後にポリシーが変化した場合に対応するために読み書き時にもチェックを行うものも存在するが，TOMOYO Linux では「1 バイトでも読み書きを認めてしまった後にポリシーが変化したところで既に手遅れである」という考えからチェックを行わない．

プロセス関連のオーバヘッドもファイル操作と同様，実行許可のチェックとドメイン遷移処理のために文字列比較が発生するためである．ただし，もともとプログラムの実行は時間のかかる処理であるため，TOMOYO Linux によるオーバヘッドは割合としては大きくない．

総じて，ファイルのオープンやプログラムの実行など文字列比較を伴う操作は遅くなるものの，通常の用途ではオープンされたファイルを読み書きする頻度の方が高いため，現実には体感できるほどの大きなオーバヘッドは生じない．

4. TOMOYO Linux LiveCD

本発表では LiveCD による TOMOYO Linux のデモンストレーションも実施する．

TOMOYO Linux プロジェクトでは，TOMOYO Linux を簡単に体験，使用できる LiveCD の ISO イメージを公開している．ISO イメージをダウンロードし，CD-R に焼いて起動することで，利用者は既存の環境に影響を与えることなく，TOMOYO Linux を使用することができる．

LiveCD のベースとなるディストリビューションは，近年デスクトップ向け Linux として人気のある Ubuntu の日本語ローカライズド Desktop CD である．通常の Ubuntu と異なるのはカーネルだけであるので，利用者にとっては Ubuntu を操作するのと同じように TOMOYO Linux を利用することができる．

LiveCD はシステム起動時から，すべてのプロセスを学習モードで起動するように設定されており，カーネルはプロセスがどのファイルにアクセスしたかをすべてポリシーとして記録する．記録されたポリシーは TOMOYO Linux のポリシーエディタで容易に閲覧することができる．ポリシーを閲覧することで，どのようなプロセスが起動されたのか，どのようなファイルにアクセスしたかを把握することができ，システムの挙動を容易に理解できる．

参 考 文 献

1) 原田季栄, 保理江高志, 田中一男, ”使いこなせて安全な Linux を目指して”, Linux Conference 2005, no.CP-09, November, 2005.
2) `http://tomoyo.sourceforge.jp/`
3) Peter Loscocco, Stephen Smalley, ”Integrating Flexible Support for Security Policies into the Linux Operating System”, Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference, pp.29-42, Boston, USA, June 2001.
4) `http://www.nsa.gov/selinux/`
5) `http://en.opensuse.org/AppArmor`
6) 松田直人, 田端利宏, 宗藤誠治, ”LSM のオーバヘッド測定によるセキュア OS の性能比較”, コンピュータセキュリティシンポジウム 2007, no.9A-4, November, 2007.