CE Linux Forum Japan Technical Jamboree12

組み込みシステム用のセキュアOSとしてのTOMOYO Linux

2006.12.08

株式会社NTTデータ
基盤システム事業本部
オープンソース開発センタ
原田季栄（Toshiharu Harada）、半田哲夫（Tetsuo Handa）

組み込み用途に便利な仕様

- TOMOYO Linuxならファイルシステム非依存
  - xattr を使わない
  - inode 番号を使わない
- TOMOYO LinuxならBusyBox対応
  - ハードリンクの場合はそのままで区別可能
  - シンボリックリンクの場合は alias 構文で区別可能
- TOMOYO Linuxなら省メモリ
  - カーネルのコンパイル時にファイルシステムの xattr サポートやLSMサポートを除外可能
  - ポリシー保持用メモリは動的に確保

組み込み用途に便利な仕様

- TOMOYO Linuxはユーザランドプログラムの修正不要
  - 一切の修正は不要
- カーネルコマンドラインの固定化
  - ポリシーローダーとして init=/.init を追加するだけ
  - 起動時にカーネルコマンドラインを操作できない環境でも使えます
- GUI環境不要
  - シリアル接続やSSH接続を利用してポリシーを操作可能

組み込み用途に便利な仕様

- 必要最小限のファイルの抽出
  - 使われないファイルを削除することでディスク領域を節約可能
- デバイスファイルのなりすまし防止
  - デバイスファイルの名前と属性の対応を強制可能
  - 例えば /dev/null が c 1 3 であることを保証
- パケットフィルタリング機能
  - ドメイン単位でIPアドレスやポートを制限可能
- audit 機能の無効化に対応
  - audit ログを保存しない場合は機能を無効化できる

初心者に優しい仕様

- 使わない機能の除外
  - 使わない機能はカーネルのコンパイル時に除外できます
- 使う機能の選択
  - ファイル制御、ネットワーク制御、ケーパビリティ制御等の機能の中から使いたい機能だけを有効にできます
- 保護範囲の選択
  - ドメイン単位で使う機能を選択できます
  - ドメインはパス名により表現されているので容易に理解できます

理解しやすさを意識した仕様

- 実行順序に基づくドメイン遷移
  - 原則としてプログラムを実行する度にドメイン遷移が発生
  - コンテキストに応じて必要最小限の許可を付与
- ドメイン遷移を伴わないプログラムの実行
  - 不要なドメイン遷移を抑制
  - ドメイン数の削減による消費メモリの削減
  - 関連するプログラム群を同一ドメインで実行可能
  - ログイン後の操作に関して実行順序を無視することが可能（Postfix等でも役に立ちます）

組み込み系要望への対応概要

- TOMOYO Linuxは純国産のセキュアOSです。
- ご質問、ご要望は日本語でどうぞ。(^O^)v
- ご要望は積極的に対応します。
- ブログやwebに書いただけで、機能に取り入れられることがあります(笑)。
- 最新バージョンである1.3.1は、既に組み込み系の方々のご意見を反映したものとなっています。
  - 次ページ以降でTOMOYO Linuxユーザml(http://lists.sourceforge.jp/mailman/listinfo/tomoyo-users)でのやりとりの一部をご紹介します。

シンボリックリンクでもドメインを分離

- M氏コメント
  - これは、欲しかった機能です。BusyBox以外にも、Qtなども同じような問題がある気がしますので・・・。今Zaurus(OpenZaurus)にTOMOYOとLIDSを入れてkexecで切り替えて遊んでいますが、おおむねよい感じです。(^^; 早速1.1.3を試してみます。
- K氏コメント
  - これは、busybox環境にはいいですね。

監査ログの有効/無効切り替え

- K氏コメント
  - セキュリティ面では問題があると思いますが、組み込み向けには、コンパイル時はなく、なんらかの動的スイッチを用意してあるほうがありがたいです。
    - コンパイルせずに監査ログを無効にできるようにしました。

設定ファイル、ツール保存場所を可変に

- M氏コメント
  - あと、細かい話なのですが、Toolの類のおき場所は/rootの下にハードコード されていましたが、コンパイル時に設定出来ると嬉しいです。
  - 組み込みでは構成として/rootがなかったり、そもそもユーザーとしてのrootがいなかったりします。例えば#define一箇所で指定されていると移植が楽かな？と思います。
    - Tool の場所についてはハードコーディングされていないのでどこに置いても構いませんが、root しか使いませんし、 root 以外が使っても意味がありません。/root/ccstools/ 以外に置く場合は /root/security/manager.txt も変更してください。
    - Policy の場所は /root/security/ でハードコーディングされています。
      - 1.3 ではカーネル内のハードコーディングは廃止されました。

ポリシーの保存場所

- S氏コメント
  - 私がTOMOYO Linuxを使って最初に違和感を感じたのは、/root以下に設定ファイル＋ツールが置かれていることでした。思うに、設定ファイルは/etc以下、例えばセキュリティ関連ですので、/etc/security以下にccs (? またはtomoyo?) ディレクトリを作ってその下においた方がすっきりする(分かりやすい)と思うのですが・・・。
    - /etc/ccsですっきりしました。
- M氏コメント
  - ポリシーファイルの保存場所は/etc/ccsが良いと思います。
    ＃ほかのMACもそういった感じで配置していますし、良いですね。
    - 1.3 以降は /etc/ccsになっています。

OpenEmbedded対応

- M氏コメント
  - 組み込み対応ということはOpenEmbeddedのサポートしたりとか？　(^^;
    BitBake用のスクリプト(RPMのSpecみたいな物)を追加するだけですので、これ自体は簡単なのですが、OEの環境構築がちょっと大変です。

    以前、原田さんがおっしゃっていましたが、**組み込みなんかでRootFSを削ぐ際のプロファイラとしてもTOMOYOは便利に使えますね。**
    - OEについては要望が多ければ対応を検討します。

Multi-call binary対応（1）

- PANDA
  - TOMOYO 用ツールも busybox 化した方がディスク領域を節約できて嬉しいかなと思って作ってみました。
- S氏
  - 「busybox化」というよりは「Multi-call binary」が正しいですね。早速試させていただきます。そのうち、busyboxへのパッチにすればさらにフットプリントが縮まりそうですね。
  - Multi-call Binary化した方で、上記の7つを使えるようにしたものだと、

```
lrwxrwxrwx 1 root root     8 11月  6 04:48 ccs-auditd -> ccstools
-rwx------ 1 root root 36940 11月  6 04:58 ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 editpolicy -> ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 findtemp -> ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 loadpolicy -> ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 savepolicy -> ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 setlevel -> ccstools
lrwxrwxrwx 1 root root     8 11月  6 04:48 sortpolicy -> ccstools
```

    と38kとなり、この場合では半分位になりました。動作も確認できました。
- M氏
  - おお、効果ありますね。これってDynamic Linkですよね？

Multi-call binary対応（2）

- S氏
  - あ、そうです。
    uClibcのライブラリへDynamic Linkしてます。
- PANDA
  - そのうち、busyboxへのパッチにすればさらにフットプリントが縮まりそうですね。
    ディスク・フットプリントは節約できますが、 メモリ・フットプリントがどれくらい増えてしまうかが心配です。
- M氏
  - 組み込みはswapがないですからねぇ。
- S氏
  - ただし、/etc/ccs/manager.txtには、リンク先の名前ではなくccstools本体のパスを書かないと使えませんでした。その場合、ツール毎の使用制限をかけられませんね。
- PANDA
  - はい、シンボリックリンクだとそうなりますね。
    ハードリンクならハードリンクされた名前を使えるのですが。

メモリ消費量（1）

- M氏コメント
  - /proc/ccs/info/meminfoですが、Shared, Private, Dynamicの3つの項目があります。それぞれ何を意味するのか教えていただけますでしょうか？
    ちなみにZaurusでみたところこんな感じでした
    Shared 131072
    Private 167936
    Dynamic 585455
    Total 884463
    ＃ちなみにdomain_policy.txtは283777バイトでした。
    ＃Policyはちょっとまだいい加減です。

    LIDSとのメモリ消費の比較をしようと思ったのですが、LIDSにはこういう便利な機能がありません・・・（欲しいです）/proc/meminfoで見る限りLIDS（2.2系）の方がメモリを消費します。

メモリ消費量（2）

- PANDA
  - Shared はパス名やドメイン名を記憶しておくために使用されているメモリの量です。

    Private はパス名やドメイン名以外のポリシーを保持しておくために使用されているメモリの量です。

    Dynamic はアクセス許可のチェックやアクセス許可ログ・拒否ログを保持しておくために「一時的」に使用されているメモリの量です。
- M氏
  - Shared+Privateが正味のAC保持のためのメモリ使用量になるわけですね。
- 参考
  - http://tree.celinuxforum.org/CelfPubWiki/MandatoryAccessControlComparison

alias機能

- M氏コメント
  - 組み込み向けの移植自体はたいした手間ではないのですが、きちんと動くPolicyを書くのは(LIDSでもTOMOYOでも)やっぱり大変ですね。

    あと、OpenZaurusではRAMDISK上に /var をおくので・・・そのあたりのinode番号が起動時に変わります。orz
    こうした場合、Pathnameが便利ですね。あと、TOMOYOの新しいalias機能はやっぱり便利です。感謝！