TOMOYO Linux for Secure Embedded

Toshiharu Harada

haradats@nttdata.co.jp

NTT DATA CORPORATION

February 24, 2008

JFYI

This slide has been under control of *Subversion*.

The version you are viewing is

```
$Id: tomoyo.tex 71 2008-03-07 10:06:22Z haradats $
```

The latest and a better version is

`http://sourceforge.jp/projects/tomoyo/document/fosdem2008.pdf`.

Visit the elinux Wiki for general information of TOMOYO Linux

`http://elinux.org/TomoyoLinux`.

About

This slide

. . . was prepared for the presentation of TOMOYO Linux at the FOSDEM'08 Embedded Track.

Toshiharu Harada <haradats@nttdata.co.jp>

. . . is project manager of TOMOYO Linux and the author/speaker/contact person of the program.

NTT DATA CORPORATION

. . . is the largest SI company in Japan and has been supporting the project.

`http://www.nttdata.co.jp/en/index.html`.

Executive Summary

- TOMOYO Linux is designed and developed to be a practical security enhancements to Linux 2.4 and 2.6.
- It can be used to analyze a system as well as protection.
- Ubuntu based LiveCD (ISO image) is available.
- There are two different versions:

  Version 1.x non LSM, full featured.

  Version 2.x LSM compliant, for mainline inclusion.
- Gorgeous GUI should be available soon.

TOMOYO Linux Security Goal

Quotes from the message posted to LKML *(Linux Kernel Mailing List)*.

> *The TOMOYO Linux's security goal is to provide "MAC that covers practical requirements for most users and keeps usable for most administrators".*
>
> *TOMOYO Linux is not a tool for security professional but for average users and administrators.*

Full message can be found at
`http://lwn.net/Articles/263179/`.

What is TOMOYO Linux

- TOMOYO Linux is a *mandatory access control* implementation to Linux **2.4 and 2.6 kernel**.
- TOMOYO Linux consists of set of kernel patch files and utilities. No userland modifications are needed for TOMOYO Linux.
- Available packages include:

  RedHat Linux 9, Fedora Core 3-6, Fedora 7/8, CentOS 4.6/5.1, Debian Sarge/Etch/Lenny, OpenSUSE 10.1/10.2/10.3, Asianux 2.0/3.0, Ubuntu 6.06/6.10/7.04/7.10, Vine Linux 4.2, Gentoo 2007.0 and Turbolinux 10/11 Server.

Project Official Information

- `http://elinux.org/TomoyoLinux` (English)
- `http://tomoyo.sourceforge.jp/` (English and Japanese)
- `http://tomoyo.sourceforge.jp/wiki-e/` (English)
- `http://sourceforge.jp/projects/tomoyo/` (English and Japanese)
- `http://tomoyo.sourceforge.jp/cgi-bin/lxr/source` (the code)

If you have not heard of `TOMOYO Linux`, elinux.org Wiki word is the best place to start.

“mandatory access control”

- often called as ‘MAC’.
- the opposite of ‘DAC’ (Discretionary Access Control), that was included in good and sweet old UNIX and early days Linux.
- MAC was designed to compensate the shortages of DAC.

MAC has been ported to most modern operating systems by now. SELinux is an implementation of the MAC to Linux.

Please refer TCSEC and NSA SELinux web site for more information.

Why developing another MAC while SELinux is available?

- The project members were not smart enough to use SELinux. :-)
- Linux is open source. So, why not? :-)

How many MACs are there on this planet?

- Already *in-tree*
  - SELinux
- Just (2.6.25) *in-tree*
  - SMACK (Simplified Mandatory Access Control Kernel)
- Want to be *in-tree*
  - AppArmor (formerly known as SubDomain)
  - TOMOYO Linux
  - LIDS (Linux Intrusion Detection System)

- "Our Own Way" for some reasons
  - RSBAC
  - GRESECURITY
- Where to find the players?
  - *"Linux Weather Forecast"* run by the Linux Foundation might help.
    `http://www.linux-foundation.org/en/Linux_Weather_Forecast`

How do they compare?

I am trying to provide a fair and helpful chart. Comments and suggestions would be appreciated.

`http://tomoyo.sourceforge.jp/wiki-e/?WhatIs#comparison`

TOMOYO Linux General Features

1. *"Policy learning mode"* revolutionary enlightens the loads of policy management tasks.
2. Policy definitions of TOMOYO Linux is exceptionally *human readable and understandable.*
3. TOMOYO Linux can live with *any file system* (built on top of the VFS layer).
4. Small footprint

Key Concepts

- TOMOYO Linux limits the access *by the behavior of the subject* while SELinux limits the access *by the attributes of the subject and object*.
- TOMOYO Linux kernel keeps track of the **process invocation history** (a.k.a. system call chains). Every process remembers its parent and ancestors.
  - This scheme quite well suits to the Linux's fork/exec mechanism. Process invocation history information is copied when `fork()` and added the new process name when `execve()` issued.

Terminology

- TOMOYO Linux Policy is composed of per DOMAIN access control definitions.
- In TOMOYO Linux, DOMAINs are *automatically defined and given names* by the TOMOYO Linux kernel, no operations required for the administrators side.
- Access control modes for each DOMAIN is specified by a number that ranges 0...255. That number is called as PROFILE id.

- Administrators need to define the meaning of each PROFILE id in `/etc/ccs/profile.conf`.
- Administrators define the entity of each PROFILE id by choosing the desired MAC functionality.

- While SELinux has system wide global modes called *"enforcing"* and *"permissive"*, TOMOYO Linux has per domain four MODEs to administrate MAC behavior.

Mode	unpermitted access	Log	domains
0 (Disabled)	granted		grows
1 (Learning)	granted and *learned*	logged	grows
2 (Permissive)	granted	logged	grows
3 (Enforcing)	rejected	logged	unchanged

- Learning mode is for designing a policy.
- Permissive mode is for testing the policy.

Profile definitions

Syntax

```
[Profile id(0-255)]-[Directive]=[Mode]
```

Example

```
1-COMMENT=----- Profile #1 Defs -----
1-MAC_FOR_FILE=1
1-MAC_FOR_NETWORK=1
2-COMMENT=----- Profile #2 Defs -----
2-MAC_FOR_FILE=2
2-MAC_FOR_NETWORK=2
```

Profile Sample

Description

The following example defines profile number 35, *"file access is enforced (access will be restricted) and network and signal are learned (access will not be restricted)"*.

Example

```
35-MAC_FOR_FILE=3
35-MAC_FOR_NETWORK=1
35-MAC_FOR_SIGNAL=1
```

TOMOYO Linux supports MAC only for files?

"Hell, No!" :-)

TOMOYO Linux supports MAC for *networking* (MAC_FOR_NETWORK), *signals* (MAC_FOR_SIGNAL), *capabilities* (MAC_FOR_CAPABILITY) ... and much much more.

For the meanings of each directives, please RTFM.

"Domain" of TOMOYO Linux

Domain name starts from the prefix string, "<kernel>".
Names of the programs (issued `execve()`) will be appended sequentially with a single space character as a separator. (quite simple, huh?)

<kernel> /sbin/init

<kernel> /sbin/init /boo /bar

<kernel> /sbin/init /foo /bar /buz

TOMOYO Linux ACL Example

Captured on Ubuntu 7.10 Desktop.

```
<kernel> /sbin/init
--x /bin/dash
rw- /dev/console
rw- /dev/null
r-- /etc/event.d/control-alt-delete
r-- /etc/event.d/logd
...
```

Policy Configuration Files

All policy configuration files are located at `/etc/ccs` in plain text format.

Name	File
Domain Policy	`domain_policy.conf`
System Policy	`system_policy.conf`
Exception Policy	`exception_policy.conf`

Policy Editor (CUI)

CUI program "`editpolicy`" is located at `/usr/lib/ccs` with other TOMOYO Linux tools.

`editpolicy` has roles of:

1. To show the current (on memory) policy settings
2. To modify them.

`editpolicy` understands and deals with the all three TOMOYO Linux policy configuration files.

Policy Editor (CUI)

When invoked, domain policy screen will be displayed.
Entering TAB key will change the screens.

Screen Name	Description
Domain Policy	Shows domain transition trees. The total number of existing domains will be displayed in the top of the screen. Move cursor control keys to choose a domain.
System Policy	Shows system global settings.
Exception Policy	Shows lists of MAC access control exceptions.

Other tools

savepolicy

Saves current policy settings under `/etc/ccs`.

loadpolicy

Loads policy settings to the kernel.

TOMOYO Linux LiveCD

Definition

TOMOYO Linux LiveCD = Ubuntu 7.10 Desktop +
TOMOYO Linux kernel and tools

For what?

Casual, safe and free trial of TOMOYO Linux.

Where can I get one?

`http://tomoyo.sourceforge.jp/wiki-e/?TomoyoLive`

LiveCD Design Specification

TOMOYO Linux LiveCD comes with predefined policy definition.

`domain_policy.conf`

```
# cat /etc/ccs/domain_policy.conf
<kernel>
use_profile 1
#
```

This means, "run the `<kernel>` domain with profile 1". The definition of profile with id 1 can be found at `profile.conf`. Since the mode for file access (MAC_FOR_FILE) of profile 1 is not enforcing, domains are created as needed.

profile.conf

```
# head /etc/ccs/profile.conf
1-MAC_FOR_FILE=1
1-MAC_FOR_NETWORK=1
1-MAC_FOR_SIGNAL=1
...
```

Profile 1 is defined to remember access for *files*, *networks*, *signals*... and keeps the results on memory. Thus, TOMOYO Linux LiveCD allows every access (as usual Ubuntu LiveCD) and keeps remembering them.

The results depend on your environment. For instance, number of domains and each ACL for specific domains.

How to use a LiveCD

1. save and burn the ISO image.
2. set the disc to your PC.
3. BOOT IT!

Or, you can define a virtual machine that reads the saved ISO image.

What can I do with TOMOYO Linux LiveCD?

In addition to anything you can do with Ubuntu,

- View how Ubuntu are running on your PC by browsing domain transitions and ACL information obtained from policy learning mode.
- Select domains and change the behavior. For instance, selecting a shell domain and change it from policy learning mode to enforcing mode.
- Exploring what will happen if you click GNOME menus. (Everything including GUI and X are monitored by TOMOYO Linux kernel)

- TOMOYO Linux LiveCD can be used to install to your PC. Here's the instruction.
  1. Boot from TOMOYO Linux LiveCD.
  2. Find icon named "Install".
  3. *Click* it, as usual.
  4. Follow the instruction from the LiveCD.

  This is by far the easiest way to install TOMOYO Linux.

Policy Editor (GUI)

Implemented as an Eclipse plugin. Runs on both Linux and Windows. Communicates using SSH. No server side modifications needed.

Coming soon.

TOMOYO Linux for Secure Embedded

Reasons to use TOMOYO Linux with Embedded

- For use with 2.4 kernel (LSM does not support 2.4)
- File system independent
- Userland modifications free
- BusyBox support
- Small footprint

Digging issues from LKML

How about labels?

Pros

- Label is bound to inode that we can trust, so unaffected by renames.

Cons

- Needs extra resource and work for labels.
- Inode can be changed by operations.
- Label is defined according to pathnames. :)

Label has clear advantages but does not solve every cases.

Label or Pathname?

That's ridiculous.

What pathname means is a "name of the place". Pathname is not bound to the attributes of the entity.

- No one can live without pathnames.
- Both Label and Pathname are incomplete and needs work.
- People can easily argue.

LABEL vs. PATHNAME argument

Most important difference is LABEL or PATHNAME.

label Assign 'labels' to the objects and judge access control according the label information. xattr (extended attributes) are used to store label information. SELinux and SMACK belong this.

pathname Don't use labels and live with traditional pathnames. AppArmor and TOMOYO Linux belong this.

Issues with pathname

Pros

- Pathnames are handy for restricting behavior of the subject.

Cons

- Pathnames can be easily changed by operations (`mount`, `chroot`, `/etc/../etc/shadow`, `ln`...).

Summary

- TOMOYO Linux is a lightweight and usable MAC implementation to Linux.
- Though it was designed for servers, it also suits embedded systems very well.
- TOMOYO Linux is useful for analyzing *your own Linux box*, so it would help embedded developers.
- We are having unclear difficulties in mainlining TOMOYO Linux. We promise to continue the work.
- We will be happy if you use our work, if TOMOYO Linux helps you. We will be pleased to help you.

Project History

March 2003 I met *Tetsuo Handa* and started working at a small building called *Kayaba-cho Tower*. Tetsuo has wrote the first version of TOMOYO Linux and is the chief architect of the project ever since.

November 2005 Made the code open (GPL2)

April 2007 First proposal posting to LKML.

April 2007 Embedded Linux Conference 2007, *""TOMOYO Linux: A Lightweight and Manageable Security System for PC and Embedded Linux"*. Thank *Tim Bird* for this cool title.

June 2007 Ottawa Linux Symposium 2007, *"TOMOYO Linux BoF"*.

November 2007 PacSec 2007, *"TOMOYO Linux: A Practical Method to Understand and Protect Your Own Linux Box"*.

December 2007 6th proposal posting to LKML.

February 2008 FOSDEM'08 presentation, *"TOMOYO Linux for Secure Embedded"* (U R here)

Every slides and papers are stored at `http://sourceforge.jp/projects/tomoyo/docman/` (choose the *category* and the *language*).

Acknowledgments

Special thanks goes to...

- *Linus Torvalds* for creating this wonderful world of Linux.
- *Donald E. Knuth* for bringing the world TEX and letting us to know *the joy of writing*.
- *Bill Joy* for creating **"the editor"**, *"vi"*. It's part of myself.
- Staff of SourceForge.jp (`http://sourceforge.jp/`) for their continuous supports and hosting.
- NTT DATA CORPORATION for supporting the project.

- *Jack Bauer* and the staff of Twenty Four series. They taught me what the brave is.
- My family for their patience and forgiveness. Also, my dog, *Wish* for taking me to a walk everyday and not biting me so often as before.

Thanks!

We will see you at ELC2008 and OLS2008.

Toshiharu Harada <haradats@nttdata.co.jp>

Tetsuo Handa <penguin-kernel@i-love.sakura.ne.jp>

Kentaro Takeda <takedakn@nttdata.co.jp>