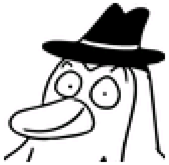
TOMOYO Linux

http://elinux.org/TomoyoLinux
http://tomoyo.sourceforge.jp/wiki-e/

BoF

June 29, 2007

Toshiharu Harada
<haradats@nttdata.co.jp>
<haradats@gmail.com>
NTT DATA CORPORATION

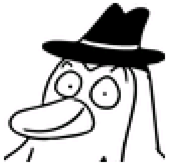


Overview

- Please see and experience our four years work.
- Please understand what TOMOYO Linux is and why we posted RFC to the LKML.
  – It’s not merely just another MAC, but the evolutionary new concept.
- Please give us your feedback.

Who am I?

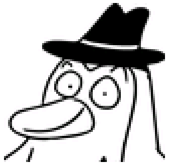
- **First public presentation: EUUG 1990 held at Munich (it was my very first foreign travel…)**
  - ***“PCSERVE: an attempt to integrate PC users into the UNIX community”***
- **Now working at NTT DATA CORPORATION as a project manager of open source development.**
- **Second public presentation at Embedded Linux Conference 2007.**
  - ***“TOMOYO Linux - A Lightweight and Manageable Security System for PC and Embedded Linux”***



What is TOMOYO Linux?

- *Another* pathname-based MAC (Mandatory Access Control)
- Available since Nov. 2005.
- Composed of patches and tools.
- Dedicated gorgeous GUI also available.
- RFC posted to LKML recently.

	LSM	Non-LSM
2.4	N/A	Ver 1.4.1
2.6	Ver 2.0	Ver 1.4.1



Feedback so far

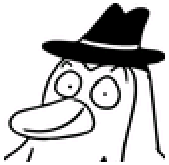
- **Russell Coker's suggestions (when he visited Japan in 2006).**
  - ***"Firstly I suggest that you use the Linux auditing system for event logging."*** **… it's done (2.0)**
  - ***"Next I suggest using LSM interfaces."*** **… it's done (2.0)**
  - ***"If you can entirely use LSM interfaces then Tomoyo can be a candidate for inclusion in kernel.org kernels."*** **… not yet :(**

Feedback so far

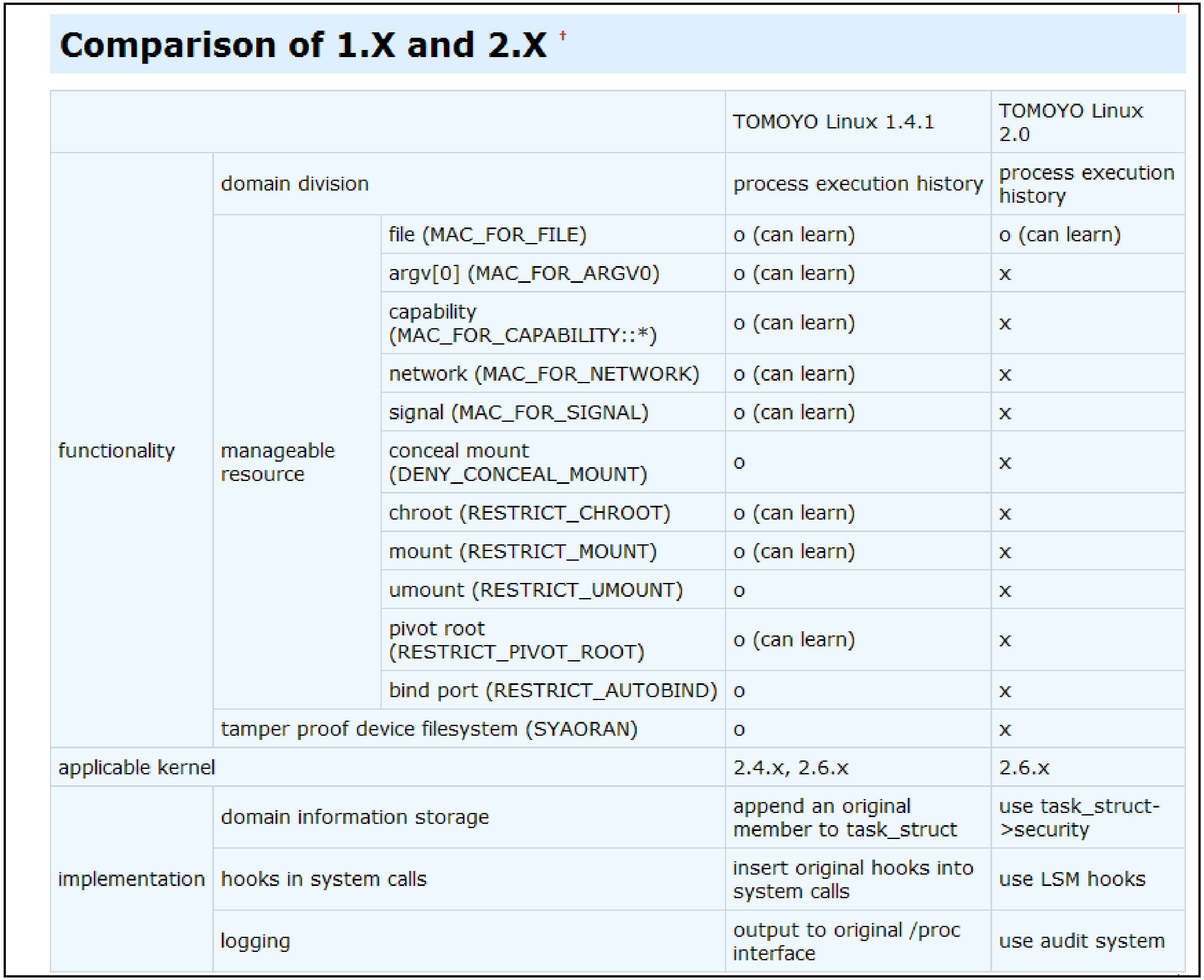
- **TOMOYO's RFC**
  - **http://lkml.org/lkml/2007/6/13/58**
  - **http://lkml.org/lkml/2007/6/14/55**
- ***Why can't you do this via SELinux domain transitions?***
  - **Stephen Smaley**
- ***SELinux audit logs (well, whatever is in /var/log/audit on my system) does show the path names of objects...***
  - **Rik van Riel**
- ***Blindly generating security policy through observation of the system is potentially dangerous for many reasons.***
  - **James Morris and Russell Coker**
- ***Please stop wasting your time on pathname-based non-solutions.***
  - **Christoph Hellwig, and other people.**

My Answer

- *Why can't you do this via SELinux domain transitions?*
  - **TOMOYO Linux can show domain transitions for you.**
- *SELinux audit logs (well, whatever is in /var/log/audit on my system) does show the path names of objects...*
  - **You need to define SELinux policy first to do that. You don’t need preparation to use TOMOYO Linux.**
- *Blindly generating security policy through observation of the system is potentially dangerous for many reasons.*
  - **Who would provide SELinux policy for my server?**
- *Please stop wasting your time on pathname-based non-solutions.*
  - **Please tell me why you think so and let’s talk.**

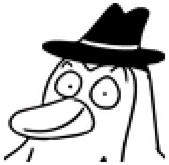


Version 1.4.1 and 2.0



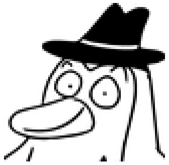
Comparison of 1.X and 2.X †

			TOMOYO Linux 1.4.1	TOMOYO Linux 2.0
functionality	domain division		process execution history	process execution history
	manageable resource	file (MAC_FOR_FILE)	o (can learn)	o (can learn)
		argv[0] (MAC_FOR_ARGV0)	o (can learn)	x
		capability (MAC_FOR_CAPABILITY::*)	o (can learn)	x
		network (MAC_FOR_NETWORK)	o (can learn)	x
		signal (MAC_FOR_SIGNAL)	o (can learn)	x
		conceal mount (DENY_CONCEAL_MOUNT)	o	x
		chroot (RESTRICT_CHROOT)	o (can learn)	x
		mount (RESTRICT_MOUNT)	o (can learn)	x
		umount (RESTRICT_UMOUNT)	o	x
		pivot root (RESTRICT_PIVOT_ROOT)	o (can learn)	x
		bind port (RESTRICT_AUTOBIND)	o	x
	tamper proof device filesystem (SYAORAN)		o	x
applicable kernel			2.4.x, 2.6.x	2.6.x
implementation	domain information storage		append an original member to task_struct	use task_struct->security
	hooks in system calls		insert original hooks into system calls	use LSM hooks
	logging		output to original /proc interface	use audit system



Eye-friendly version

- TOMOYO 1.X
  - MAC for file, network, signal, capabilities, and more…
  - Our own hooks, capabilities and audit.
- TOMOYO 2.X
  - MAC for file (so far, to be extended)
  - Prepared for LKML posting
- TOMOYO common
  - Per domain access control modes
  - Policy learning mode.



From SELinux ml

2007/6/19, Daniel J Walsh dwalsh@redhat.com>:

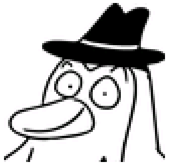
> Steven mentioned in another conversion the idea of a Per Domain
>
> Permissive Mode. This is something our customers are looking for.

...

> Having a simple domain that would run in permissive mode while the rest
>
> of the machine ran enforcing would satisfy this need.

Good point.

It’s already done with TOMOYO Linux.



From my recent posting to the legendry AA thread

Date Sun, 24 Jun 2007 09:10:42 +0900
From Toshiharu Harada <>
Subject Re: [AppArmor 39/45] AppArmor: Profile loading and manipulation,
pathname matching

This thread is amazing. With so many smart people's precious time,

What are the results?
What are the issues anyway?
Is anyone happy? (I'm not and I assume Chris is not)

Yes, "waste of time" is taking place here, but
it's not for "pathname-based MAC" but for "wrongly posted messages",
I believe. I'm a relatively new to this ml, let me ask.

Is this ml a place of judge or battle? (not to help or support?)

Nothing is perfect, so we can work to make things to better, right?
I have suggestions:

Let's clarify issues first.
- problems (or limitations) of pathname-based MAC
- advantages of pathname-based MAC
- how can pathname-based MAC supplement label based
(Stephen, James and Kyle, please help)

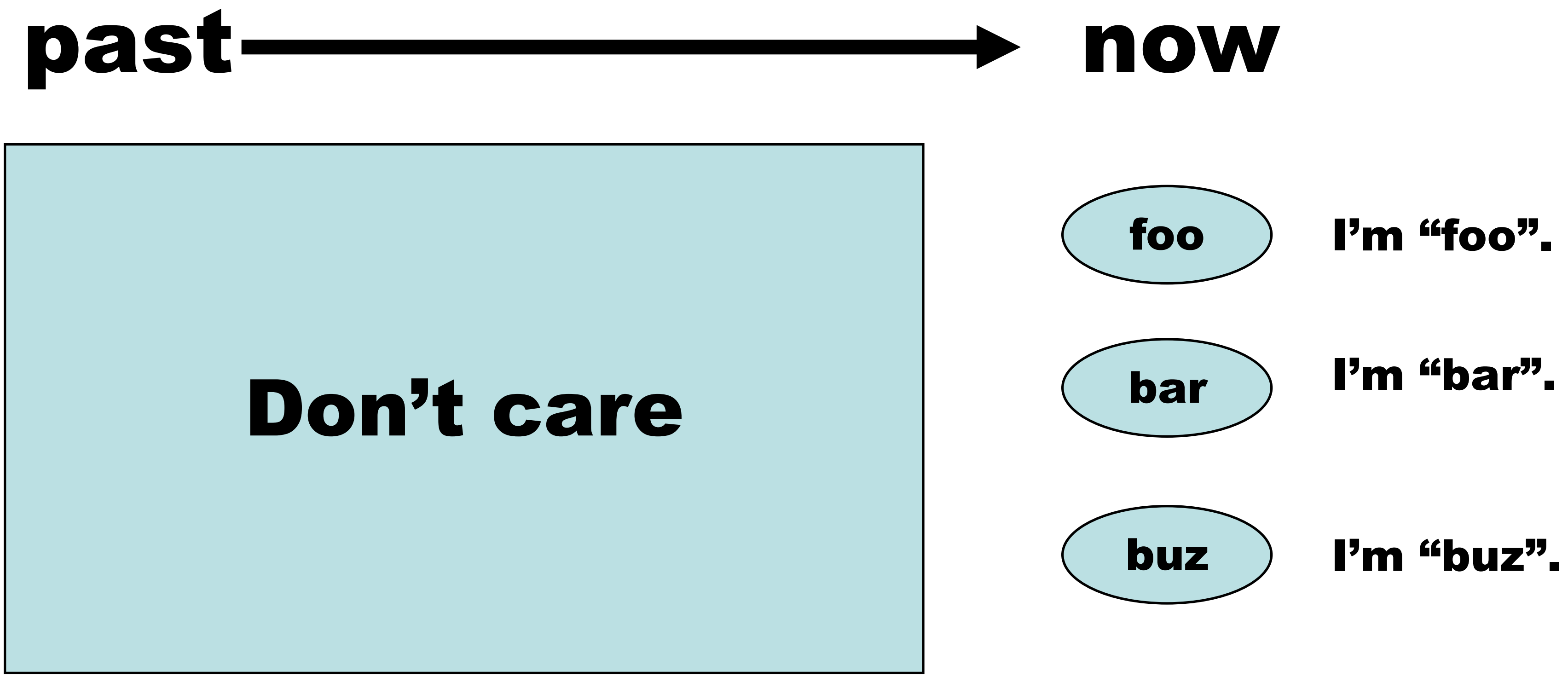
The result

- My message was **totally ignored and I'm very sad... ;(**
- My question is
  - "Am I the only one?"
  - "What do you feel with AA thread?"
- We came here all the long way from Japan (transit at Chicago!) to talk, so please talk with us.

Anyway…

- Yes,
  - TOMOYO Linux is a pathname-based MAC. (I don’t deny it :)
- No,
  - TOMOYO Linux is a sample implementation of our new idea of “having Linux to remember process invocation history”
- OK,
  - I’ll explain the idea.

Conventional way



past → now

Don't care

foo

I'm "foo".

bar

I'm "bar".

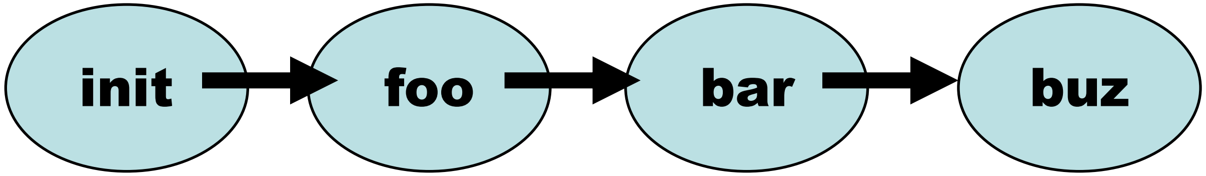
buz

I'm "buz".

Each process doesn't know its ancestors.

AppArmor and SELinux belong to this category.

TOMOYO Linux Way



init → foo → bar → buz

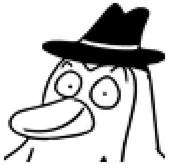
I'm "init -> foo".

I'm "init -> foo -> bar".

I'm "init -> foo -> bar -> buz".

Every process knows its ancestors (or process invocation history aka call chain) at any point of time.

This idea is what we implemented and what we want to share with you.

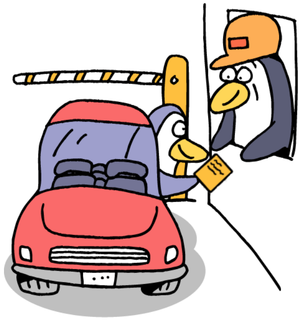


Need figures?

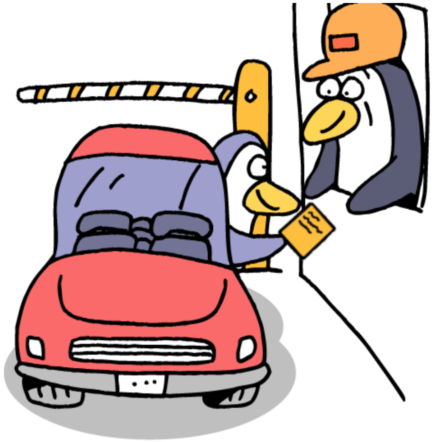
<kernel> /sbin/mingetty /bin/login /bin/bash /bin/ls

<kernel> /sbin/mingetty /bin/login /bin/bash

<kernel> /sbin/mingetty /bin/login



<kernel> /sbin/mingetty



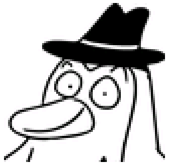
How we implemented

- TOMOYO Linux version 1.X
  - We invented “our own hooks”
  - We invented “our own capabilities”
  - We wrote “our own audit functions”
- TOMOYO Linux version 2.0
  - It’s using LSM now.
  - Kernel patch is just 43 lines. (LSM is great!)

What else can it be applied other than MAC?

- Profiling
- Auditing
- And …

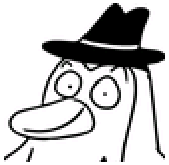
- We would like to know the ideas and possibilities.
- Please let us know and we’ll be happy to help.



Essences of TOMOYO Linux

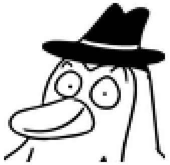
Linux with
process invocation history
mechanism

We applied this idea to classify each difference call chain as an independent “domain”. That’s the spirit of TOMOYO Linux.



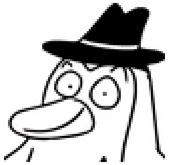
I guess

- It’s time for demonstration.
- See what TOMOYO Linux does.



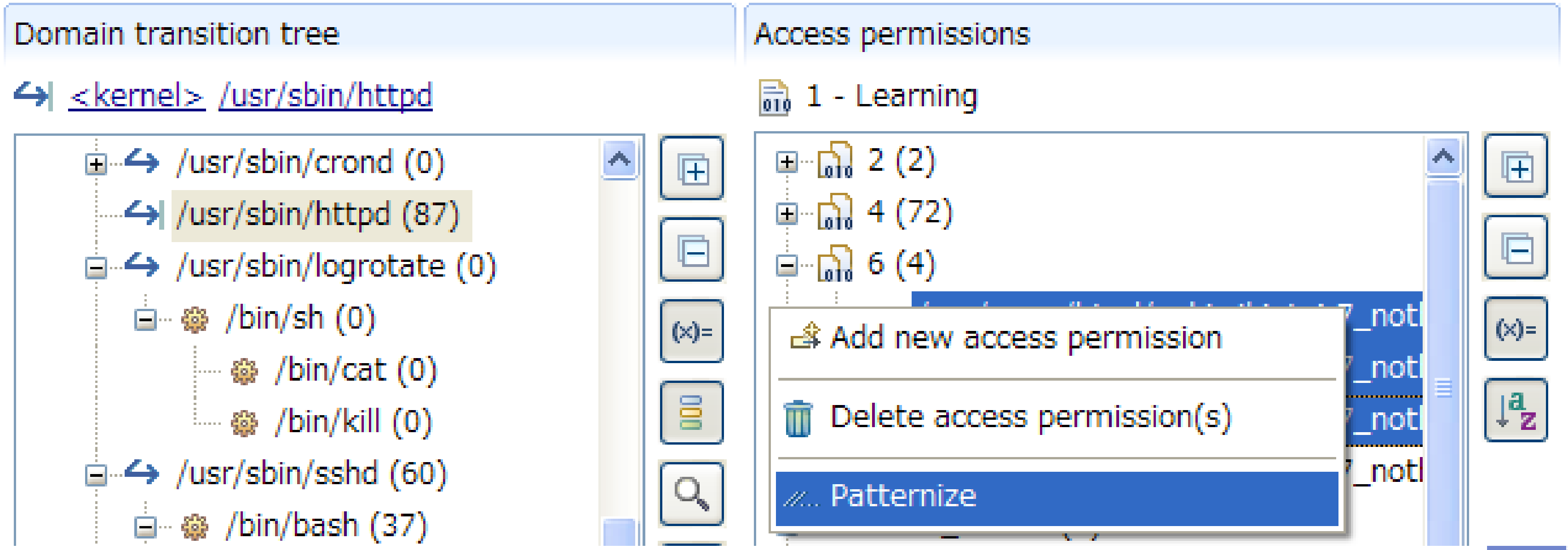
Now it's your turn

- To give us feedback. :)



TOMOYO ~~GUY~~GUI

- **Superb dedicated GUI for TOMOYO Linux is ready to go.**
- **It’s open source as TOMOYO Linux is.**



<kernel> /usr/sbin/httpd

/usr/sbin/crond (0)
/usr/sbin/httpd (87)
/usr/sbin/logrotate (0)
/bin/sh (0)
/bin/cat (0)
/bin/kill (0)
/usr/sbin/sshd (60)
/bin/bash (37)

1 - Learning

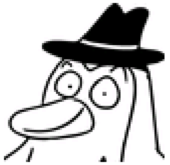
2 (2)
4 (72)
6 (4)

SELinux and TOMOYO Linux

- There’s no doubt SELinux and LSM are great work.
- It’s a matter of choice and freedom, isn’t it?
- We see no reason to disallow non-SELinux MAC to Linux.
- TOMOYO Linux’s idea may even enhance SELinux.

AA and TOMOYO Linux

- As MAC implementations,
  - They look quite similar.
- “Policy generation” function looks similar, but that’s not true.
  - TOMOYO Linux does it on the fly while what AA does is converting from log (like audit2allow).
  - TOMOYO Linux takes care from booting to shutdown.
- AA’s “ChangeHat” is a nice idea.

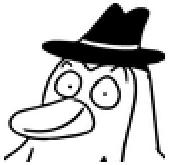


Thank you

- For coming.
- For your valuable feedback.

- It’s a great pleasure for us to interact with you.
- We love you all and Linux. (we are open minded)

- See you again. (May TOMOYO Linux be with you...)



I’ll be back.

(if possible)