KERNEL DEVELOPMENT: DRAWING LESSONS FROM "MISTAKES"

Japan Linux Symposium 2009
October 23, 2009
Toshiharu Harada
haradats@nttdata.co.jp
NTT DATA CORPORATION
Every kernel developer knows that Linux comes with plenty of precious documentation as an integral part. From coding style to how to post patches, almost everything has been documented. However, history shows that error is human nature. Sometimes developers do not well know Don’ts, but there are also cases when they make mistakes despite being aware of such rules. Why this happen is unsolved, but a documentation, so far missing, of the consequences of this misbehavior could discourage it. The presenter is project manager of TOMOYO Linux, a security enhancement feature merged in version 2.6.30. Thinking open-minded, he decided to share the errors his project made, wishing it could be a helpful warning to other projects, especially newcomers. In this presentation, it will try to explain the mistake circumstances in TOMOYO Linux project, highlighting the thoughts of project members and the community reactions.
“Experience is the name everyone gives to his mistakes” --- Oscar Wild
WHAT’S THIS ALL ABOUT
WHAT’S THIS ALL ABOUT

• Linux comes with a set of great documentation
WHAT’S THIS ALL ABOUT

• Linux comes with a set of great documentation

• DOs and DON’Ts are there already
WHAT’S THIS ALL ABOUT

• Linux comes with a set of great documentation
• DOs and DON’Ts are there already
• Yet we keep on making mistakes
WHAT’S THIS ALL ABOUT

• Linux comes with a set of great documentation
• DOs and DON’Ts are there already
• Yet we keep on making mistakes
• Finding a missing piece
WHAT IS MISSING?
WHAT IS MISSING?

• In my humble opinion:
WHAT IS MISSING?

• In my humble opinion:
  • Human nature (hard to fix)
WHAT IS MISSING?

• In my humble opinion:
  • Human nature (hard to fix)
  • Most of us don’t really like readings (hard to fix)
WHAT IS MISSING?

• In my humble opinion:
  • Human nature (hard to fix)
  • Most of us don’t really like readings (hard to fix)
  • Real-life examples taken from TOMOYO Linux project
WHO AM I?
WHO AM I?

• Project manager of TOMOYO Linux
WHO AM I?

• Project manager of TOMOYO Linux

• What is “project manager”?
WHO AM I?

• Project manager of TOMOYO Linux

• What is "project manager"?

  • Something put in between an Open Source projects and a Company
WHO AM I?

• Project manager of TOMOYO Linux

• What is “project manager”?

  • Something put in between an Open Source projects and a Company

• It’s an adventurous role (experimental stage)
OUCH, IS THIS “SECURE LINUX” TALK?

• No. (so please remain seated, you are safe)
COVERED TOPICS

• Chapter 1: Where to find DOs and DON’Ts
• Chapter 2: TOMOYO Linux posting history overview
• Chapter 3: Step by step introduction of DOs and DON’Ts of the TOMOYO Linux
CHAPTER 1
Where to find DOs and DON’Ts
Gentle Reminder
Gentle Reminder

Documentation is a part of Linux Kernel
Gentle Reminder

- Documentation is a part of Linux Kernel
- After checking out the kernel, cd to "Documentation"
Gentle Reminder

Documentation is a part of Linux Kernel

After checking out the kernel, cd to “Documentation”

Problem is “there are just too many files and directories” and people prefer coding than reading
How Great is It?
How Great is It?

$Documentation/ManagementStyle
How Great is It?

$Documentation/ManagementStyle

“Most people are idiots, and being a manager means you'll have to deal with it, and perhaps more importantly, that __they__ have to deal with __you__.”
How Great is It?
How Great is It?

$Documentation/ManagementStyle
How Great is It?

$Documentation/ManagementStyle

"Thing will go wrong, and people want somebody to blame. Tag you're it."
Okay, I’ll do so
Okay, I’ll do so
Okay, I’ll do so
The truth is people will blame you regardless of you are tagged or not (you can omit)

Just these two statements illustrate the essential part of managements

Linux documentation is so practical
Minimal Reading
Minimal Reading

Entire Scheme
Minimal Reading

- Entire Scheme
- Documentation/HOWTO
Minimal Reading

- Entire Scheme
- Documentation/HOWTO
- Submitting Patches
Minimal Reading

- Entire Scheme
  - $Documentation/HOWTO
- Submitting Patches
  - $Documentation/SubmitChecklist
Minimal Reading

- Entire Scheme
  - $Documentation/HOWTO
- Submitting Patches
  - $Documentation/SubmitChecklist
  - $Documentation/CodingStyle
References

HOW TO PARTICIPATE IN THE LINUX COMMUNITY A new guide by Jon Corbet
References

Note that the title is “How to Participate in the Linux Community”
How to Participate in the Linux Community

Note that the title is “How to Participate in the Linux Community”

Making your code upstream means your participation in the Linux Community (Be nice!)
My favorite one


Joining the herd of cats: how to work with the kernel development process

Jonathan Corbet
corbet@lwn.net
Patch lifecycle: repeat

- Gleam
- RFC
- Post code
- Fix things

Mainline

Subsystem tree
Lifecycle: the corporate view
Lifecycle: the corporate view
I was laughing when I saw the slides for the first time (in 2007)

When I came to realize that it was true, I couldn’t laugh any more ...

They kept asking me “not yet?” ;-}
CHAPTER 2
TOMOYO Linux by Numbers
Leo Tolstoy said

“All Happy Families Resemble Each Other, Each Unhappy Family Is Unhappy in Its Own Way.”
9,230
Number of employees of NTT DATA CORPORATION
Number of project members

2,700,000
0.0325%
Possibilities to be assigned to the project
TOMOYO is the 3rd (and the latest) LSM module merged upstream
Number of people who expected TOMOYO would be merged
We posted patches 15 times
716
Merged since 716 days after the first post
Number of comments from LKML
(0.2 comments/day)
## Proposal History


<table>
<thead>
<tr>
<th>No. (article link)</th>
<th>Date (interval)</th>
<th>Subject (thread link)</th>
<th>Parts</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>06/13/07</td>
<td>[RFC] TOMOYO Linux</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>06/14/07 (1d)</td>
<td>[TOMOYO 0/9] TOMOYO Linux security module.</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>08/24/07 (71d)</td>
<td>[TOMOYO 00/15] TOMOYO Linux - MAC based on process invocation history</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>10/02/07 (37d)</td>
<td><a href="repost">TOMOYO 00/15</a> TOMOYO Linux - MAC based on process invocation history.</td>
<td>15</td>
<td>22</td>
</tr>
<tr>
<td>4</td>
<td>10/11/07 (9d)</td>
<td>[TOMOYO #4 00/13] TOMOYO Linux - MAC based on process invocation history.</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>11/17/07 (36d)</td>
<td>[TOMOYO #5 00/18] TOMOYO Linux - MAC based on process invocation history.</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>12/25/07</td>
<td>TOMOYO Linux Security Goal</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>01/08/08 (27d)</td>
<td>[TOMOYO #6 00/21] TOMOYO Linux - MAC based on process invocation history.</td>
<td>21</td>
<td>7</td>
</tr>
<tr>
<td>7</td>
<td>04/04/08 (113d)</td>
<td>[TOMOYO #7 00/30] TOMOYO Linux 1.6.0 released</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>8</td>
<td>05/01/08 (63d)</td>
<td>[TOMOYO #8 (2.6.25-mm1) 0/7] TOMOYO Linux</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>09/24/08 (143d)</td>
<td>[TOMOYO #9 (2.6.27-rc7-mm1) 0/6] TOMOYO Linux</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>10/09/08 (15d)</td>
<td>[TOMOYO #10 (linux-next) 0/8] TOMOYO Linux</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>10/20/08 (11d)</td>
<td>[TOMOYO #11 (linux-next) 00/11] TOMOYO Linux</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>11/04/08 (14d)</td>
<td>[TOMOYO #12 (2.6.28-rc2-mm1) 00/11] TOMOYO Linux</td>
<td>11</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>11/20/08 (16d)</td>
<td>[TOMOYO #13 (mmotm 2008-11-19-02-19) 00/11] TOMOYO Linux</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>01/01/09 (41d)</td>
<td>[TOMOYO #14 (mmotm 2008-12-30-16-05) 00/10] TOMOYO Linux</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>02/05/09 (34d)</td>
<td>[TOMOYO #15 0/8] TOMOYO Linux</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>06/09/09 (124d)</td>
<td>Linux Kernel 2.6.30 release</td>
<td></td>
<td>162 comments</td>
</tr>
<tr>
<td>15 times</td>
<td>716 days</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 3

Drawing Lessons from the “Mistakes” of TOMOYO Linux Project
CHAPTER 3

Drawing Lessons from the “Mistakes” of TOMOYO Linux Project
I wrote:

<snip>

All right, that's almost everything. Please visit the following URL for the code and documents:

http://tomoyo.sourceforge.jp/wiki-e/

If you want to see the code first, then:

DON'T
Send URL

Send patches
DON’T
Send URL

Send patches
WHAT HAPPENED?

• I got a personal message from **Stephen Smalley**, a maintainer of that famous SELinux

  “If you really want feedback or to get your code into the kernel, you need to do more than post a URL to the code - you need to break your code down into a number of patches and post them, just like the AppArmor folks have been doing.”
SO WE RUSHED TO POSTED PATCHES NEXT DAY

- Pavel Machek gave a comment

- “Looks whitespace-damaged to me.”
DON’T
Ignore the Linux standard coding style

Always apply checkpatch.pl
DON’T
Ignore the Linux standard coding style

Always apply checkpatch.pl
DO
• Carefully read the $Document/CodingStyle
DO

• Carefully read the $Document/CodingStyle

• Check your code with $scripts/checkpatch.pl
DO

- Carefully read the $Document/CodingStyle
- Check your code with $scripts/checkpatch.pl
- Also use other $scripts/check*.pl
• **Jiri Kosina** pointed us to make patches bisectable

• “Just a trivial minor nitpick - IMHO this breaks bisectability. It might be better to add the Kconfig/Makefile patch at the end of the whole series, so that bisect doesn't end up in the tree in which Makefile references non-existing files/directories.”
• Add the Kconfig/Makefile patch at the end of the whole series, so that bisect doesn't end up in the tree in which Makefile references non-existing files/directories."
• Add the Kconfig/Makefile patch at the end of the whole series, so that bisect doesn't end up in the tree in which Makefile references non-existing files/directories.”
IN THE 3RD PATCH

• **James Morris** taught us series of patches should form a thread

• “I'd also suggest making all of the patches a reply to the first email, so they can be threaded.”
<table>
<thead>
<tr>
<th><a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</th>
<th>Kentaro Takeda</th>
</tr>
</thead>
<tbody>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>James Morris</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>YOSHIFUJI Hideaki / 吉藤英明</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>James Morris</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Andi Kleen</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>YOSHIFUJI Hideaki / 吉藤英明</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>James Morris</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>YOSHIFUJI Hideaki / 吉藤英明</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Peter Zijlstra</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>KaiGai Kohei</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Peter Zijlstra</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>YOSHIFUJI Hideaki / 吉藤英明</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Peter Zijlstra</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Peter Zijlstra</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Jiri Kosina</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>James Morris</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>David P. Quigley</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
<tr>
<td>Re: <a href="repost">TOMOYO 05/15</a> Domain transition handler functions.</td>
<td>Tetsuo Handa</td>
</tr>
</tbody>
</table>
DO

• Send series of patches as children of the first message so that they can form a thread or you want people to read your messages
DO

• Send series of patches as children of the first message so that they can form a thread or you want people to read your messages
• **James Morris** said:
  
  • “Please use standard kernel list handling, per include/linux/list.h”

• **YOSHIFUJI Hideaki** also mentioned:
  
  • You're introducing a custom API, which is open-coded repeatedly throughout your module.
  
  • All linked lists (at least, new ones) must use the standard kernel list API.
DON’T
Propose new data structure

Use existing one
DON’T
Propose new data structure

Use existing one
IN THE 5TH POSTING

• **James Morris** suggested to CC netdev mailing list

• “You should send anything which touches core networking to netdev, too, and get an ack from one of the core developers there.”
• Carefully choose CCs and get a review from them
• Carefully choose CCs and get a review from them
IN THE 6TH POSTING

• Tetsuo posted 30 series of messages with the subject, “Subject: [TOMOYO #7 00/30] TOMOYO Linux 1.6.0 released”

• The problem was “TOMOYO 1.6.0” did not use LSM and implemented different hooks
DON’T
Try to invent a new API

Respect a standard and follow one
DON’T
Try to invent a new API
Respect a standard and follow one
• Tetsuo and I knew that posting such patches will never be accepted

• However, we had been stuck and we couldn’t find another way

• Our posting was thoughtless, but we were so serious to make our code upstream
I changed my mind and wrote:

We apologize for the confusion we caused in the last posting, but we don't want to give up returning our work to the mainline. **We cordially request LSM changes to pass vfsmount parameters.**
・Stephen Smalley kindly responded on the list

—“Don't cordially request it - submit patches to make it happen. Or work with others who have been submitting such patches. “
DON’T
Cordially ask to merge your code ;-)

Make good patches and post them
DON’T
Cordially ask to merge your code ;-)

Make good patches and post them
IN THE 16TH POSTING

• Tetsuo posted 25 series of messages with the subject, “Subject: [TOMOYO #16 00/25] Starting TOMOYO 2.3”

• The patches included enhancements as well as garbage collector functionality (so Tetsuo had a reason for 25)
• Pavel Machek commented:

• You are expected to submit diffs in smaller steps, not "here it is, totally rewritten, take it or leave it".
• Serge E. Hallyn once gave us a same comment:

• First let me point out that reviewing patches is always a lot of work. What you've done here by posting an entirely new 30-patch implementation of tomoyo when (I hope) you're not even serious about that is to basically tell us our time means nothing to you...
• Our mistakes, presented in this slides, are merely the tip of the iceberg

• We made many failures and we sometimes behaved very badly

• Nevertheless, this Linux community reacted and even merged our code
• Feedbacks were not limited by mailing list replies

• There were people who sent off-list messages and left advices in face
WITH ALL MY EXPERIENCES
I CAN SAY
WITH ALL MY EXPERIENCES
I CAN SAY

• Linux is not just free code
WITH ALL MY EXPERIENCES
I CAN SAY

• Linux is not just free code
• Linux is great because people are great
WITH ALL MY EXPERIENCES
I CAN SAY

• Linux is not just free code

• Linux is great because people are great

• Sending your code is a conversation with people
WITH ALL MY EXPERIENCES I CAN SAY

• Linux is not just free code
  • Linux is great because people are great
  • Sending your code is a conversation with people
  • They might not appear friendly for the first time, but try to speak them first
ACKNOWLEDGMENTS

Thank you for your participation in the 1st Annual LinuxCon- we couldn’t have done it without you!

The Linux Foundation
Al Viro, Albert Cahalan, Alexey Dobriyan, Andi Kleen, Andrew Morton, Bodo Eggert, Casey Schaufler, Chris Wright, Christoph Hellwig, Crispin Cowan, Daniel Walker, David Howells, David Lang, David P. Quigley, Greg KH, James Morris, Jamie Lokier, Jiri Kosina, Jonathan Corbet, Joshua Brindle, KaiGai Kohei, Kamezawa Hiroyuki, KOSAKI Motohiro, Kyle Moffett, Linus Torvalds, Matthew Wilcox, Miklos Szeredi, Paul E. McKenney, Paul Moore, Pavel Machek, Peter Dolding, Peter Zijlstra, Rik van Riel, Serge E. Hallyn, Seth Arnorld, Shaya Potter, Stephen Smalley, Tim Bird, Trond Myklebust, Valdis Kletnieks, William Leibzon, YOSHIFUJI Hideaki
WE COULDN’T HAVE DONE IT WITHOUT YOUR HELP

Al Viro, Albert Cahalan, Alexey Dobriyan, Andi Kleen, Andrew Morton, Bodo Eggert, Casey Schaufler, Chris Wright, Christoph Hellwig, Crispin Cowan, Daniel Walker, David Howells, David Lang, David P. Quigley, Greg KH, James Morris, Jamie Lokier, Jiri Kosina, Jonathan Corbet, Joshua Brindle, KaiGai Kohei, Kamezawa Hiroyuki, KOSAKI Motohiro, Kyle Moffett, Linus Torvalds, Matthew Wilcox, Miklos Szeredi, Paul E. McKenney, Paul Moore, Pavel Machek, Peter Dolding, Peter Zijlstra, Rik van Riel, Serge E. Hallyn, Seth Arnorld, Shaya Potter, Stephen Smalley, Tim Bird, Trond Myklebust, Valdis Kletnieks, William Leibzon, YOSHIFUJI Hideaki
AND OF COURSE

• We thank NTT DATA CORPORATION for their support of the project
Realities of Mainlining
- Case of the TOMOYO Linux Project -

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

NTT DATA CORPORATION
July 9, 2008

Time to Glean
MAC for Linux, history and the future

July 25, 2008

Toshiharu Harada
<haradats@nttdata.co.jp>
Kentaro Takeda
Tetsuo Handa

NTT DATA CORPORATION
What does it mean being an Open Source project manager in Enterprise Edition

LinuxCon 2009 (Business)
September 23, 2009
Toshiharu Harada
<haradats@nttdata.co.jp>
NTT DATA CORPORATION

What does it mean being an Open Source project manager in Enterprise Edition

LinuxCon 2009 (Business)
September 23, 2009
Toshiharu Harada
<haradats@gmail.com>
TOMOYO Linux Project
TRADEMARKS

• Linux is a registered trademark of Linus Torvalds in Japan and other countries

• TOMOYO is a registered trademark of NTT DATA CORPORATION in Japan