

CE Linux Forum
Worldwide Embedded Linux Conference 2007

TOMOYO Linux – Tutorial session

Kei Masumoto and Kentaro Takeda



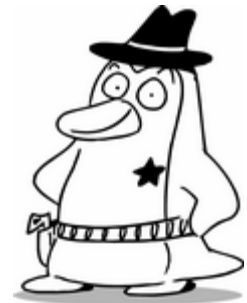
NTT DATA CORPORATION

<http://www.nttdata.co.jp/en/index.html>



TOMOYO Linux Project

<http://tomoyo.sourceforge.jp/>



Contents

- Background knowledge of TOMOYO Linux
- Part 1 – How to use “Automatic policy generation” functionality.
- Part 2 – How to protect Internet servers using TOMOYO Linux.
- Part 3 – Advanced TOMOYO Linux functionality.
 - Network access control, etc..

Background Knowledge

- Main points.
 - 1. Operation mode of TOMOYO Linux.
 - Disabled → Generating → Permissive → Enforcing.
 - Details will be shown in part 1.
 - 2. How to read TOMOYO Linux access policy.
 - Similar to UNIX-OS permission expression.

How to read the policy of TOMOYO Linux

| | | | |
|---------------------------|---|------------------------|--|
| domain | <code><kernel> /sbin/mingetty /bin/login</code> | | |
| access control list | <code>1 /bin/bash</code> | <code># 1 = --x</code> | |
| | <code>4 /etc/passwd</code> | <code># 4 = r--</code> | |
| | <code>4 /etc/shadow</code> | <code># 4 = r--</code> | |
| | <code>6 /var/log/lastlog</code> | <code># 6 = rw-</code> | |
| | <code>2 /var/run/wtmp</code> | <code># 2 = -w-</code> | |

- **/bin/login** executed by **/sbin/mingetty** can access only following resources
 - executing **/bin/bash** (domain transition)
 - reading **/etc/passwd**, **/etc/shadow**
 - reading/writing to **/var/log/lastlog**
 - writing to **/var/run/wtmp**

1. Policy generation and enforce

- Main points.
 - The “Automatic policy generation” functionality is the essential of TOMOYO Linux.
 - Users can use that functionality **only executing a few commands.**
 - In enforcing mode, an user can execute **only operations executed in policy generation mode.**

1. Policy generation and enforce

GENERATING

```

[root@tomoyo ~ DISABLED ]# setprofile -r 1 '<kernel> /usr/sbin/sshd /bin/bash' > /dev/null
[root@tomoyo ~ GENERATING]# head -3 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[root@tomoyo ~ GENERATING]# bash
[root@tomoyo ~ GENERATING]## tail -3 /etc/passwd
sshd:x:101:65534:./var/run/sshd:/bin/false
canna:x:104:104:Canna server,,,:/var/lib/canna:/bin/false
bind:x:103:105:./var/cache/bind:/bin/false
[root@tomoyo ~ GENERATING]## exit
exit
[root@tomoyo ~ GENERATING]# setprofile -r 3 '<kernel> /usr/sbin/sshd /bin/bash' > /dev/null
[root@tomoyo ~ ENFORCING]# head -3 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[root@tomoyo ~ ENFORCING]# ls
-bash: /busybox/bin/ls: Operation not permitted
[root@tomoyo ~ ENFORCING]# shutdown -h now
-bash: /sbin/shutdown: Operation not permitted
[root@tomoyo ~ ENFORCING]# rm -rf /
-bash: /busybox/bin/rm: Operation not permitted
[root@tomoyo ~ ENFORCING]# head -3 /etc/mtab
head: /etc/mtab: Operation not permitted
[root@tomoyo ~ ENFORCING]# tail -3 /etc/mtab
-bash: /busybox/usr/bin/tail: Operation not permitted
[root@tomoyo ~ ENFORCING]# bash
[root@tomoyo ~ ENFORCING]## tail -3 /etc/mtab
tail: /etc/mtab: Operation not permitted
[root@tomoyo ~ ENFORCING]## head -3 /etc/passwd
bash: /busybox/usr/bin/head: Operation not permitted
[root@tomoyo ~ ENFORCING]## exit
exit
[root@tomoyo ~ ENFORCING]# █
  
```

ENFORCING

only operations
executed in policy
generation mode
are permitted

1. Policy generation and enforce

```
<kernel> /usr/sbin/sshd /bin/bash
```

```
1 /busybox/usr/bin/head
```

```
1 /bin/bash
```

```
<kernel> /usr/sbin/sshd /bin/bash /busybox/usr/bin/head
```

```
4 /etc/passwd
```

```
<kernel> /usr/sbin/sshd /bin/bash /bin/bash
```

```
1 /busybox/usr/bin/tail
```

```
<kernel> /usr/sbin/sshd /bin/bash /bin/bash /busybox/usr/bin/tail
```

```
4 /etc/mtab
```

- These policies are automatically generated by operations.
- Each execution invokes a domain transition.
 - Even if one uses BusyBox, TOMOYO Linux can separate domains.

2. Apache policy

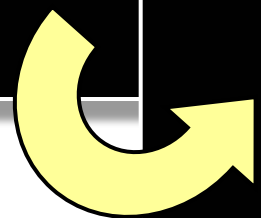
- Main points
 - The main reason to use TOMOYO Linux is to protect Internet servers from being cracked.
 - This tutorial shows **how TOMOYO Linux protects Internet servers from being cracked.**
 - Also, **users can use “patterns”** when the server accesses various resources and access policy is complex.

2. Apache policy

```

Armadillo
<<< Domain Policy Editor >>>      22 entries      '?' for help

<kernel> /busybox/usr/sbin/httpd
0: 4 /var/www/en/1.4/1st-step/centos4.4/editpolicy1.png
1: 4 /var/www/en/1.4/1st-step/centos4.4/editpolicy2.png
2: 4 /var/www/en/1.4/1st-step/centos4.4/editpolicy3.png
3: 4 /var/www/en/1.4/1st-step/centos4.4/grub.png
4: 4 /var/www/en/1.4/1st-step/centos4.4/index.html
5: 4 /var/www/en/1.4/1st-step/centos4.4/init.png
6: 4 /var/www/en/1.4/1st-step/centos4.4/login.png
7: 4 /var/www/en/1.4/1st-step/centos4.4/
8: 4 /var/www/en/1.4/1st-step/centos4.4/
9: 4 /var/www/en/1.4/1st-step/fc6/editp
10: 4 /var/www/en/1.4/1st-step/fc6/editp
11: 4 /var/www/en/1.4/1st-step/fc6/editp
12: 4 /var/www/en/1.4/1st-step/fc6/grub.p
13: 4 /var/www/en/1.4/1st-step/fc6/index
14: 4 /var/www/en/1.4/1st-step/fc6/init.p
15: 4 /var/www/en/1.4/1st-step/fc6/login
16: 4 /var/www/en/1.4/1st-step/fc6/ope_a
17: 4 /var/www/en/1.4/1st-step/fc6/ope_e
18: 4 /var/www/en/1.4/1st-step/startup.c
19: 4 /var/www/index.html
20: 4 /var/www/tomoyo.css
21: 4 /var/www/tomoyo.png
  
```



```

Armadillo
<<< Domain Policy Editor >>>      5 entries      '?' for help

<kernel> /busybox/usr/sbin/httpd
0: 4 /var/www/*
1: 4 /var/www/*/*
2: 4 /var/www/*/*/*
3: 4 /var/www/*/*/*/*
4: 4 /var/www/*/*/*/*/*
  
```

2. Apache policy

- Read accesses to the web contents

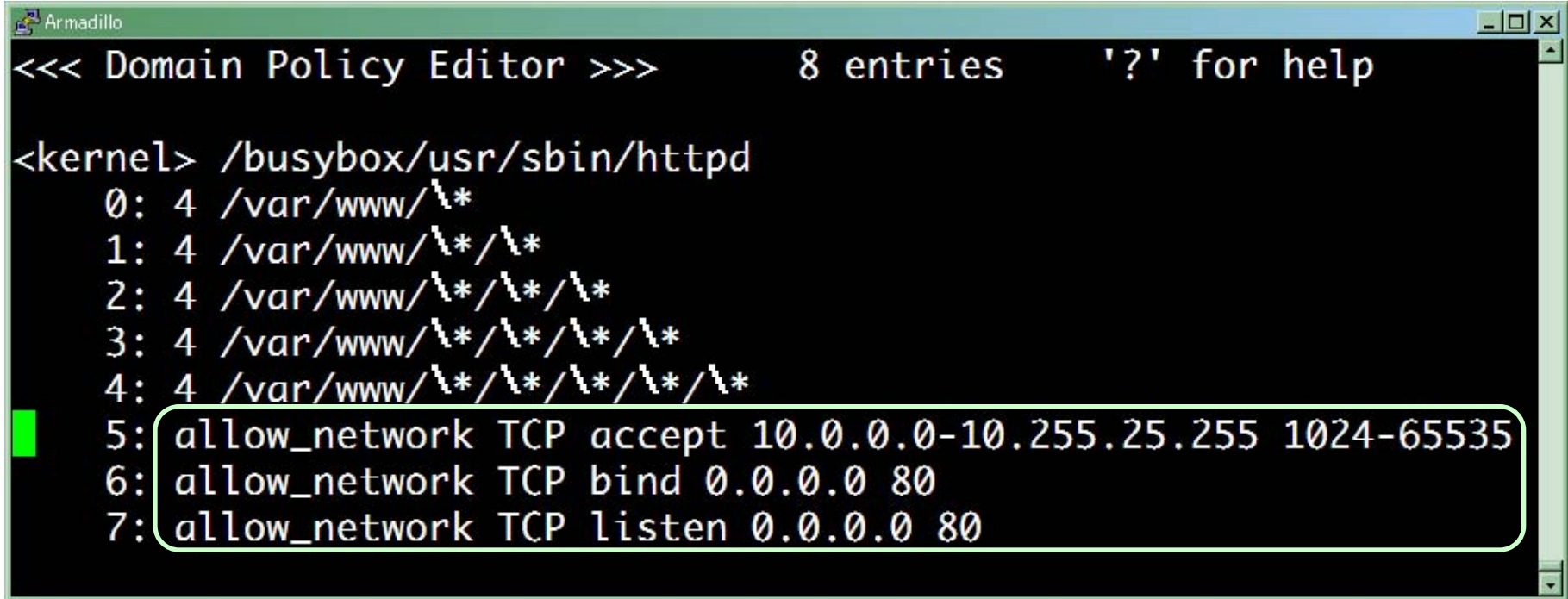
```
4 /var/www/¥*  
4 /var/www/¥*/¥*  
4 /var/www/¥*/¥*/¥*  
4 /var/www/¥*/¥*/¥*/¥*  
4 /var/www/¥*/¥*/¥*/¥*/¥*
```

- Some of the available patterns
 - **¥*** : any letters which do not contain '/'
 - **¥?** : any single letter except for '/'
 - **¥\$** : any decimal number
 - **¥X** : any hexadecimal number

3. Towards higher security

- Main points.
 - TOMOYO Linux can control not only files but also various resources.
 - “Network access control” is a major functionality towards higher security.
 - Users can use this functionality in the same way
 - using “change to advanced generation mode”

3. Towards higher security



```
<<< Domain Policy Editor >>>      8 entries      '?' for help

<kernel> /usr/sbin/httpd
0: 4 /var/www/*
1: 4 /var/www/*/*
2: 4 /var/www/*/*/*
3: 4 /var/www/*/*/*/*
4: 4 /var/www/*/*/*/*/*
5: allow_network TCP accept 10.0.0.0-10.255.25.255 1024-65535
6: allow_network TCP bind 0.0.0.0 80
7: allow_network TCP listen 0.0.0.0 80
```

3. Towards higher security

- Network access control

```
allow_network  
    TCP accept           : protocol  
    10.0.0.0-10.255.255.255 : IP address range  
    1024-65535           : port range
```

- Similar to iptables for each domain
- Other controllable resources
 - capability, signal, argv0
 - mount/umount, chroot, pivot root

Thank you for your attention

Any Questions?

