TOMOYO Linux

TOMOYO Linux (http://tomoyo.sourceforge.jp/) is Mandatory Access Control (MAC) implementation for Linux. It was launched in March 2003 and is sponsored by NTT Data Corporation (http://www.nttdata.co.jp/en/). TOMOYO Linux focuses on the behaviour of a system, allowing each process to declare behaviours and resources needed to achieve its purpose. It can be used as a system analysis tool as well as an access restriction tool.

The security goal of TOMOYO Linux is to provide "MAC that covers practical requirements for most users and keeps usable for most administrators". TOMOYO Linux is not a tool for just security professionals, but also for average users and administrators.

Note: This article does not aim to be an exhaustive guide and should be used as a supplement to the extensive user documentation (http://tomoyo.sourceforge.jp/documentation.html) provided by the project.

Tip: The TOMOYO Linux 2.x will eventually come closer to reaching feature parity with the 1.x branch, but for those wanting an easy start the 2.x branch is easy to install. The TOMOYO Linux 1.x branch is for those wanting the greatest security, while AKARI is somewhere in between.

Contents

Introduction

TOMOYO Linux attempts to make the system where everything is prearranged in an easy to understand way:

- Make all access requests that will occur at least once during the lifetime of the kernel known in advance
- Allow the administrator to write a policy that only allows expected and desirable access requests

Unlike AppArmor, TOMOYO Linux is intended to protect the whole system from attackers exploiting vulnerabilities in applications. TOMOYO Linux addresses this threat by recording the behaviour of all applications in the test environment and then forcing all applications to act within these recorded behaviours in the production environment.

TOMOYO Linux is not for users wanting ready-made policy files supplied by others. It involves creating policy from scratch, aided by the "learning mode" which can automatically generate policy files with necessary and sufficient permissions for a specific system. TOMOYO Linux reports what is happening within the Linux system and can therefore be used as a system analysis tool. It resembles strace and reports what is being executed by each program and what files/networks are accessed.

This **table (http://tomoyo.sourceforge.jp/wiki-e/?WhatIs#comparison)** provides a comprehensive comparison of TOMOYO Linux with **AppArmor**, **SELinux** and **SMACK (http://schaufler-ca.com/)**.

Branches of development

TOMOYO Linux 1.x (http://tomoyo.sourceforge.jp/1.8/index.html.en) is the original branch of development. TOMOYO Linux was first released on 11th November 2005. It was implemented as a patch that can be applied to the Linux kernel and is still in active development. It can coexist with other security modules such as SELinux, SMACK and AppArmor.

TOMOYO Linux 2.x (http://tomoyo.sourceforge.jp/2.3/index.html.en) is the Linux mainline kernel branch of development. In June 2009, TOMOYO was merged into the Linux kernel version 2.6.30 and it uses standard Linux Security Module (LSM) hooks. However, the LSM hooks must be extended further in order to port the full MAC functionality of TOMOYO Linux into the Linux kernel. Thus, it does not yet provide equal functionality with the 1.x branch of development. This **chart (http://tomoyo.sourceforge.jp/comparison.html.en)** compares the differences between each branch.

AKARI (http://akari.sourceforge.jp/) is based on the TOMOYO Linux 1.x branch and is implemented as a Loadable Kernel Module (LKM). It therefore has the advantage of not requiring the user to patch and recompile the kernel. This **table (http://akari.sourceforge.jp/comparison.html)** provides a comprehensive comparison of AKARI with the TOMOYO Linux 1.x and 2.x branches.

TOMOYO Linux 1.x

Implementing TOMOYO Linux 1.x using a kernel patched with ccs-patch provides the full functionality obtainable from the TOMOYO Linux project. However, implementation of this branch requires the most hurdles to be overcome, as the kernel must be patched with **ccs-patch (http://sourceforge.jp/projects/tomoyo/)** and subsequently recompiled.

Both *linux-ccs* and the userspace tools must be installed. A package for **linux-ccs (https://aur.archlinux.org/packages.php?ID=51669)** and a package for **ccs-tools (https://aur.archlinux.org/packages.php?ID=42606)** are available on the AUR.

Initializing configuration

The policy must first be initialized:

```
# /usr/lib/ccs/init_policy
```

The policy files are saved in the `/etc/css/` directory and can be edited by running:

```
# ccs-editpolicy
```

AKARI

Limitations of AKARI

AKARI has the advantage of not requiring kernel recompilation. If using the TOMOYO Linux project purely for system analysis, then AKARI is the easiest method of achieving this. If using the TOMOYO Linux project for system restriction, it is a minimal effort way to gain most of the functionality of the TOMOYO Linux 1.x branch. However, there are a few limitations that must be considered:

- It depends on the kernel version and configuration provided by the distribution:

```
CONFIG_SECURITY=y [required]
CONFIG_KALLSYMS=y [required]
CONFIG_PROC_FS=y [required]
CONFIG_MODULES=y [required]
CONFIG_SECURITY_PATH=y [optional: for using absolute pathnames]
CONFIG_SECURITY_NETWORK=y [optional: for providing network restriction]
```

- The restriction of a few advanced networking operations are limited or unavailable due to the absence of required LSM hooks
- Restricting use of **capabilities** is not possible
- Looking up per-task variables is slower as they are managed outside "struct task_struct" in order to keep KABI unchanged. However, this should not be noticeable for the typical end-user as performance decrease by pathname based permission checking is dominant

This **table (http://akari.sourceforge.jp/comparison.html)** provides a comprehensive comparison of AKARI with the TOMOYO Linux 1.x and 2.x branches.

Note: The Arch Linux kernel from 2.6.36 onwards provides all of the configuration options required for full functionality.

Installation

Both AKARI and the userspace tools must be installed. A package for `akari (https://aur.archlinux.org/packages/akari/)` AUR and a package for `ccs-tools (https://aur.archlinux.org/packages/ccs-tools/)` AUR are available on the AUR.

The bootloader configuration must be changed in order to activate AKARI:

```
title  Arch Linux
root   (hd0,0)
kernel /boot/vmlinuz-linux root=/dev/sda1 ro init=/sbin/ccs-init
initrd /boot/initramfs-linux.img
```

Initializing configuration

The policy must first be initialized:

```
# /usr/lib/ccs/init_policy --module_name=akari
```

The policy files are saved in the `/etc/css/` directory and can be edited by running:

```
# ccs-editpolicy
```

TOMOYO Linux 2.x

Limitations of TOMOYO Linux 2.x

The implementation of TOMOYO Linux 2.x into the Linux mainline kernel is not yet complete but is very close to 1.x since 2.5.x. There are a few features that still need to be implemented as compared to the 1.x branch. This chart (http://tomoyo.sourceforge.jp/comparison.html.en) has a comprehensive comparison of the differences between each branch of development.

Installation

TOMOYO Linux 2.x is part of the Linux mainline kernel and requires the following kernel configuration:

```
CONFIG_SECURITY=y
CONFIG_SECURITYFS=y
CONFIG_SECURITY_NETWORK=y [disabled in the Arch Linux kernel]
CONFIG_SECURITY_PATH=y
CONFIG_SECURITY_TOMOYO=y [disabled in the Arch Linux kernel]
```

Note: FS#42910 (https://bugs.archlinux.org/task/42910) has been opened requesting that TOMOYO be enabled in the linux (https://www.archlinux.org/packages/?name=linux) package again, and the linux-lts (https://www.archlinux.org/packages/?name=linux-lts) package would follow that lead.

If the kernel supports TOMOYO Linux 2.x, then only the userspace tools (from AUR `tomoyo-tools` (https://aur.archlinux.org/packages/tomoyo-tools/)AUR) need to be installed.

For kernel versions between 2.6.30 and 2.6.35, tomoyo-tools 2.2.x should be installed. A package is available on the **AUR (https://aur.archlinux.org/packages.php?ID=42272)**

Activation

If all ok, append **security=tomoyo TOMOYO_trigger=/usr/lib/systemd/systemd** to parameter GRUB_CMDLINE_LINUX_DEFAULT in `/etc/default/grub`:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet security=tomoyo TOMOYO_trigger=/usr/lib/systemd/systemd"
```

After, recompile `grub.cfg`:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

So, TOMOYO will load all saved policies from `/etc/tomoyo/policy/current` when `/usr/lib/systemd/systemd` executes.

Next, check whether the activation was successful. You should have the following lines (or similar) in your dmesg output:

```
$ dmesg |grep -A 1 -B 1 TOMOYO
[    0.003375] Security Framework initialized
[    0.003387] TOMOYO Linux initialized
[    0.003396] AppArmor: AppArmor disabled by boot time parameter
--
[    6.829798] Calling /usr/bin/tomoyo-init to load policy. Please wait.
[    6.833709] TOMOYO: 2.5.0
[    6.833712] Mandatory Access Control activated.
```

For first time, you may want to auto-save in-memory policies to filesystem when computer goes to shutdown/reboot. If yes, write `/usr/lib/systemd/system/tomoyo-savepolicy.service` script:

```
/usr/lib/systemd/system/tomoyo-savepolicy.service
```

```
[Unit]
Description=Tomoyo savepolicy

[Service]
Type=oneshot
ExecStart=/bin/true
ExecStop=/usr/bin/tomoyo-savepolicy
StandardInput=tty
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target
```

You can enable/disable it with systemctl:

```
# systemctl enable tomoyo-savepolicy.service
```

Initializing configuration

The policy must first be initialized:

```
# /usr/lib/tomoyo/init_policy
```

The policy files are saved in the `/etc/tomoyo/` directory and can be edited by running:

```
# tomoyo-editpolicy
```

By default, tomoyo will start with "Disabled" profile (see profile-table below). You may want to enable learning mode for everybody right now. Just switch profile for `<kernel>` namespace in `/etc/tomoyo/policy/current/domain_policy.conf`:

```
<kernel>
use_profile 1
use_group 0
```

If unsure if such wide learning is needed, just ignore this step. You can switch profiles later using **tomoyo-editpolicy** in "Domain transition editor" by pressing **S** on any selected domain (domains).

Now, the computer should be restarted.

Log daemon

For tomoyo exists the log-daemon `/usr/sbin/tomoyo-auditd`. It is useful for monitoring the behavior of closed-source applications. The initial configuration file is well explained and can be found in `/etc/tomoyo/tools/auditd.conf` whereas the log files can be found in `/var/log/tomoyo`.

To use it with systemd create the file `/lib/systemd/system/tomoyo-auditd.service` with the content described in chapter 4.6 in the official **documentation (http://tomoyo.sourceforge.jp/2.5/chapter-4.html.en)**.

Usage

It is important to consult the relevant documentation in order to use TOMOYO Linux or AKARI effectively:

- **TOMOYO Linux documentation (http://tomoyo.sourceforge.jp/documentation.html.en)**
- **AKARI documentation (http://akari.sourceforge.jp/index.html.en)**

Run the policy editor to begin editing. If using TOMOYO Linux 1.x or AKARI, then *ccs-tools* should be used:

```
# /usr/sbin/ccs-editpolicy
```

If using TOMOYO Linux 2.x, then *tomoyo-tools* should be used:

```
# /usr/sbin/tomoyo-editpolicy
```

As the system runs, TOMOYO Linux will create domains and add them to the tree. The access analysis/restriction in TOMOYO Linux is applied via domains. Every process belongs to a single domain and the process will transit to a different domain whenever it executes a program. The name of a domain is a concatenated string expression for the process execution history. For example, the name of the domain which the kernel belongs to is "<kernel>"; the name of domain which `/sbin/init` invoked by the kernel belongs to is "<kernel> /sbin/init"; if `/sbin/init` invokes `/etc/rc.d/rc` then the domain it belongs to is " <kernel> /sbin/init /etc/rc.d/rc". You can suppress or initialize domain transitions as needed.

Profiles can be assigned to each domain. There are four default profiles:

Disabled	Works as if regular kernel.
Learning	Do not reject an access request if it violates policy. Append the request to policy.
Permissive	Do not reject an access request if it violates policy. Do not append the request to policy.
Enforcing	Reject an access request if it violates policy. Do not append the request to policy.

The learning profile can be used to analyse the system or a specific application. Once all of the desired access requests of a domain have been identified, the policy for that domain can be edited as required before selecting the enforcing profile. This can be done for any and all domains from the start of system boot.

References

- TOMOYO Linux Home Page (http://tomoyo.osdn.jp/)
- TOMOYO Linux Wiki (http://tomoyo.osdn.jp/wiki-e/)
- AKARI Home Page (http://akari.osdn.jp/index.html.en)
- AKARI Ddocumentation (http://akari.osdn.jp/documentation.html.en)
- AKARI/TOMOYO functionality comparison table (http://akari.osdn.jp/comparison.html)
- TOMOYO Linux 1.8.x : The Official Guide (http://tomoyo.osdn.jp/1.8/index.html.en)
- TOMOYO Linux 2.5.x : The Official Guide (http://tomoyo.osdn.jp/2.5/index.html.en)
- TOMOYO Linux Security Goal (http://lwn.net/Articles/263179/)
- Policy sample (http://tomoyo.sourceforge.jp/cgi-bin/lxr/source/centos5.5/domain_policy.conf?v=policy-sample)
- TOMOYO Linux on the Embedded Linux Wiki (http://elinux.org/TomoyoLinux)
- Presentation slides from PacSec 2007 (https://osdn.net/projects/tomoyo/docs/PacSec2007-en-demo.pdf)

See also

- AppArmor
- SELinux

Retrieved from "https://wiki.archlinux.org/index.php?title=TOMOYO_Linux&oldid=506851"

- This page was last edited on 11 January 2018, at 11:52.
-