Test of FreeBSD MAC Framework

Zhouyi Zhou (zhouzhouyi at gmail.com) [1]

March 23, 2008

[1] Directed by Robert N. M. Watson (robert.watson@cl.cam.ac.uk)

1. Introduction
   - Introduction to MAC Framework
2. MAC Framework Test
   - Static Analysis of MAC Framework
   - Runtime Regrssion Test of MAC Framework
3. The End

Why MAC Framework

Besides of patching the operating system against security holes, adding access control extensions to operating system has been an active way to guarantee the security of systems.
We need a generic Framework of kernel authorization hooks because most of access control extensions fall short two vital areas [2]

- Lack of support by the operating system vendor for various security extension providers.

[2] Robert Watson etc. The TrustedBSD MAC Framework: Extensible Kernel Access Control for FreeBSD 5.0. FREENIX 2003

Why MAC Framework

Besides of patching the operating system against security holes, adding access control extensions to operating system has been an active way to guarantee the security of systems.
We need a generic Framework of kernel authorization hooks because most of access control extensions fall short two vital areas [2]

- Lack of support by the operating system vendor for various security extension providers.
- Highly redundant implementation of support infrastructure for security extension providers.

[2] Robert Watson etc. The TrustedBSD MAC Framework: Extensible Kernel Access Control for FreeBSD 5.0. FREENIX 2003

MAC Framework for FreeBSD

Support for Mandatory Access Control (MAC) was introduced into the FreeBSD operating system as of FreeBSD 5.0. FreeBSD MAC Framework provides a set of generic authorization hooks that are inserted into the kernel source that enable individual security modules to enforce their access control policy.

MAC Framework for FreeBSD

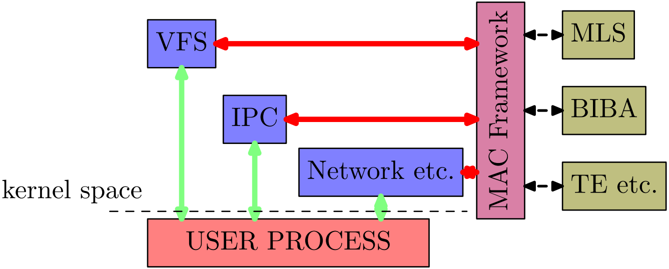


Figure: MAC Framework

MAC Framework Test

- Tests categorized by Objectivity
  - Checking: discover the MAC Framework vulnerabilities
  - Regression Test: prevent the already discovered bugs reappear.

MAC Framework Test

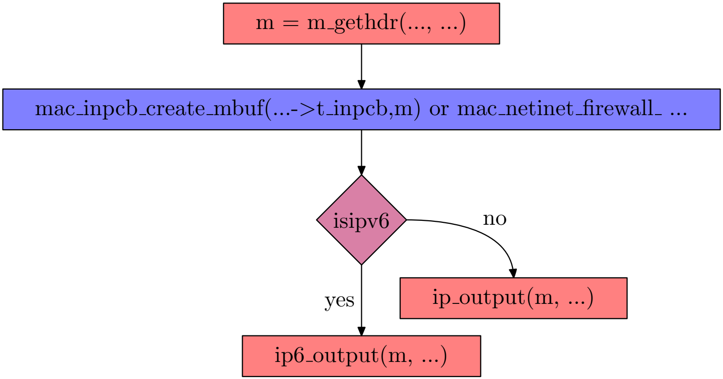
- Tests categorized by Objectivity
  - Checking: discover the MAC Framework vulnerabilities
  - Regression Test: prevent the already discovered bugs reappear.
- Tests categorized by test time
  - Runtime Test
  - Static Analysis

Static Analysis Overview

- Static Analysis can be seen as an “advanced” grep ☺
- Static Analysis is complete: no dark corners

Static Analysis Objectivity

- Complete Initialization

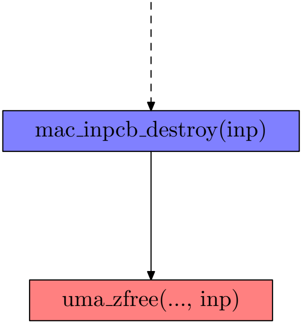


Failing of complete initialization will lead to kernel crash [3]

[3] pf.c revision 1.34.2.3

Static Analysis Objectivity

- Complete Destruction



mac_inpcb_destroy(inp)

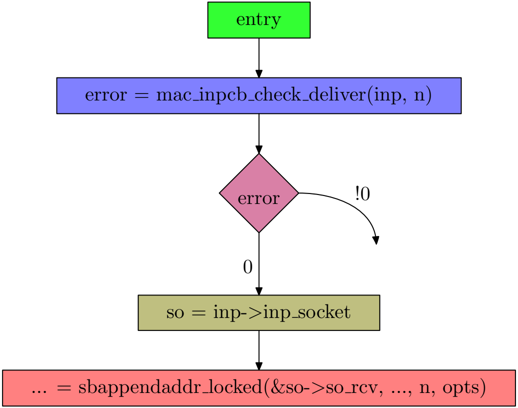
uma_zfree(..., inp)

Failing of complete destruction will lead to memory leakage [4]

[4] in_pcb.c revision 1.197

Static Analysis Objectivity

- Complete Authorization


entry
error = mac_inpcb_check_deliver(inp, n)
error
!0
0
so = inp->inp_socket
... = sbappendaddr_locked(&so->so_rcv, ..., n, opts)

Failing of complete authorization will lead to privilege leakages
[5]

[5] NFS and fifo

Static Analysis

- Use modified version of MyGCC
- Has been submitted to USENIX Security 2008 [6]

[6] Z Zhou, R Watson, Y He, H Liang and C Peron. Permanent Checking on the FreeBSD MAC Framework Using Mygcc

Runtime Regression Test

- Provide a end-to-end Security Guarantee from user space label mechanism to kernel policy module arbitration.
- Easy to run and understand.

Regression Test Framework

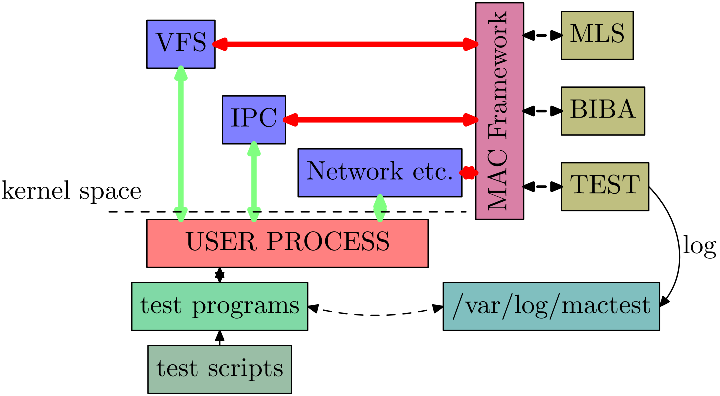


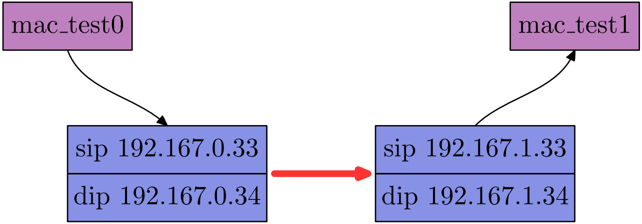
Figure: Log and Compare the invoked hooks

Regression Test Framework

- In the kernel, every non-null label is externalized into human readable string and recorded in a tail queue together with the name of hook that got invoked and optional flags or modes. There is a thread much like audit subsystem's audit_worker logging the queue into a userspace file. The logging file is truncated to zero every time the logging mechanism is retriggered.
- In userspace, a bison based parsing tool is used to parse the logged file and reconstruct the record chain which will be compared with testsuite supplied configuration file to examine if expected hooks is got invoked and the labels/flags/modes are correct.

Regression Test Framework

- Have constructed a pair of pseudo-ethernet drivers used for network interface related tests. To avoid the packet go through the lo interface, the IP address in the packet is twisted in the driver. The idea is inspired by [7]



[7] Jonathan Corbet, Allesandro Rubini, and Greg Kroah-Hartman. Linux. Device Drivers 3rd Edition. OReilly, 2005.

Regression Test

- The system is well covered: File System, Process Control, IPC, Network .etc

Thank you