LinuxCon Japan 2014 (Tokyo)
2014/5/20-2014/5/22

How to obtain information for troubleshooting enterprise servers

Tetsuo Handa

Who am I?

- A retired employed programmer who is in a position of responding to various troubles related to Linux kernels.

- My involvement with Linux
  - 2001.10-2003.3 Developing user space applications that run on Linux systems.
  - 2003.4-2012.3 Developing kernel mechanisms for improving security of Linux systems.
  - 2012.4-2013.3 Providing user support service for troubleshooting Linux systems.
  - 2013.4- Providing technical support service for troubleshooting Linux systems.

What do I speak today?

- When analyzing the cause of unexpected behavior such as kernel panic, system freeze, system reboot, service failover and unidentified execution of programs, it is important that the administrator knows how to obtain information in advance.

- Today I speak about my experience, utilities and methodology I learned or developed for obtaining information. Main targets are RHEL 5 and RHEL 6 servers running on x86_64 architecture but not limited to. This presentation is meant for understanding what you can prepare in advance for smarter management.

Why do I speak it today?

- Because I'm frequently seeing regrettable cases where information for analyzing was not recorded because the administrator did not know how to record it.
  - I can offer ideas for troubleshooting, but I don't have problematic hardware/software to troubleshoot.
  - Customers have problematic hardware/software to troubleshoot, but I don't have chances to introduce my ideas to customers.

- I hope that this presentation material helps troubleshooting problematic hardware/software.
  - Feedbacks are welcome.

Chapter 1

Preparation for kernel space problems.

Index of "Preparation for kernel space problems."

What to do with unexpected kernel panic?

- Set up kdump in advance and get the memory image as of kernel panic.
  - The kdump may fail due to hardware problem.
    - BIOS/firmware on the latest hardware may not be sufficiently tested.
    - Older versions of VMware products have a bug which fails to kdump RHEL 6 guests.
  - The "crashkernel=auto" command line parameter may not work.
    - Some device drivers require a lot of memory, which results in kdump failure due to OOM killer while dumping the memory image.
    - Check the amount of memory reserved for kdump.

What to do with unexpected system freeze?

- Set up SysRq functionality in advance if you can trust console users.
  - Set /proc/sys/kernel/sysrq to 1 using /etc/sysctl.conf .
  - The "sysrq_always_enabled" command line parameter is available for Linux 2.6.20 and later kernels.
- Unexpected system freeze may happen during the boot process.
  - The "quiet" and "rhgb" command line parameters should be removed unless you want to save a few seconds of the boot procedure.
- Default kernel log buffer size may not be large enough for holding SysRq-T messages.
  - Use of the "log_buf_len=" command line parameter will help.

What is OOM killer?

- A built-in feature which tries to survive out of memory condition by killing some processes.
  - Linux kernel is not carefully designed enough to completely avoid OOM killer deadlock.
    - CPU usage will become 100% and no more logs will be recorded to /var/log/messages if deadlock happened.
- Do not rely on the OOM killer too much.
  - There is a choice of setting /proc/sys/vm/panic_on_oom to 1 if you can trust login users.
- Do not abuse /proc/pid/oom_score_adj or /proc/pid/oom_adj .
  - Excluding enterprise application processes (e.g. java) from the OOM killer will make the system more prone to trigger kernel panic due to out of memory condition.

What to do with unexpected system reboot?

- One of most annoying troubles in Linux because it is difficult to understand the reason of rebooting.
  - The kernel panic allows kdump, but the reboot does not.
- The kernel may have printed crucial messages immediately before rebooting.
  - But examining /var/log/messages unlikely helps, for there is no time for syslog daemon to save such messages to /var/log/messages .
  - Setting up serial console or netconsole may help catching such messages.

What is serial console?

- A relatively reliable way to capture kernel messages.
  - Messages saved to /var/log/messages have time stamp, but messages sent to console may not have timing data.
    - It is difficult to judge relevance of trouble without knowing when the message was printed.
      - You can prefix timing data to each kernel message by setting /sys/module/printk/parameters/time (RHEL 6) or /sys/module/printk/parameters/printk_time (RHEL 5) to 1 from /etc/rc.local .
      - Although there is "printk.time=1" (RHEL 6) / "time" (RHEL 5) command line parameter, use of this parameter may cause boot failures due to kernel initialization ordering.

What is serial console?

- Some hardware support redirection of serial console.
  - Powerful enterprise servers likely provide access to serial console over TCP/IP networking instead of physical RS-232C cable. Please check your server's manual.
    - HP iLO remote serial console
    - DELL IPMI remote serial console
    - VMware / KVM etc.
  - External Serial-Ethernet converters are also available.
    - You might be able to forward kernel messages via TCP or UDP if serial adapter is available on your server.

What is netconsole?

- A handy way to capture kernel messages.
  - It would be useful for virtualized hosts where login to physical hosts are not allowed. Also, it would be useful for servers which are difficult to restart the system in order to enable/disable the serial console.
  - A network interface (e.g. eth0) and networking are required because netconsole forwards kernel messages via UDP.
    - The network interface is not available during boot up and initialization of the kdump kernel.
    - You might want to use external Serial-Ethernet converters rather than netconsole if you want to receive all kernel messages.

What is netconsole?

- A utility for saving kernel messages is available.
  - http://sourceforge.jp/projects/akari/scm/svn/tree/head/branches/udplogger/
    - It is difficult to add time stamp because kernel messages (especially SysRq messages) sent via UDP are not record oriented. This utility can buffer until a line completes and save with time stamp.
    - Also, this utility can run with small resource footprint and least dependency, has built-in log file switching, is robust against temporary resource failures (e.g. too many processes or open files / unexpected denial of program execution or file open) compared to shell scripts.

What to do with unexpected service fail over?

- The fail over will happen without any prior warning if the timeout of watchdog is shorter than timeout of kernel warning mechanisms.
  - Consider setting /proc/sys/kernel/hung_task_timeout_secs to very small value (e.g. 3) and /proc/sys/kernel/hung_task_warnings to very large value (e.g. 1000000).
    - Note that this hung check timer can warn only threads waiting in uninterruptible state.
    - Not seeing hung check timer warnings cannot guarantee that no thread hung up. Seeing hung check timer warnings cannot guarantee that some thread hung up.

What to do with unexpected service fail over?

- The cause of timeout can be within hardware drivers or hardware itself.
  - Serial ATA's Native Command Queuing can cause small disk I/O to delay for too long when large disk I/O is in progress.
  - Under virtualized environments, watch out for local disks because logically isolated disks (e.g. /dev/sda) seen from the guests may be sharing a physically shared disk on the host.
  - Check how you can access information which you don't have control.

What to do with unexpected service fail over?

- The antivirus software can cause watchdog applications to trigger.
  - Timed out due to on-access scanning taking too long.
  - EUNATCH error upon opening a regular file.
  - ETXTBUSY error upon executing a program.
  - By error deleting files under /tmp directory.
- Programs like shell scripts are vulnerable to this kind of disturbance.
  - Heartbeat/watchdog software need painstaking error checking and retry mechanisms.

What tools can we use for recording unexpected events?

- System call auditing
  - Can be used for recording failed system calls.
  - Would be sufficient if you need to know only whether errno was given by system call or not.
  - Would be insufficient if you need to know why specific errno was returned by system call.
- SystemTap
  - A powerful candidate for troubleshooting tool.
  - Various examples are available.
    - https://sourceware.org/systemtap/examples/
  - To demonstrate how handy SystemTap is, several examples which I wrote are embedded into this presentation material.

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.

- Obtaining SysRq-T messages might be helpful for analysis when stack trace of a hung thread is printed.
  - SystemTap could be used in conjunction with watchdog for automatically obtaining SysRq-T messages.

```
---------- program start ----------
# stap -e '
probe kernel.statement("*@kernel/hung_task.c:$N") {
  system("echo t > /proc/sysrq-trigger"); exit();
}'
---------- program end ----------
```

Note: $N needs to be replaced with the line number of "if (!sysctl_hung_task_warnings)" in check_hung_task(). For example, $N = 89 for kernel-2.6.32-431.11.2.el6 and $N = 87 for kernel-2.6.18-371.8.1.el5 .

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.

Innovative R&D by NTT

```
---------- linux-2.6.32-431.11.2.el6/kernel/hung_task.c ----------
72:static void check_hung_task(struct task_struct *t, unsigned long timeout)
73:{
74:     unsigned long switch_count = t->nvcsw + t->nivcsw;
(...snipped...)
84:
85:     if (switch_count != t->last_switch_count) {
86:             t->last_switch_count = switch_count;
87:             return;
88:     }
89:     if (!sysctl_hung_task_warnings)
90:             return;
91:     sysctl_hung_task_warnings--;
92:
(...snipped...)
112:}
 ---------- linux-2.6.32-431.11.2.el6/kernel/hung_task.c ----------
```

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.

```
---------- output example start ----------
[ 2293.846244] INFO: task dd:3713 blocked for more than 1 seconds.
[ 2293.846645]       Not tainted 2.6.32-431.11.2.el6.x86_64 #1
[ 2293.847035] "echo 0 > /proc/sys/kernel/hung_task_timeout_secs" disables
  this message.
[ 2293.847889] dd            D 0000000000000000     0  3713   3359
  0x00000080
[ 2293.848671]  ffff88003e1d3750 0000000000000086 0000000000000000
  0000000000000001
[ 2293.849582]  ffff8800022968a8 ffff8800022968d8 ffff8800022968a8
  ffff880039523578
[ 2293.850514]  ffff88003d99bab8 ffff88003e1d3fd8 000000000000fbc8
  ffff88003d99bab8
[ 2293.851462] Call Trace:
[ 2293.851784]  [<ffffffff8152a5a5>] rwsem_down_failed_common+0x95/0x1d0
[ 2293.852310]  [<ffffffff8152a736>] rwsem_down_read_failed+0x26/0x30
[ 2293.852722]  [<ffffffff8128ea84>] call_rwsem_down_read_failed+0x14/0x30
(...snipped...)
```

Messages by hung check timer is printed, and ...

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.

```
[ 2293.862983]  [<ffffffff8128ee71>] ? __clear_user+0x21/0x70
[ 2293.863380]  [<ffffffff81226496>] ? security_file_permission+0x16/0x20
[ 2293.863794]  [<ffffffff81189048>] vfs_write+0xb8/0x1a0
[ 2293.863975]  [<ffffffff81189941>] sys_write+0x51/0x90
[ 2293.863975]  [<ffffffff810e1e4e>] ? __audit_syscall_exit+0x25e/0x290
[ 2293.865101]  [<ffffffff8100b072>] system_call_fastpath+0x16/0x1b
[ 2294.379709] SysRq : Show State
[ 2294.380035]   task                        PC stack   pid father
[ 2294.380035] init          S 0000000000000000     0     1      0
  0x00000000
[ 2294.380035]  ffff88003f38f908 0000000000000086 0000000000000000
  ffffffff8109b27f
[ 2294.380035]  ffff88000eb40200 ffff88001b7c4ba8 0000000000000000
  ffff880010e8b3d8
[ 2294.380035]  ffff88003f38dab8 ffff88003f38ffd8 000000000000fbc8
  ffff88003f38dab8
[ 2294.380035] Call Trace:
[ 2294.380035]  [<ffffffff8109b27f>] ? wake_up_bit+0x2f/0x40
[ 2294.380035]  [<ffffffff8112d551>] ? get_page_from_freelist+0x3d1/0x870
(...snipped...)
```

messages by SysRq-T immediately follows.

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.


```
[ 2294.380035]   .load_period                  : 0.000000
[ 2294.380035]   .load_contrib                 : 0
[ 2294.380035]   .load_tg                      : 0
[ 2294.380035]
[ 2294.380035] rt_rq[1]:/
[ 2294.380035]   .rt_nr_running                : 0
[ 2294.380035]   .rt_throttled                 : 0
[ 2294.380035]   .rt_time                      : 0.000000
[ 2294.380035]   .rt_runtime                   : 900.000000
[ 2294.380035]
[ 2294.380035] runnable tasks:
[ 2294.380035]             task   PID         tree-key  switches  prio
   exec-runtime         sum-exec        sum-sleep
[ 2294.380035] ------------------------------------------------------------
   ----------------------------------------------
[ 2294.380035]
--------- output example end ----------
```

SystemTap Example 1 - Trigger SysRq-T upon hung check timeout.

- What is nice with this script?
  - We might be able to capture SysRq-T when the administrator is not ready to respond (e.g. failures at midnight).
  - Given that the hang up is not system-wide, a series of SysRq commands could be triggered automatically.

SystemTap Example 2 - Show threads terminated by signal.

- Find out which thread is exiting by signal.

```
---------- program start ----------
# stap -e '
probe kernel.function("do_exit") {
  if ($code & 0x7F)
    printf("%s %s(%u) exiting with signal %u¥n",
           ctime(gettimeofday_s()), execname(), pid(), $code & 0x7F);
}'
---------- program end ----------

---------- output example start ----------
Sat May  3 06:00:39 2014 a.out(2101) exiting with signal 11
Sat May  3 06:00:48 2014 sleep(2102) exiting with signal 2
Sat May  3 06:01:17 2014 sleep(2105) exiting with signal 9
Sat May  3 06:01:21 2014 a.out(2131) exiting with signal 11
---------- output example end ----------
```

SystemTap Example 2 - Show threads terminated by signal.

- What does this script differ from system call auditing?
  - This script can handle all cases because the hook is inserted into a common function which is called whenever a thread exits.
  - System call auditing cannot handle several cases (e.g. terminating /bin/sleep process executed from terminal by pressing Ctrl-C) because the hook is inserted into limited locations.

SystemTap Example 3 - Show disk I/O accounting upon thread exit.

- Find out which thread is causing heavy disk I/O requests.

```
---------- program start ----------
# stap -g -e '%{ #if !defined task_aux %} %{ #define task_aux(x)
   (x) %} %{ #endif %}
probe begin {
  printf("time                          ¥tpid¥ttid¥tppid¥tread¥twrite¥tcomm¥n");
}
probe kernel.function("do_exit") {
  read = %{ task_aux(current)->ioac.read_bytes %};
  write = %{ task_aux(current)->ioac.write_bytes %};
  if (read || write)
    printf("%s¥t%u¥t%u¥t%u¥t%u¥t%u¥t%s¥n", ctime(gettimeofday_s()),
           pid(), tid(), ppid(), read, write, execname());
}'
---------- program end ----------
```

SystemTap Example 3 - Show disk I/O accounting upon thread exit.


```
---------- output example start ----------
time                          pid    tid    ppid   read    write  comm
Sat May  3 06:47:37 2014      2679   2679   2678   28672   0      id
Sat May  3 06:47:37 2014      2678   2678   2677   24576   0      bash
Sat May  3 06:47:37 2014      2681   2681   2680   16384   0
  hostname
Sat May  3 06:47:37 2014      2683   2683   2682   24576   0      tty
Sat May  3 06:47:37 2014      2684   2684   2682   307200  0      tput
Sat May  3 06:47:37 2014      2682   2682   2677   8192    0      bash
Sat May  3 06:47:37 2014      2686   2686   2685   45056   0
  dircolors
Sat May  3 06:47:37 2014      2687   2687   2677   413696  0      grep
Sat May  3 06:47:37 2014      2689   2689   2688   8192    0
  consoletype
Sat May  3 06:47:43 2014      2677   2677   2676   352256  0      bash
---------- output example end ----------
```

SystemTap Example 3 - Show disk I/O accounting upon thread exit.

- What does this script differ from iotop/pidstat programs?
  - Event driven approach is suitable for catching short-living threads (e.g. which exits within a second) compared to interval timer approach.
  - Event driven sampling can save CPU usage compared to short interval sampling.

SystemTap Example 4 - Detect the dentry cache bouncing bomb.

- The unused dentries can grow in an uncontrolled way, which in turn results in huge delay to the level where fail over due to timeout can happen upon memory reclaim.

```
---------- program start ----------
# stap -e '
global stat_start;
global shrink_start;
global counter;

probe kernel.function("vfs_stat") {
  counter++;
  stat_start[tid()] = gettimeofday_ns();
}
```

SystemTap Example 4 - Detect the dentry cache bouncing bomb.

Innovative R&D by NTT

```
probe kernel.function("vfs_stat").return {
  t = gettimeofday_ns() - stat_start[tid()];
  if (t >= 1000000)
    printf("%s: vfs_stat(%u) took %u ms¥n", execname(), counter, t /
   1000000);
}
probe kernel.function("shrink_slab") {
  printf("%s: shrink_slab() started¥n", execname());
  shrink_start[tid()] = gettimeofday_ns();
}
probe kernel.function("shrink_slab").return {
  t = gettimeofday_ns() - shrink_start[tid()];
  printf("%s: shrink_slab() took %u ms¥n", execname(), t / 1000000);
}'
---------- program end ----------
```

SystemTap Example 4 - Detect the dentry cache bouncing bomb.

We can observe that when the bomb exploded, a simple stat() can take longer than a second.

```
---------- output example start ----------
kswapd1: shrink_slab() started
a.out: vfs_stat(305840783) took 282 ms
a.out: vfs_stat(305840784) took 999 ms
a.out: vfs_stat(305840785) took 335 ms
a.out: vfs_stat(305840786) took 3654 ms
a.out: vfs_stat(305840787) took 3009 ms
a.out: vfs_stat(305840788) took 999 ms
a.out: vfs_stat(305840790) took 1228 ms
a.out: vfs_stat(305847338) took 2 ms
(...snipped...)
a.out: vfs_stat(305914482) took 12 ms
kswapd1: shrink_slab() took 11810 ms
---------- output example end ----------
```

Watch out for integer overflow problems.

- Warning for x86_64 servers which create many dentries.
  - shrink_dcache_memory() is a function which is prone to by error return a negative value due to integer overflow.

```
---------- linux-2.6.32-431.11.2.el6/fs/dcache.c ----------
889:static int shrink_dcache_memory(struct shrinker *shrink, int nr, gfp_t
   gfp_mask)
890:{
891:    if (nr) {
892:            if (!(gfp_mask & __GFP_FS))
893:                    return -1;
894:            prune_dcache(nr);
895:    }
896:    return (dentry_stat.nr_unused / 100) * sysctl_vfs_cache_pressure;
897:}
---------- linux-2.6.32-431.11.2.el6/fs/dcache.c ----------
```

Watch out for integer overflow problems.

- shrink_dcache_memory() is one of functions called via (*shrinker->shrink)() calls in shrink_slab().

```
---------- linux-2.6.32-431.11.2.el6/mm/vmscan.c ----------
223:unsigned long shrink_slab(unsigned long scanned, gfp_t gfp_mask,
224:                          unsigned long lru_pages)
225:{
226:    struct shrinker *shrinker;
227:    unsigned long ret = 0;
228:
229:    if (scanned == 0)
230:            scanned = SWAP_CLUSTER_MAX;
231:
232:    if (!down_read_trylock(&shrinker_rwsem)) {
233:            /* Assume we'll be able to shrink next time */
234:            ret = 1;
235:            goto out;
```

Watch out for integer overflow problems.

```
236:    }
237:
238:    list_for_each_entry(shrinker, &shrinker_list, list) {
239:            unsigned long long delta;
240:            unsigned long total_scan;
241:            unsigned long max_pass;
242:
243:            max_pass = (*shrinker->shrink)(shrinker, 0, gfp_mask);
244:            delta = (4 * scanned) / shrinker->seeks;
245:            delta *= max_pass;
246:            do_div(delta, lru_pages + 1);
247:            shrinker->nr += delta;
248:            if (shrinker->nr < 0) {
249:                    printk(KERN_ERR "shrink_slab: %pF negative objects
  to "
250:                           "delete nr=%ld¥n",
251:                           shrinker->shrink, shrinker->nr);
```

Watch out for integer overflow problems.

```
252:                    shrinker->nr = max_pass;
253:            }
254:
255:            /*
256:             * Avoid risking looping forever due to too large nr value:
257:             * never try to free more than twice the estimate number of
258:             * freeable entries.
259:             */
260:            if (shrinker->nr > max_pass * 2)
261:                    shrinker->nr = max_pass * 2;
262:
263:            total_scan = shrinker->nr;
264:            shrinker->nr = 0;
265:
266:            while (total_scan >= SHRINK_BATCH) {
267:                    long this_scan = SHRINK_BATCH;
268:                    int shrink_ret;
```

Watch out for integer overflow problems.

```
269:                    int nr_before;
270:
271:                    nr_before = (*shrinker->shrink)(shrinker, 0,
    gfp_mask);
272:                    shrink_ret = (*shrinker->shrink)(shrinker, this_scan,
273:                                                     gfp_mask);
274:                    if (shrink_ret == -1)
275:                            break;
276:                    if (shrink_ret < nr_before)
277:                            ret += nr_before - shrink_ret;
278:                    count_vm_events(SLABS_SCANNED, this_scan);
279:                    total_scan -= this_scan;
280:
281:                    cond_resched();
282:            }
283:
284:            shrinker->nr += total_scan;
```

Watch out for integer overflow problems.

```
285:    }
286:    up_read(&shrinker_rwsem);
287:out:
288:    cond_resched();
289:    return ret;
290:}
---------- linux-2.6.32-431.11.2.el6/mm/vmscan.c ----------
```

- On x86_64, "int" is 32 bits and "unsigned long" is 64 bits. This means that if the number of unused dentries (second number in /proc/sys/fs/dentry-state) divided by 100 multiplied by /proc/sys/vm/vfs_cache_pressure at line 896 exceeded INT_MAX, the value of "max_pass" at line 243 becomes nearly ULONG_MAX due to sign extension.

Watch out for integer overflow problems.

- As a result, the value of "total_scan" at line 263 also becomes unexpectedly large, and any thread which called shrink_slab() for memory reclaim will fall into silent unkillable almost-infinite busy loop as if the thread hang up.
- To avoid such risk, /proc/sys/vm/vfs_cache_pressure should be kept small (i.e. between 1 and 100).

- This kind of problem one day suddenly happens.
  - Don't forget to check resource usage monitor like sar in sysstat package.
  - So-called "208.5 days problem" in RHEL 6 was fixed in kernel-2.6.32-358.23.2.el6 .
    - If you cannot update kernel for some reason, don't forget when your server was (cold / warm) rebooted.

SystemTap Example 5 - Find the process sending unidentified packet.

- Find out which process is sending UDP packets to host 127.0.0.1 port 53.

```
---------- program start ----------
# stap -e '
probe kernel.statement("*@net/ipv4/udp.c:$N") {
  if ($dport == htons(53) && $daddr == htonl(0x7F000001))
    printf("pid=%d comm=%s¥n", pid(), execname());
}'
---------- program end ----------
```

Note: $N needs to be replaced with the line number in udp_sendmsg() somewhere after "daddr" and "dport" variables are assigned.
For example, $N = 693 for kernel-2.6.32-431.11.2.el6 and
$N = 577 for kernel-2.6.18-371.8.1.el5 .

SystemTap Example 5 - Find the process sending unidentified packet.

```
---------- linux-2.6.32-431.11.2.el6/net/ipv4/udp.c ----------
615:int udp_sendmsg(struct kiocb *iocb, struct sock *sk, struct msghdr *msg,
616:                size_t len)
617:{
(...snipped...)
625:    __be32 daddr, faddr, saddr;
626:    __be16 dport;
(...snipped...)
669:    if (msg->msg_name) {
670:                struct sockaddr_in * usin = (struct sockaddr_in *)msg-
  >msg_name;
(...snipped...)
678:                daddr = usin->sin_addr.s_addr;
679:                dport = usin->sin_port;
(...snipped...)
```

SystemTap Example 5 - Find the process sending unidentified packet.

```
682:    } else {
(...snipped...)
685:            daddr = inet->daddr;
686:            dport = inet->dport;
(...snipped...)
691:    }
692:    ipc.addr = inet->saddr;
693:
----------------------------------
694:    ipc.oif = sk->sk_bound_dev_if;
(...snipped...)
853:}
---------- linux-2.6.32-431.11.2.el6/net/ipv4/udp.c ----------

---------- output example start ----------
pid=4460 comm=a.out
---------- output example end ----------
```

SystemTap Example 5 - Find the process sending unidentified packet.

- What does this script differ from strace program?
  - Doing strace on all processes would be a distant idea.
    (e.g. too slow, too complicated, might change the behavior)
- What does this script differ from system call auditing?
  - IP address and port number are packed into a structured variable, but system call auditing is not good at auditing variables in a structure.
- What does this script differ from auditing via LSM interface?
  - This script is very handy.

Is SystemTap good at everything?

- SystemTap can be used for not only measuring performance of functionality but also tracing functionality.
  - LSM interface allows probing at only locations where LSM callback hooks are provided, for it is designed for making security decision and auditing.

- SystemTap allows probing at almost everywhere (not only the start/end of a function but also any line number in a function in the source code).
  - For example, you can find out the exact location in the source code where the errno the system call auditing would record was set, by writing a SystemTap script which probes at specific line number.

Is SystemTap good at everything?

- Unfortunately, SystemTap is not a tool designed for monitoring throughout years.
  - LSM modules do not skip events nor stop working until shutdown, but SystemTap might skip events or stop working due to SystemTap's safety mechanism (and/or external factors like SIGKILL) before the event you want to record occurs.

- Please check whether SystemTap is suitable for solving your problem.
  - There will be cases where system call auditing is better.
  - There will be cases where single function LSM modules explained later is better.

Chapter 2

Preparation for user space problems.

Index of "Preparation for user space problems."

- What tools can we use for tracking user space problems?
- TOMOYO Linux
- AKARI
- Single function LSM modules
- SystemTap Example 6 - Track program execution.
- SystemTap Example 7 - A bit longer script.
- CaitSith

What tools can we use for tracking user space problems?

- When a system trouble happened, you need to retrieve and examine log files as soon as possible.
  - Where is the location of log files?
    - Mainly /var/log/ directory but not limited to.
- Which application is using which log files / configuration files?
  - There are tools for understanding what is happening in your system. You can utilize these tools before you actually encounter a system trouble.
    - TOMOYO Linux / AKARI / Single function LSM modules / SystemTap / CaitSith etc.

TOMOYO Linux

- My main contribution in Linux.
  - A tool for tracking/restricting various operations from boot.
  - Mainlined version is available since Linux 2.6.30 kernel.
- Process history / tree-view style tracking.

```
---------- history example start ----------
  0: <kernel>
  1:     /sbin/init
  2:         /bin/sh
  3:             /bin/awk
  4:             /bin/cat
  5:             /bin/grep
  6:             /bin/plymouth
  7:             /etc/rc.d/rc
  8:                 /bin/plymouth
  9:                 /etc/rc.d/init.d/auditd
```

Indentation represents caller/callee relationship. That is, kernel has executed /sbin/init , and /sbin/init executed by kernel has executed /bin/sh , and /bin/sh executed by /sbin/init executed by kernel has executed /etc/rc.d/rc , and so on.

TOMOYO Linux

```
10:                 /bin/bash
11:                     /sbin/auditd
12:                         /sbin/audispd
13:                 /bin/touch
14:                 /sbin/auditctl
15:             /etc/rc.d/init.d/blk-availability
16:                 /bin/touch
17:             /etc/rc.d/init.d/crond
18:                 /bin/bash
19:                     /usr/sbin/crond
20:                 /bin/touch
21:             /etc/rc.d/init.d/ip6tables
22:                 /bin/awk
23:                 /bin/cat
24:                 /bin/grep
25:                 /bin/touch
26:                 /sbin/ip6tables-restore
```

TOMOYO Linux

```
27:                 /sbin/modprobe
28:             /sbin/lsmod
29:             /sbin/modprobe
30:         /etc/rc.d/init.d/iptables
31:             /bin/awk
32:             /bin/cat
33:             /bin/grep
34:             /bin/touch
35:             /sbin/iptables-restore
36:                 /sbin/modprobe
37:             /sbin/lsmod
38:             /sbin/modprobe
39:         /etc/rc.d/init.d/iscsi
40:             /bin/grep
41:         /etc/rc.d/init.d/iscsid
42:             /bin/awk
43:             /bin/grep
```

TOMOYO Linux

```
44:                 /sbin/pidof
45:             /etc/rc.d/init.d/lvm2-monitor
46:                 /bin/touch
47:                 /sbin/vgs
48:             /etc/rc.d/init.d/mdmonitor
49:                 /usr/bin/id
50:             /etc/rc.d/init.d/netfs
51:                 /bin/awk
52:                 /bin/mount
53:                 /bin/touch
54:             /etc/rc.d/init.d/network
55:                 /bin/egrep
56:                 /bin/fgrep
57:                 /bin/ls
58:                 /bin/sed
59:                 /bin/sort
60:                 /bin/touch
```

TOMOYO Linux

```
61:                     /etc/sysconfig/network-scripts/ifup
62:                         /bin/awk
63:                         /bin/sed
64:                         /etc/sysconfig/network-scripts/ifup-eth
65:                             /bin/awk
66:                             /bin/cat
67:                             /bin/grep
68:                             /bin/ipcalc
69:                             /bin/sed
70:                             /etc/sysconfig/network-scripts/ifup-ipv6
71:                                 /bin/awk
72:                                 /bin/sed
73:                             /etc/sysconfig/network-scripts/ifup-post
74:                                 /bin/awk
75:                                 /bin/hostname
76:                                 /bin/ipcalc
77:                                 /bin/sed
```

```
78:                                 /etc/sysconfig/network-scripts/ifup-
  aliases
79:                                     /bin/awk
80:                                     /sbin/ip
81:                                 /etc/sysconfig/network-scripts/ifup-
  routes
82:                                     /sbin/ip
83:                               /sbin/dhclient
84:                                 /sbin/dhclient-script
85:                                     /bin/awk
86:                                     /bin/cat
87:                                     /bin/cut
88:                                     /bin/grep
89:                                     /bin/ipcalc
90:                                     /bin/mktemp
91:                                     /bin/rm
92:                                     /sbin/arping
```

TOMOYO Linux

```
 93:                                 /sbin/consoletype
 94:                                 /sbin/ip
 95:                                 /sbin/restorecon
 96:                                 /usr/bin/logger
 97:                         /sbin/ethtool
 98:                         /sbin/ip
 99:                 /etc/sysconfig/network-scripts/init.ipv6-global
100:                     /sbin/ip
101:                     /sbin/sysctl
102:                 /sbin/arp
103:                 /sbin/sysctl
104:         /etc/rc.d/init.d/postfix
105:                 /bin/basename
106:                 /bin/touch
107:                 /usr/sbin/postconf
108:                 /usr/sbin/postfix
109:                     /usr/libexec/postfix/postfix-script
```

TOMOYO Linux

```
110:                    /bin/sed
111:                    /usr/libexec/postfix/master
112:                        /usr/libexec/postfix/pickup
113:                        /usr/libexec/postfix/qmgr
114:                    /usr/libexec/postfix/postfix-script
115:                        /bin/egrep
116:                        /bin/find
117:                        /bin/grep
118:                        /bin/ls
119:                        /bin/sed
120:                        /bin/sh
121:                            /bin/grep
122:                            /bin/sed
123:                            /bin/uname
124:                            /usr/sbin/postconf
125:                        /usr/bin/cmp
126:                        /usr/sbin/postconf
```

TOMOYO Linux

```
127:                                    /usr/sbin/postsuper
128:                                /usr/sbin/postconf
129:                                /usr/sbin/postlog
130:                        /etc/rc.d/init.d/rsyslog
131:                            /bin/bash
132:                                /sbin/rsyslogd
133:                            /bin/touch
134:                        /etc/rc.d/init.d/sshd
135:                            /bin/cat
136:                            /bin/touch
137:                            /sbin/runlevel
138:                            /usr/sbin/sshd
139:                                /usr/sbin/sshd
140:                                    /bin/bash
141:                                        /bin/grep
142:                                        /bin/hostname
143:                                        /sbin/consoletype
```

TOMOYO Linux

```
144:                                    /usr/bin/dircolors
145:                                    /usr/bin/id
146:                                    /usr/bin/tput
147:                                    /usr/bin/tty
148:                                    /usr/sbin/ccs-editpolicy
149:                                    /usr/sbin/ccs-savepolicy
150:                    /etc/rc.d/init.d/udev-post
151:                        /sbin/udevadm
152:                    /etc/rc.d/rc.local
153:                        /bin/touch
154:                    /sbin/consoletype
155:                    /sbin/initctl
156:                    /sbin/runlevel
157:                /sbin/initctl
158:                /sbin/mingetty
159:                    /bin/login
160:                        /bin/bash
```

TOMOYO Linux

```
161:                        /bin/grep
162:                        /bin/hostname
163:                        /sbin/consoletype
164:                        /usr/bin/dircolors
165:                        /usr/bin/id
166:                        /usr/bin/tput
167:                        /usr/bin/tty
168:            /sbin/telinit
169:        /etc/rc.d/rc.sysinit
170:            /bin/awk
171:            /bin/cat
172:            /bin/chgrp
173:            /bin/chmod
174:            /bin/chown
175:            /bin/dd
176:            /bin/dmesg
177:            /bin/find
```

TOMOYO Linux

```
178:            /bin/rm
179:        /bin/hostname
180:        /bin/mkdir
181:        /bin/mount
182:            /sbin/mount.tmpfs
183:                /bin/cut
184:                /bin/grep
185:                /bin/ls
186:                /bin/mount
187:        /bin/mv
188:        /bin/plymouth
189:        /bin/rm
190:        /bin/sed
191:        /bin/touch
192:        /sbin/consoletype
193:        /sbin/fsck
194:            /sbin/fsck.ext4
```

TOMOYO Linux

```
195:            /sbin/lvm
196:            /sbin/modprobe
197:            /sbin/pidof
198:            /sbin/rmmod
199:            /sbin/start_udev
200:                /bin/awk
201:                /bin/cat
202:                /bin/chown
203:                /bin/ln
204:                /bin/mkdir
205:                /bin/mknod
206:                /sbin/consoletype
207:                /sbin/fstab-decode
208:                    /bin/echo
209:                /sbin/modprobe
210:                /sbin/pidof
211:                /sbin/restorecon
```

TOMOYO Linux

```
212:                /sbin/rmmod
213:                /sbin/udevadm
214:                /sbin/udevd
215:                    /bin/bash
216:                    /etc/sysconfig/network-scripts/net.hotplug
217:                    /lib/udev/cdrom_id
218:                    /lib/udev/console_check
219:                    /lib/udev/console_init
220:                        /bin/loadkeys
221:                            /bin/sh
222:                                /bin/gzip
223:                        /bin/setfont
224:                            /bin/sh
225:                                /bin/gzip
226:                    /lib/udev/edd_id
227:                    /lib/udev/fstab_import
228:                    /lib/udev/path_id
```

TOMOYO Linux

```
229:                    /lib/udev/pci-db
230:                    /lib/udev/rename_device
231:                    /lib/udev/scsi_id
232:                    /lib/udev/write_net_rules
233:                    /sbin/blkid
234:                    /sbin/hwclock
235:                    /sbin/ifup
236:                        /bin/sed
237:                        /etc/sysconfig/network-scripts/ifup-eth
238:                            /bin/grep
239:                            /bin/ipcalc
240:                            /bin/sed
241:                            /etc/sysconfig/network-scripts/ifup-ipv6
242:                                /bin/sed
243:                                /sbin/consoletype
244:                            /etc/sysconfig/network-scripts/ifup-post
245:                                /bin/sed
```

TOMOYO Linux

```
246:                                    /etc/sysconfig/network-scripts/ifup-
  aliases
247:                                        /bin/awk
248:                                        /sbin/consoletype
249:                                        /sbin/ip
250:                                    /etc/sysconfig/network-scripts/ifup-
  routes
251:                                        /sbin/consoletype
252:                                    /sbin/consoletype
253:                                    /sbin/ip
254:                                /sbin/consoletype
255:                            /sbin/modprobe
256:                            /sbin/multipath
257:                    /sbin/swapon
258:                    /sbin/sysctl
259:            /sbin/modprobe
---------- history example end ----------
```

TOMOYO Linux

- Name based access tracking.
  - For example, the syslog daemon (/sbin/rsyslogd) is accessing resources listed below.

```
---------- access example start ----------
<kernel> /sbin/init /bin/sh /etc/rc.d/rc /etc/rc.d/init.d/rsyslog /bin/bash
  /sbin/rsyslogd
 0: file append  /var/log/cron
 1: file append  /var/log/maillog
 2: file append  /var/log/messages
 3: file append  /var/log/secure
 4: file chmod   /dev/log 0666
 5: file create  /var/run/syslogd.pid 0644
 6: file getattr /etc/ld.so.cache
 7: file getattr /etc/localtime
 8: file getattr /etc/rsyslog.conf
 9: file getattr /lib64/libc-2.12.so
```

This is the name of this process history.

These are resources accessed by threads with this process history.

TOMOYO Linux

```
10: file getattr /lib64/libdl-2.12.so
11: file getattr /lib64/libgcc_s-4.4.7-20120601.so.1
12: file getattr /lib64/libpthread-2.12.so
13: file getattr /lib64/librt-2.12.so
14: file getattr /lib64/libz.so.1.2.3
15: file getattr /lib64/rsyslog/imklog.so
16: file getattr /lib64/rsyslog/imuxsock.so
17: file getattr /lib64/rsyslog/lmnet.so
18: file getattr /var/log/cron
19: file getattr /var/log/maillog
20: file getattr /var/log/messages
21: file getattr /var/log/secure
22: file getattr /var/run/syslogd.pid
23: file getattr proc:/kmsg
24: file ioctl   /var/log/cron 0x5401
25: file ioctl   /var/log/maillog 0x5401
26: file ioctl   /var/log/messages 0x5401
```

TOMOYO Linux

```
27: file ioctl   /var/log/secure 0x5401
28: file mksock  /dev/log 0700
29: file read    /dev/tty
30: file read    /etc/ld.so.cache
31: file read    /etc/localtime
32: file read    /etc/rsyslog.conf
33: file read    /etc/rsyslog.d/
34: file read    /lib64/ld-2.12.so
35: file read    /lib64/libc-2.12.so
36: file read    /lib64/libdl-2.12.so
37: file read    /lib64/libgcc_s-4.4.7-20120601.so.1
38: file read    /lib64/libpthread-2.12.so
39: file read    /lib64/librt-2.12.so
40: file read    /lib64/libz.so.1.2.3
41: file read    /lib64/rsyslog/imklog.so
42: file read    /lib64/rsyslog/imuxsock.so
43: file read    /lib64/rsyslog/lmnet.so
```

TOMOYO Linux

```
44: file read    /var/run/syslogd.pid
45: file read    proc:/kmsg
46: file unlink  /dev/log
47: file write   /dev/tty
48: file write   /var/run/syslogd.pid
49: misc env     CONSOLETYPE
50: misc env     LANG
51: misc env     LANGSH_SOURCED
52: misc env     PATH
53: misc env     PREVLEVEL
54: misc env     PWD
55: misc env     RUNLEVEL
56: misc env     SHLVL
57: misc env     TERM
58: misc env     UPSTART_EVENTS
59: misc env     UPSTART_INSTANCE
60: misc env     UPSTART_JOB
```

TOMOYO Linux

```
61: misc env     _
62: misc env     previous
63: misc env     runlevel
64: network unix dgram bind /dev/log
65: network unix dgram recv anonymous
---------- access example end ----------
```

- For another example, the listener process of ssh daemon (/usr/sbin/sshd) is accessing resources listed below.

```
---------- access example start ----------
<kernel> /sbin/init /bin/sh /etc/rc.d/rc /etc/rc.d/init.d/sshd
   /usr/sbin/sshd
  0: capability    use_route
  1: file create   /var/run/sshd.pid 0666
  2: file execute  /usr/sbin/sshd exec.realpath="/usr/sbin/sshd"
   exec.argv[0]="/usr/sbin/sshd"
  3: file getattr  /dev/null
  4: file getattr  /dev/urandom
```

TOMOYO Linux

```
 5: file getattr  /etc/gai.conf
 6: file getattr  /etc/ld.so.cache
 7: file getattr  /etc/localtime
 8: file getattr  /etc/nsswitch.conf
 9: file getattr  /etc/passwd
10: file getattr  /etc/pki/tls/openssl.cnf
11: file getattr  /etc/ssh/ssh_host_dsa_key
12: file getattr  /etc/ssh/ssh_host_rsa_key
13: file getattr  /etc/ssh/sshd_config
14: file getattr  /lib64/libaudit.so.1.0.0
15: file getattr  /lib64/libc-2.12.so
16: file getattr  /lib64/libcom_err.so.2.1
17: file getattr  /lib64/libcrypt-2.12.so
18: file getattr  /lib64/libdl-2.12.so
19: file getattr  /lib64/libfipscheck.so.1.1.0
20: file getattr  /lib64/libfreebl3.so
21: file getattr  /lib64/libgssapi_krb5.so.2.2
```

TOMOYO Linux

```
22: file getattr  /lib64/libk5crypto.so.3.1
23: file getattr  /lib64/libkeyutils.so.1.3
24: file getattr  /lib64/libkrb5.so.3.3
25: file getattr  /lib64/libkrb5support.so.0.1
26: file getattr  /lib64/libnsl-2.12.so
27: file getattr  /lib64/libnspr4.so
28: file getattr  /lib64/libnss_files-2.12.so
29: file getattr  /lib64/libpam.so.0.82.2
30: file getattr  /lib64/libplc4.so
31: file getattr  /lib64/libplds4.so
32: file getattr  /lib64/libpthread-2.12.so
33: file getattr  /lib64/libresolv-2.12.so
34: file getattr  /lib64/librt-2.12.so
35: file getattr  /lib64/libselinux.so.1
36: file getattr  /lib64/libutil-2.12.so
37: file getattr  /lib64/libwrap.so.0.7.6
38: file getattr  /lib64/libz.so.1.2.3
```

TOMOYO Linux

```
39: file getattr  /usr/lib64/libcrypto.so.1.0.1e
40: file getattr  /usr/lib64/libnss3.so
41: file getattr  /usr/lib64/libnssutil3.so
42: file getattr  /var/empty/sshd/
43: file getattr  /var/run/sshd.pid
44: file getattr  proc:/filesystems
45: file getattr  proc:/self/oom_score_adj
46: file read     /dev/null
47: file read     /dev/tty
48: file read     /dev/urandom
49: file read     /etc/gai.conf
50: file read     /etc/ld.so.cache
51: file read     /etc/localtime
52: file read     /etc/nsswitch.conf
53: file read     /etc/passwd
54: file read     /etc/pki/tls/openssl.cnf
55: file read     /etc/ssh/ssh_host_dsa_key
```

TOMOYO Linux

```
56: file read     /etc/ssh/ssh_host_rsa_key
57: file read     /etc/ssh/sshd_config
58: file read     /lib64/ld-2.12.so
59: file read     /lib64/libaudit.so.1.0.0
60: file read     /lib64/libc-2.12.so
61: file read     /lib64/libcom_err.so.2.1
62: file read     /lib64/libcrypt-2.12.so
63: file read     /lib64/libdl-2.12.so
64: file read     /lib64/libfipscheck.so.1.1.0
65: file read     /lib64/libfreebl3.so
66: file read     /lib64/libgssapi_krb5.so.2.2
67: file read     /lib64/libk5crypto.so.3.1
68: file read     /lib64/libkeyutils.so.1.3
69: file read     /lib64/libkrb5.so.3.3
70: file read     /lib64/libkrb5support.so.0.1
71: file read     /lib64/libnsl-2.12.so
72: file read     /lib64/libnspr4.so
```

TOMOYO Linux

```
73: file read     /lib64/libnss_files-2.12.so
74: file read     /lib64/libpam.so.0.82.2
75: file read     /lib64/libplc4.so
76: file read     /lib64/libplds4.so
77: file read     /lib64/libpthread-2.12.so
78: file read     /lib64/libresolv-2.12.so
79: file read     /lib64/librt-2.12.so
80: file read     /lib64/libselinux.so.1
81: file read     /lib64/libutil-2.12.so
82: file read     /lib64/libwrap.so.0.7.6
83: file read     /lib64/libz.so.1.2.3
84: file read     /usr/lib64/libcrypto.so.1.0.1e
85: file read     /usr/lib64/libnss3.so
86: file read     /usr/lib64/libnssutil3.so
87: file read     proc:/filesystems
88: file read     proc:/self/fd/
89: file read     proc:/self/oom_score_adj
```

TOMOYO Linux

```
 90: file truncate proc:/self/oom_score_adj
 91: file write    /dev/null
 92: file write    /dev/tty
 93: file write    /var/run/sshd.pid
 94: file write    proc:/self/oom_score_adj
 95: misc env      CONSOLETYPE
 96: misc env      LANG
 97: misc env      LANGSH_SOURCED
 98: misc env      PATH
 99: misc env      PREVLEVEL
100: misc env      PWD
101: misc env      RUNLEVEL
102: misc env      SHLVL
103: misc env      SSH_USE_STRONG_RNG
104: misc env      TERM
105: misc env      UPSTART_EVENTS
106: misc env      UPSTART_INSTANCE
```

TOMOYO Linux

```
107: misc env      UPSTART_JOB
108: misc env      _
109: misc env      previous
110: misc env      runlevel
111: network inet  dgram send 0.0.0.0 22
112: network inet  dgram send :: 22
113: network inet  stream accept 192.168.0.3 2371
114: network inet  stream bind 0.0.0.0 22
115: network inet  stream bind :: 22
116: network inet  stream listen 0.0.0.0 22
117: network inet  stream listen :: 22
118: network unix  dgram send /dev/log
119: network unix  stream connect /var/run/nscd/socket
---------- access example end ----------
```

- For yet another example, the worker process of ssh daemon is accessing resources listed below.

TOMOYO Linux

```
---------- access example start ----------
<kernel> /sbin/init /bin/sh /etc/rc.d/rc /etc/rc.d/init.d/sshd
   /usr/sbin/sshd /usr/sbin/sshd
  0: capability    SYS_NICE
  1: capability    SYS_VHANGUP
  2: capability    use_route
  3: file chroot   /var/empty/sshd/
  4: file execute  /bin/bash exec.realpath="/bin/bash" exec.argv[0]="-bash"
  5: file getattr  /bin/bash
  6: file getattr  /dev/urandom
  7: file getattr  /etc/environment
  8: file getattr  /etc/gai.conf
  9: file getattr  /etc/group
 10: file getattr  /etc/host.conf
 11: file getattr  /etc/hosts
 12: file getattr  /etc/hosts.allow
 13: file getattr  /etc/hosts.deny
```

TOMOYO Linux

```
14: file getattr  /etc/ld.so.cache
15: file getattr  /etc/localtime
16: file getattr  /etc/motd
17: file getattr  /etc/nsswitch.conf
18: file getattr  /etc/pam.d/
19: file getattr  /etc/pam.d/other
20: file getattr  /etc/pam.d/password-auth-ac
21: file getattr  /etc/pam.d/sshd
22: file getattr  /etc/passwd
23: file getattr  /etc/pki/tls/openssl.cnf
24: file getattr  /etc/protocols
25: file getattr  /etc/resolv.conf
26: file getattr  /etc/security/limits.conf
27: file getattr  /etc/security/limits.d/90-nproc.conf
28: file getattr  /etc/security/pam_env.conf
29: file getattr  /etc/security/sepermit.conf
30: file getattr  /etc/selinux/config
```

TOMOYO Linux

```
31: file getattr  /etc/selinux/targeted/seusers
32: file getattr  /etc/shadow
33: file getattr  /etc/ssh/moduli
34: file getattr  /etc/ssh/ssh_host_dsa_key
35: file getattr  /etc/ssh/ssh_host_rsa_key
36: file getattr  /lib64/libaudit.so.1.0.0
37: file getattr  /lib64/libc-2.12.so
38: file getattr  /lib64/libcom_err.so.2.1
39: file getattr  /lib64/libcrypt-2.12.so
40: file getattr  /lib64/libdl-2.12.so
41: file getattr  /lib64/libfipscheck.so.1.1.0
42: file getattr  /lib64/libfreebl3.so
43: file getattr  /lib64/libgssapi_krb5.so.2.2
44: file getattr  /lib64/libk5crypto.so.3.1
45: file getattr  /lib64/libkeyutils.so.1.3
46: file getattr  /lib64/libkrb5.so.3.3
47: file getattr  /lib64/libkrb5support.so.0.1
```

TOMOYO Linux

```
48: file getattr  /lib64/libnsl-2.12.so
49: file getattr  /lib64/libnspr4.so
50: file getattr  /lib64/libnss_dns-2.12.so
51: file getattr  /lib64/libnss_files-2.12.so
52: file getattr  /lib64/libpam.so.0.82.2
53: file getattr  /lib64/libplc4.so
54: file getattr  /lib64/libplds4.so
55: file getattr  /lib64/libpthread-2.12.so
56: file getattr  /lib64/libresolv-2.12.so
57: file getattr  /lib64/librt-2.12.so
58: file getattr  /lib64/libselinux.so.1
59: file getattr  /lib64/libutil-2.12.so
60: file getattr  /lib64/libwrap.so.0.7.6
61: file getattr  /lib64/libz.so.1.2.3
62: file getattr  /lib64/security/pam_cracklib.so
63: file getattr  /lib64/security/pam_deny.so
64: file getattr  /lib64/security/pam_env.so
```

TOMOYO Linux

```
65: file getattr  /lib64/security/pam_keyinit.so
66: file getattr  /lib64/security/pam_limits.so
67: file getattr  /lib64/security/pam_localuser.so
68: file getattr  /lib64/security/pam_loginuid.so
69: file getattr  /lib64/security/pam_nologin.so
70: file getattr  /lib64/security/pam_permit.so
71: file getattr  /lib64/security/pam_selinux.so
72: file getattr  /lib64/security/pam_sepermit.so
73: file getattr  /lib64/security/pam_succeed_if.so
74: file getattr  /lib64/security/pam_unix.so
75: file getattr  /usr/lib64/libcrack.so.2.8.1
76: file getattr  /usr/lib64/libcrypto.so.1.0.1e
77: file getattr  /usr/lib64/libnss3.so
78: file getattr  /usr/lib64/libnssutil3.so
79: file getattr  /var/empty/sshd/
80: file getattr  /var/log/lastlog
81: file getattr  devpts:/0
```

TOMOYO Linux

```
82: file getattr  proc:/filesystems
83: file getattr  proc:/sys/crypto/fips_enabled
84: file ioctl    /dev/null 0x5401
85: file ioctl    /dev/ptmx 0x40045431
86: file ioctl    /dev/ptmx 0x5401
87: file ioctl    /dev/ptmx 0x5414
88: file ioctl    /dev/ptmx 0x80045430
89: file ioctl    devpts:/0 0x5401
90: file ioctl    devpts:/0 0x5402
91: file ioctl    devpts:/0 0x540E
92: file ioctl    socket:[family=2:type=2:protocol=17] 0x541B
93: file read     /dev/null
94: file read     /dev/ptmx
95: file read     /dev/tty
96: file read     /dev/urandom
97: file read     /etc/environment
98: file read     /etc/gai.conf
```

TOMOYO Linux

```
 99: file read     /etc/group
100: file read     /etc/host.conf
101: file read     /etc/hosts
102: file read     /etc/hosts.allow
103: file read     /etc/hosts.deny
104: file read     /etc/ld.so.cache
105: file read     /etc/localtime
106: file read     /etc/motd
107: file read     /etc/nsswitch.conf
108: file read     /etc/pam.d/other
109: file read     /etc/pam.d/password-auth-ac
110: file read     /etc/pam.d/sshd
111: file read     /etc/passwd
112: file read     /etc/pki/tls/openssl.cnf
113: file read     /etc/protocols
114: file read     /etc/resolv.conf
115: file read     /etc/security/limits.conf
```

TOMOYO Linux

```
116: file read     /etc/security/limits.d/
117: file read     /etc/security/limits.d/90-nproc.conf
118: file read     /etc/security/pam_env.conf
119: file read     /etc/security/sepermit.conf
120: file read     /etc/selinux/config
121: file read     /etc/selinux/targeted/seusers
122: file read     /etc/shadow
123: file read     /etc/ssh/moduli
124: file read     /etc/ssh/ssh_host_dsa_key
125: file read     /etc/ssh/ssh_host_rsa_key
126: file read     /lib64/ld-2.12.so
127: file read     /lib64/libaudit.so.1.0.0
128: file read     /lib64/libc-2.12.so
129: file read     /lib64/libcom_err.so.2.1
130: file read     /lib64/libcrypt-2.12.so
131: file read     /lib64/libdl-2.12.so
132: file read     /lib64/libfipscheck.so.1.1.0
```

TOMOYO Linux

```
133: file read     /lib64/libfreebl3.so
134: file read     /lib64/libgssapi_krb5.so.2.2
135: file read     /lib64/libk5crypto.so.3.1
136: file read     /lib64/libkeyutils.so.1.3
137: file read     /lib64/libkrb5.so.3.3
138: file read     /lib64/libkrb5support.so.0.1
139: file read     /lib64/libnsl-2.12.so
140: file read     /lib64/libnspr4.so
141: file read     /lib64/libnss_dns-2.12.so
142: file read     /lib64/libnss_files-2.12.so
143: file read     /lib64/libpam.so.0.82.2
144: file read     /lib64/libplc4.so
145: file read     /lib64/libplds4.so
146: file read     /lib64/libpthread-2.12.so
147: file read     /lib64/libresolv-2.12.so
148: file read     /lib64/librt-2.12.so
149: file read     /lib64/libselinux.so.1
```

TOMOYO Linux

```
150: file read     /lib64/libutil-2.12.so
151: file read     /lib64/libwrap.so.0.7.6
152: file read     /lib64/libz.so.1.2.3
153: file read     /lib64/security/pam_cracklib.so
154: file read     /lib64/security/pam_deny.so
155: file read     /lib64/security/pam_env.so
156: file read     /lib64/security/pam_keyinit.so
157: file read     /lib64/security/pam_limits.so
158: file read     /lib64/security/pam_localuser.so
159: file read     /lib64/security/pam_loginuid.so
160: file read     /lib64/security/pam_nologin.so
161: file read     /lib64/security/pam_permit.so
162: file read     /lib64/security/pam_selinux.so
163: file read     /lib64/security/pam_sepermit.so
164: file read     /lib64/security/pam_succeed_if.so
165: file read     /lib64/security/pam_unix.so
166: file read     /usr/lib64/libcrack.so.2.8.1
```

TOMOYO Linux

```
167: file read     /usr/lib64/libcrypto.so.1.0.1e
168: file read     /usr/lib64/libnss3.so
169: file read     /usr/lib64/libnssutil3.so
170: file read     /var/log/lastlog
171: file read     /var/run/utmp
172: file read     devpts:/0
173: file read     proc:/filesystems
174: file read     proc:/self/fd/
175: file read     proc:/sys/crypto/fips_enabled
176: file read     proc:/sys/kernel/ngroups_max
177: file truncate proc:/self/loginuid
178: file write    /dev/null
179: file write    /dev/ptmx
180: file write    /dev/tty
181: file write    /var/log/lastlog
182: file write    /var/log/wtmp
183: file write    /var/run/utmp
```

```
184: file write    devpts:/0
185: file write    proc:/self/loginuid
186: misc env      CONSOLETYPE
187: misc env      LANG
188: misc env      LANGSH_SOURCED
189: misc env      PATH
190: misc env      PREVLEVEL
191: misc env      PWD
192: misc env      RUNLEVEL
193: misc env      SHLVL
194: misc env      SSH_USE_STRONG_RNG
195: misc env      TERM
196: misc env      UPSTART_EVENTS
197: misc env      UPSTART_INSTANCE
198: misc env      UPSTART_JOB
199: misc env      _
200: misc env      previous
```

```
201: misc env      runlevel
202: network inet  dgram recv 192.168.0.1 53
203: network inet  dgram send 0.0.0.0 22
204: network inet  dgram send 192.168.0.1 53
205: network inet  dgram send :: 22
206: network unix  dgram send /dev/log
207: network unix  stream connect /var/run/nscd/socket
---------- access example end ----------
```

TOMOYO Linux

- What is nice with TOMOYO Linux as a tracking tool?
  - You can track file accesses (e.g. read/write/execute) of each program running on your system.
    - You can bring your system out of black box.
      - If you understand which application is using which log files / configuration files, you will be able to promptly collect necessary files for troubleshooting.
      - If you understand which process is accessing which resources, you will be able to make better decision about how to apply updated packages to your system.
- If you are interested in TOMOYO Linux
  - Please visit http://tomoyo.sourceforge.jp/

AKARI

- TOMOYO Linux is not available in Fedora/RHEL kernels.
  - Replacing kernel packages rebuilt with TOMOYO Linux enabled is not acceptable for enterprise servers?
    - Then, you can use AKARI, a loadable kernel module (LKM) version of TOMOYO Linux.
      - AKARI's efficiency is lower and functionality is limited compared to TOMOYO Linux.
      - AKARI is useful for understanding summary of your Fedora/RHEL systems without replacing the kernel package.
- If you are interested in AKARI
  - Please visit http://akari.sourceforge.jp/

Single function LSM modules

- AKARI is an unexpected usage of LSM interface but useful technique for implementing various "single function LSM module as LKM" which you need.
  - AKARI will serve as a template for implementing various "single function LSM module as LKM" which you need.
- TaskTracker is an example of "single function LSM module as LKM" for embedding TOMOYO's process history into auditing logs.
  - http://sourceforge.jp/projects/akari/scm/svn/tree/head/branches/tasktracker/
  - TaskTracker module cannot work on RHEL 5, for SELinux's hook is directly called until Linux 2.6.25 kernel.

Single function LSM modules


```
---------- output example start ----------
type=SYSCALL msg=audit(1399165623.768:149): arch=c000003e syscall=59
   success=yes exit=0 a0=7f635b769c30 a1=7f635b7742e0 a2=7f635b764ed0 a3=8
   items=2 ppid=1 pid=1411 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="sh" exe="/bin/bash"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-01:07:03)" key=(null)
type=SYSCALL msg=audit(1399165623.771:150): arch=c000003e syscall=59
   success=yes exit=0 a0=1c8ab80 a1=1c8ac00 a2=1c8acd0 a3=7fff042e3bf0
   items=2 ppid=1 pid=1411 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="mingetty"
   exe="/sbin/mingetty" subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)" key=(null)
type=SYSCALL msg=audit(1399165627.169:151): arch=c000003e syscall=59
   success=yes exit=0 a0=402655 a1=7fff917530a0 a2=7fff91755270
   a3=7fff91754eb0 items=2 ppid=1 pid=1411 auid=4294967295 uid=0 gid=0
   euid=0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=tty1 ses=4294967295
   comm="login" exe="/bin/login" subj="init(2014/05/04-
   01:04:33)=>sh(2014/05/04-01:07:03)=>mingetty(2014/05/04-
   01:07:03)=>login(2014/05/04-01:07:07)" key=(null)
```

Single function LSM modules

```
type=USER_AUTH msg=audit(1399165629.963:152): user pid=1411 uid=0
   auid=4294967295 ses=4294967295 subj="init(2014/05/04-
   01:04:33)=>sh(2014/05/04-01:07:03)=>mingetty(2014/05/04-
   01:07:03)=>login(2014/05/04-01:07:07)" msg='op=PAM:authentication
   acct="root" exe="/bin/login" hostname=? addr=? terminal=tty1 res=success'
type=USER_ACCT msg=audit(1399165629.963:153): user pid=1411 uid=0
   auid=4294967295 ses=4294967295 subj="init(2014/05/04-
   01:04:33)=>sh(2014/05/04-01:07:03)=>mingetty(2014/05/04-
   01:07:03)=>login(2014/05/04-01:07:07)" msg='op=PAM:accounting acct="root"
   exe="/bin/login" hostname=? addr=? terminal=tty1 res=success'
type=LOGIN msg=audit(1399165629.963:154): pid=1411 uid=0
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-01:07:07)" old
   auid=4294967295 new auid=0 old ses=4294967295 new ses=10
type=USER_START msg=audit(1399165629.965:155): user pid=1411 uid=0 auid=0
   ses=10 subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-01:07:07)"
   msg='op=PAM:session_open acct="root" exe="/bin/login" hostname=? addr=?
   terminal=tty1 res=success'
```

Single function LSM modules

```
type=CRED_ACQ msg=audit(1399165629.965:156): user pid=1411 uid=0 auid=0
   ses=10 subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-01:07:07)"
   msg='op=PAM:setcred acct="root" exe="/bin/login" hostname=? addr=?
   terminal=tty1 res=success'
type=USER_LOGIN msg=audit(1399165629.965:157): user pid=1411 uid=0 auid=0
   ses=10 subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-01:07:07)"
   msg='op=login id=0 exe="/bin/login" hostname=? addr=? terminal=tty1
   res=success'
type=SYSCALL msg=audit(1399165629.967:158): arch=c000003e syscall=59
   success=yes exit=0 a0=1ecd5e0 a1=7fff1eb1ff18 a2=1ede3c0 a3=7fff1eb1f9f0
   items=2 ppid=1411 pid=1412 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="bash" exe="/bin/bash"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)" key=(null)
```

Single function LSM modules

```
type=SYSCALL msg=audit(1399165629.974:159): arch=c000003e syscall=59
   success=yes exit=0 a0=1a86030 a1=1a86330 a2=1a82df0 a3=7fff0e92e7e0
   items=2 ppid=1413 pid=1414 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="id" exe="/usr/bin/id"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>id(2014/05/04-01:07:09)" key=(null)
type=SYSCALL msg=audit(1399165629.982:160): arch=c000003e syscall=59
   success=yes exit=0 a0=1a85650 a1=1a84670 a2=1a867c0 a3=7fff0e92ed30
   items=2 ppid=1415 pid=1416 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="hostname" exe="/bin/hostname"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>hostname(2014/05/04-01:07:09)"
   key=(null)
```

Single function LSM modules

```
type=SYSCALL msg=audit(1399165629.991:161): arch=c000003e syscall=59
   success=yes exit=0 a0=1a8b6c0 a1=1a8b9c0 a2=1a8b160 a3=7fff0e92db30
   items=2 ppid=1417 pid=1418 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="tty" exe="/usr/bin/tty"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>tty(2014/05/04-01:07:09)"
   key=(null)
type=SYSCALL msg=audit(1399165629.995:162): arch=c000003e syscall=59
   success=yes exit=0 a0=1a8b500 a1=1a8b980 a2=1a8b160 a3=7fff0e92db30
   items=2 ppid=1417 pid=1419 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="tput" exe="/usr/bin/tput"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>tput(2014/05/04-01:07:09)"
   key=(null)
```

Single function LSM modules

```
type=SYSCALL msg=audit(1399165630.003:163): arch=c000003e syscall=59
   success=yes exit=0 a0=1a8af90 a1=1a86930 a2=1a8b160 a3=7fff0e92e250
   items=2 ppid=1420 pid=1421 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="dircolors"
   exe="/usr/bin/dircolors" subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>dircolors(2014/05/04-01:07:10)"
   key=(null)
type=SYSCALL msg=audit(1399165630.008:164): arch=c000003e syscall=59
   success=yes exit=0 a0=1a8a220 a1=1a8a830 a2=1a8b160 a3=7fff0e92e8e0
   items=2 ppid=1412 pid=1422 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="grep" exe="/bin/grep"
   subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>grep(2014/05/04-01:07:10)"
   key=(null)
```

Single function LSM modules

```
type=SYSCALL msg=audit(1399165630.016:165): arch=c000003e syscall=59
   success=yes exit=0 a0=1a8a8f0 a1=1a94a70 a2=1a8c790 a3=7fff0e92e0c0
   items=2 ppid=1423 pid=1424 auid=0 uid=0 gid=0 euid=0 suid=0 fsuid=0
   egid=0 sgid=0 fsgid=0 tty=tty1 ses=10 comm="consoletype"
   exe="/sbin/consoletype" subj="init(2014/05/04-01:04:33)=>sh(2014/05/04-
   01:07:03)=>mingetty(2014/05/04-01:07:03)=>login(2014/05/04-
   01:07:07)=>bash(2014/05/04-01:07:09)=>consoletype(2014/05/04-01:07:10)"
   key=(null)
---------- output example end ----------
```

- Currently, adding single function LSM modules to mainline Linux kernel is difficult.
  - One of reasons is due to LSM's exclusiveness.
    - When multiple concurrent LSM support (developed by Casey Schaufler) is added, the barrier could be lowered.
  - Consult me if you need single function LSM modules.

SystemTap Example 6 - Track program execution.

- AKARI requires rebooting the system in order to obtain complete process history from boot, and to unload the AKARI's LKM.
  - Rebooting would be acceptable for troubleshooting development environment but may not be acceptable for troubleshooting already running production environment because it involves restarting all processes?
    - If process history without rebooting the system is sufficient for solving problems, you could to some degree mimic TOMOYO's process history using SystemTap.

SystemTap Example 6 - Track program execution.

```
---------- program start ----------
# stap -g -DMAXSTRINGLEN=4096 -e '
global task_domain[32768];
function get_current:long() {
  return task_current() & %{ ULONG_MAX %};
}
function is_success:long(ret:long) {
  return ret <= -4096 || ret >= 0;
}
function make_domain:string() {
  task = get_current();
  if (task_domain[task] == "")
    task_domain[task] = sprintf("%s(%d)", execname(), pid());
  return task_domain[task];
}
probe kernel.function("copy_process").return {
  if (is_success($return))
```

Returns pointer to the current thread.

Create a process history if it does not exist.

Called when a thread was successfully created.

SystemTap Example 6 - Track program execution.

```
    task_domain[$return] = make_domain();
}
probe kernel.function("do_execve") {
  make_domain();
}
probe kernel.function("do_execve").return {
  if (is_success($return)) {
    task = get_current();
    domain = task_domain[task];
    if (domain != "") {
      filename = kernel_string($filename);
      printf("[%s] starting %s by uid=%d from %s¥n", ctime(gettimeofday_s()),
             filename, uid(), domain);
      task_domain[task] .= " " . filename;
    }
  }
}
```

Process history is inherited here.

Called when a program was successfully executed.

Process history is updated here.

Process history is printed here.

SystemTap Example 6 - Track program execution.

```
probe kernel.function("free_task") {
  delete task_domain[$tsk];
}
probe end {
  delete task_domain;
}'
---------- program end ----------
---------- output example start ----------
[Sun May  4 00:54:16 2014] starting /bin/sh by uid=0 from init(1)
[Sun May  4 00:54:16 2014] starting /sbin/mingetty by uid=0 from init(1)
  /bin/sh
[Sun May  4 00:54:18 2014] starting /bin/login by uid=0 from init(1) /bin/sh
  /sbin/mingetty
[Sun May  4 00:54:20 2014] starting /bin/bash by uid=0 from init(1) /bin/sh
  /sbin/mingetty /bin/login
[Sun May  4 00:54:20 2014] starting /usr/bin/id by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
```

Called when a thread terminated.

SystemTap Example 6 - Track program execution.

```
[Sun May  4 00:54:20 2014] starting /bin/hostname by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:20 2014] starting /usr/bin/tty by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:20 2014] starting /usr/bin/tput by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:20 2014] starting /usr/bin/dircolors by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:20 2014] starting /bin/grep by uid=0 from init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:20 2014] starting /sbin/consoletype by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Sun May  4 00:54:35 2014] starting /usr/bin/ssh by uid=0 from init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
---------- output example end ----------
```

SystemTap Example 7 - A bit longer script.

- If you add probes at file open functions, you can to some degree mimic TOMOYO's read/write/execute tracking.
  - Although there are limitations (e.g. the pathnames recorded are not always absolute ones, entries each process history can remember is limited), the script shown here will be handy for rough profiling.
  - This script is just an example. You can customize this script for your needs.

SystemTap Example 7 - A bit longer script.

```
---------- program start ----------
# stap -g -DMAXSTRINGLEN=4096 -e '
global task_domain[32768];
global history_domain;
global history_execve;
global history_read;
global history_write;

probe begin {
  printf("Probe start!¥n");
}
function get_current:long() {
  return task_current() & %{ ULONG_MAX %};
}
function is_success:long(ret:long) {
  return ret <= -4096 || ret >= 0;
}
```

SystemTap Example 7 - A bit longer script.

```
function make_domain:string() {
  task = get_current();
  if (task_domain[task] == "")
    task_domain[task] = sprintf("%s(%d)", execname(), pid());
  if (history_domain[task_domain[task]] == "")
    history_domain[task_domain[task]] = task_domain[task];
  return history_domain[task_domain[task]];
}
probe kernel.function("copy_process").return {
  if (is_success($return))
    task_domain[$return] = make_domain();
}
probe kernel.function("do_execve") {
     make_domain();
}
```

SystemTap Example 7 - A bit longer script.

```
probe kernel.function("do_execve").return {
  if (is_success($return)) {
    task = get_current();
    domain = task_domain[task];
    if (domain != "") {
      filename = kernel_string($filename);
      printf("[%s] execve %s by %s¥n", ctime(gettimeofday_s()),
             filename, domain);
      name = "  " . filename . "¥n";
      if (isinstr(history_execve[domain], name) == 0)
        history_execve[domain] .= name;
      task_domain[task] .= " " . filename;
      history_domain[task_domain[task]] = task_domain[task];
    }
  }
}
```

SystemTap Example 7 - A bit longer script.

```
probe kernel.function("do_sys_open").return {
  if (is_success($return)) {
    domain = make_domain();
    filename = user_string($filename);
    if (($flags & 3) != 3)
      printf("[%s] %s %s by %s¥n", ctime(gettimeofday_s()),
             ($flags & 3) == 0 ? "read" : (($flags & 3) == 1 ? "write" :
             "read/write"), filename, domain);
    name = "  " . filename . "¥n";
    if (($flags & 3) == 0 || ($flags & 3) == 2)
      if (isinstr(history_read[domain], name) == 0)
        history_read[domain] .= name;
    if (($flags & 3) == 1 || ($flags & 3) == 2)
      if (isinstr(history_write[domain], name) == 0)
        history_write[domain] .= name;
  }
}
```

Called when a file was successfully opened.

SystemTap Example 7 - A bit longer script.

```
probe kernel.function("free_task") {
  delete task_domain[$tsk];
}
probe end {
  printf("Probe end!¥n");
  foreach(i in history_domain) {
    domain = history_domain[i];
    printf("domain: %s¥n", domain);
    if (history_execve[domain] != "")
      printf("execve:¥n%s", history_execve[domain]);
    if (history_read[domain] != "")
      printf("read:¥n%s", history_read[domain]);
    if (history_write[domain] != "")
      printf("write:¥n%s", history_write[domain]);
    printf("¥n");
  }
```

Called when the SystemTap process is about to terminate.

SystemTap Example 7 - A bit longer script.

```
  delete history_domain;
  delete history_read;
  delete history_write;
  delete history_execve;
}'
---------- program end ----------
---------- output example start ----------
Probe start!
[Mon May  5 13:44:36 2014] read /var/run/utmp by init(1)
[Mon May  5 13:44:36 2014] read/write /var/run/utmp by init(1)
[Mon May  5 13:44:36 2014] write /var/log/wtmp by init(1)
[Mon May  5 13:44:36 2014] read/write /dev/null by init(1)
[Mon May  5 13:44:36 2014] read /var/run/utmp by init(1)
[Mon May  5 13:44:36 2014] read/write /var/run/utmp by init(1)
[Mon May  5 13:44:36 2014] execve /bin/sh by init(1)
[Mon May  5 13:44:36 2014] read /etc/ld.so.cache by init(1) /bin/sh
[Mon May  5 13:44:36 2014] read /lib64/libtinfo.so.5 by init(1) /bin/sh
```

Access logs with process history are printed here.

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:36 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
[Mon May  5 13:44:36 2014] read /lib64/libc.so.6 by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /proc/meminfo by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /etc/nsswitch.conf by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /etc/ld.so.cache by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /lib64/libnss_files.so.2 by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /etc/passwd by init(1) /bin/sh
[Mon May  5 13:44:37 2014] execve /sbin/mingetty by init(1) /bin/sh
[Mon May  5 13:44:37 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] read /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] read/write /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] write /var/log/wtmp by init(1) /bin/sh
  /sbin/mingetty
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:37 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:37 2014] read /etc/issue by init(1) /bin/sh /sbin/mingetty
[Mon May  5 13:44:38 2014] execve /bin/login by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:38 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libpam.so.0 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libpam_misc.so.0 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libselinux.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libaudit.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:38 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libcrypt.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libfreebl3.so by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/pam.d/login by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_securetty.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/pam.d/system-auth by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_env.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:38 2014] read /lib64/security/pam_unix.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libnsl.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_succeed_if.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_deny.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_nologin.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/pam.d/system-auth by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_localuser.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_permit.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:38 2014] read /etc/pam.d/system-auth by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_cracklib.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /usr/lib64/libcrack.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_selinux.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_loginuid.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_console.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_namespace.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_keyinit.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:38 2014] read /etc/pam.d/system-auth by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/security/pam_limits.so by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/pam.d/other by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/nsswitch.conf by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /lib64/libnss_files.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/securetty by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:38 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:38 2014] read /etc/shadow by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/shadow by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /lib64/libnspr4.so by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/sys/crypto/fips_enabled by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /lib64/libnspr4.so by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/shadow by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/sys/kernel/ngroups_max by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/group by init(1) /bin/sh /sbin/mingetty
  /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] write /proc/self/loginuid by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /proc/self/task/1178/attr/exec by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /etc/security/namespace.d by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/security/namespace.conf by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/security/limits.conf by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/security/limits.d by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/security/limits.d/90-nproc.conf by
  init(1) /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /etc/localtime by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/security/pam_env.conf by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/environment by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read/write /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] write /var/log/wtmp by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read/write /var/log/lastlog by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/group by init(1) /bin/sh /sbin/mingetty
  /bin/login
[Mon May  5 13:44:41 2014] read /etc/motd by init(1) /bin/sh /sbin/mingetty
  /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] execve /bin/bash by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /lib64/libtinfo.so.5 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read/write /dev/tty by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /proc/meminfo by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/nsswitch.conf by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /lib64/libnss_files.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/profile by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /usr/bin/id by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /lib64/libselinux.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /etc/nsswitch.conf by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /lib64/libnss_files.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/id
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /bin/hostname by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/hostname
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /lib64/libselinux.so.1 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/hostname
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/hostname
[Mon May  5 13:44:41 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/hostname
[Mon May  5 13:44:41 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/hostname
[Mon May  5 13:44:41 2014] read /etc/profile.d/ by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/profile.d/colorls.sh by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /usr/bin/tty by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/tty
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/tty
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /usr/bin/tput by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/tput
[Mon May  5 13:44:41 2014] read /lib64/libtinfo.so.5 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/tput
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/tput
[Mon May  5 13:44:41 2014] read /usr/share/terminfo/l/linux by init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash /usr/bin/tput
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] execve /usr/bin/dircolors by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/dircolors
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/dircolors
[Mon May  5 13:44:41 2014] read /etc/DIR_COLORS by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /usr/bin/dircolors
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /bin/grep by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/grep
[Mon May  5 13:44:41 2014] read /lib64/libpcre.so.0 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/grep
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/grep
[Mon May  5 13:44:41 2014] read /etc/DIR_COLORS by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /bin/grep
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] write /dev/null by init(1) /bin/sh /sbin/mingetty
  /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/profile.d/glib2.sh by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/profile.d/lang.sh by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/sysconfig/i18n by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /usr/lib/locale/locale-archive by init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /usr/lib64/gconv/gconv-modules.cache by
  init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] execve /sbin/consoletype by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /sbin/consoletype
[Mon May  5 13:44:41 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash /sbin/consoletype
[Mon May  5 13:44:41 2014] read /etc/profile.d/less.sh by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/profile.d/which2.sh by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /root/.bash_profile by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /root/.bashrc by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/bashrc by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:41 2014] read /root/.bash_history by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /root/.bash_history by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /usr/share/terminfo/l/linux by init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:41 2014] read /etc/inputrc by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:46 2014] read /usr/share/locale/locale.alias by init(1)
  /bin/sh /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:46 2014] read /root/.bash_logout by init(1) /bin/sh
  /sbin/mingetty /bin/login /bin/bash
[Mon May  5 13:44:46 2014] read /etc/security/pam_env.conf by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /etc/environment by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:46 2014] read /proc/filesystems by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /proc/self/task/1178/attr/exec by init(1)
  /bin/sh /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /etc/passwd by init(1) /bin/sh
  /sbin/mingetty /bin/login
[Mon May  5 13:44:46 2014] read /var/run/utmp by init(1)
[Mon May  5 13:44:46 2014] read/write /var/run/utmp by init(1)
[Mon May  5 13:44:46 2014] write /var/log/wtmp by init(1)
[Mon May  5 13:44:46 2014] read/write /dev/null by init(1)
[Mon May  5 13:44:46 2014] read /var/run/utmp by init(1)
[Mon May  5 13:44:46 2014] read/write /var/run/utmp by init(1)
[Mon May  5 13:44:46 2014] execve /bin/sh by init(1)
[Mon May  5 13:44:46 2014] read /etc/ld.so.cache by init(1) /bin/sh
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:46 2014] read /lib64/libtinfo.so.5 by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /lib64/libdl.so.2 by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /lib64/libc.so.6 by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /proc/meminfo by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /etc/nsswitch.conf by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /etc/ld.so.cache by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /lib64/libnss_files.so.2 by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /etc/passwd by init(1) /bin/sh
[Mon May  5 13:44:46 2014] execve /sbin/mingetty by init(1) /bin/sh
[Mon May  5 13:44:46 2014] read /etc/ld.so.cache by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read /lib64/libc.so.6 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read/write /var/run/utmp by init(1) /bin/sh
  /sbin/mingetty
```

SystemTap Example 7 - A bit longer script.

```
[Mon May  5 13:44:46 2014] write /var/log/wtmp by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read/write /dev/tty1 by init(1) /bin/sh
  /sbin/mingetty
[Mon May  5 13:44:46 2014] read /etc/issue by init(1) /bin/sh /sbin/mingetty
Probe end!
domain: init(1)
execve:
  /bin/sh
read:
  /var/run/utmp
  /dev/null
write:
  /var/run/utmp
  /var/log/wtmp
```

Received SIGINT here.

Summary is printed here for each process history.

SystemTap Example 7 - A bit longer script.

```
  /dev/null

domain: init(1) /bin/sh
execve:
  /sbin/mingetty
read:
  /etc/ld.so.cache
  /lib64/libtinfo.so.5
  /lib64/libdl.so.2
  /lib64/libc.so.6
  /proc/meminfo
  /etc/nsswitch.conf
  /lib64/libnss_files.so.2
  /etc/passwd

domain: init(1) /bin/sh /sbin/mingetty
execve:
```

SystemTap Example 7 - A bit longer script.

```
    /bin/login
  read:
    /etc/ld.so.cache
    /lib64/libc.so.6
    /var/run/utmp
    /dev/tty1
    /etc/issue
  write:
    /var/run/utmp
    /var/log/wtmp
    /dev/tty1

  domain: init(1) /bin/sh /sbin/mingetty /bin/login
  execve:
    /bin/bash
  read:
    /etc/ld.so.cache
```

SystemTap Example 7 - A bit longer script.

```
/lib64/libpam.so.0
/lib64/libpam_misc.so.0
/lib64/libselinux.so.1
/lib64/libaudit.so.1
/lib64/libc.so.6
/lib64/libdl.so.2
/lib64/libcrypt.so.1
/lib64/libfreebl3.so
/proc/filesystems
/dev/tty1
/etc/pam.d/login
/lib64/security/pam_securetty.so
/etc/pam.d/system-auth
/lib64/security/pam_env.so
/lib64/security/pam_unix.so
/lib64/libnsl.so.1
/lib64/security/pam_succeed_if.so
```

SystemTap Example 7 - A bit longer script.

```
/lib64/security/pam_deny.so
/lib64/security/pam_nologin.so
/lib64/security/pam_localuser.so
/lib64/security/pam_permit.so
/lib64/security/pam_cracklib.so
/usr/lib64/libcrack.so.2
/lib64/security/pam_selinux.so
/lib64/security/pam_loginuid.so
/lib64/security/pam_console.so
/lib64/security/pam_namespace.so
/lib64/security/pam_keyinit.so
/lib64/security/pam_limits.so
/etc/pam.d/other
/etc/nsswitch.conf
/lib64/libnss_files.so.2
/etc/passwd
/etc/securetty
```

SystemTap Example 7 - A bit longer script.

```
/etc/shadow
/lib64/libnspr4.so
/proc/sys/crypto/fips_enabled
/proc/sys/kernel/ngroups_max
/etc/group
/proc/self/task/1178/attr/exec
/etc/security/namespace.d
/etc/security/namespace.conf
/etc/security/limits.conf
/etc/security/limits.d
/etc/security/limits.d/90-nproc.conf
/var/run/utmp
/etc/localtime
/etc/security/pam_env.conf
/etc/environment
/var/log/lastlog
/etc/motd
```

SystemTap Example 7 - A bit longer script.

```
write:
  /dev/tty1
  /proc/self/loginuid
  /var/run/utmp
  /var/log/wtmp
  /var/log/lastlog

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash
execve:
  /usr/bin/id
  /bin/hostname
  /usr/bin/tty
  /usr/bin/tput
  /usr/bin/dircolors
  /bin/grep
  /sbin/consoletype
read:
```

SystemTap Example 7 - A bit longer script.

```
/etc/ld.so.cache
/lib64/libtinfo.so.5
/lib64/libdl.so.2
/lib64/libc.so.6
/dev/tty
/proc/meminfo
/etc/nsswitch.conf
/lib64/libnss_files.so.2
/etc/passwd
/etc/profile
/etc/profile.d/
/etc/profile.d/colorls.sh
/etc/profile.d/glib2.sh
/etc/profile.d/lang.sh
/etc/sysconfig/i18n
/usr/lib/locale/locale-archive
/usr/lib64/gconv/gconv-modules.cache
```

SystemTap Example 7 - A bit longer script.

```
  /etc/profile.d/less.sh
  /etc/profile.d/which2.sh
  /root/.bash_profile
  /root/.bashrc
  /etc/bashrc
  /root/.bash_history
  /usr/share/terminfo/l/linux
  /etc/inputrc
  /usr/share/locale/locale.alias
  /root/.bash_logout
write:
  /dev/tty
  /dev/null

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash /usr/bin/id
read:
  /etc/ld.so.cache
```

SystemTap Example 7 - A bit longer script.

```
  /lib64/libselinux.so.1
  /lib64/libc.so.6
  /lib64/libdl.so.2
  /proc/filesystems
  /etc/nsswitch.conf
  /lib64/libnss_files.so.2
  /etc/passwd

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash /bin/hostname
read:
  /etc/ld.so.cache
  /lib64/libselinux.so.1
  /lib64/libc.so.6
  /lib64/libdl.so.2
  /proc/filesystems

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash /usr/bin/tty
```

SystemTap Example 7 - A bit longer script.

```
read:
  /etc/ld.so.cache
  /lib64/libc.so.6

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash /usr/bin/tput
read:
  /etc/ld.so.cache
  /lib64/libtinfo.so.5
  /lib64/libc.so.6
  /usr/share/terminfo/l/linux

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash
   /usr/bin/dircolors
read:
  /etc/ld.so.cache
  /lib64/libc.so.6
  /etc/DIR_COLORS
```

SystemTap Example 7 - A bit longer script.

```
domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash /bin/grep
read:
  /etc/ld.so.cache
  /lib64/libpcre.so.0
  /lib64/libc.so.6
  /etc/DIR_COLORS

domain: init(1) /bin/sh /sbin/mingetty /bin/login /bin/bash
   /sbin/consoletype
read:
  /etc/ld.so.cache
  /lib64/libc.so.6

---------- output example end ----------
```

CaitSith

- A new type of rule based in-kernel access auditing and restricting tool.
  - Used as an auditing tool at Security Contest 2013 held in Japan.
    - http://2013.seccon.jp/ (Excuse me, but written in Japanese.)
  - Useful for targeted protection.
    - Good for those who are feeling that existing LSM modules are too difficult to use.
- If you are interested in CaitSith
  - Please visit http://caitsith.sourceforge.jp/

    or read http://I-love.SAKURA.ne.jp/tomoyo/CaitSith-en.pdf

Conclusion

- Troubleshooting is something that compares past state and current state.
- It is important that you know what the normal state is before you encounter troubles.
- There are parameters and tools which help you understand what the normal state of your system is and what is happening to your system.