
Proposal for FreeBSD audit system testing

Vincenzo Iozzo¹

FreeBSD Project
snagg@freebsd.org

1 Adding features

In order to improve the behavior of the auditpipe subsystem we should add some preselection routines for processing new type of audit records. The basic idea is to add a new mode in the auditpipe preselection functions, namely `AUDITPIPE_PRESELECT_MODE_EVENT`. The preselection function will check for a given mask and pid to see whether one of the pipe is interested in that event. As regards the ioctl preferences a new struct will be created, namely `auditpipe_ioctl_preselect_syscall`, with a bitmask for the syscall class and auid. This struct will be processed as usual with `auditpipe_ioctl`. Following are some pseudo-code examples:

```
1 struct auditpipe_ioctl_preselect {
2     au_id_t      aip_auid;
3     au_mask_t    aip_mask;
4     au_class_t   aip_event; //or something like short event[];
5     int          aip_sel;
6     pid_t        pid;
7 };
```

While the `preselect_find_syscall` would compare against the class and auid

```
1 static struct audit_pipe_preselect *
2 audit_pipe_preselect_find_sysclass(struct audit_pipe *ap, au_id_t auid, au_class_t class)
```

which would be checked by the `AUDITPIPE_PRESELECT_MODE_EVENT` mode.

As regards the ioctl functions, I'm planning to add something like `audit_pipe_preselect_set` which would take a mask or something like that to compare against the syscall table and generate a valid `au_mask_t`. The idea is to create some bitmasks for each event defined in the FreeBSD manual to be used with ioctl, then they will be mapped to syscalls as it is done by `au_event_class()` for example.

2 Testing

Testing the infrastructure would imply to check whether or not the behavior of auditpipe is consistent. In order to do so some test programs would be created. For example let's suppose we'd like to trace file operations. We would specify that we'd like to trace file operations. We will the struct that way:

```
1 struct auditpipe_ioctl_preselect query;
2 query.aip_event = AUE_OPEN | AUE_READ | AUE_WRITE
3 query.pid=getpid();
4 query.aip_sel= AU_PRS_BOTH;
```

We'll then call ioctl with specifying the preselection mode, `AUDITPIPE_PRESELECT_MODE_SYSCALL`. After that we would compare the results taken from the device with some logs describing the expected behavior of the testing program which would be later checked against the auditpipe log.