

Saturday, February 2, 2008

## Less known Solaris features: Auditing

One of the less known features in Solaris is the Auditing. Auditing solves an important problem: What happens on my system, and whodunnit. When something strange happens on your system or you recognize, that you are not the only one who owns your system, it's a good thing to have some logs for analysis.

The nice thing about the auditing in Solaris: It's quite simple to activate. In this article i will give you a short overview to enable and use the auditing in Solaris. This feature is really old, it's in Solaris for since the last century but nevertheless it's a less known Solaris feature.

### Some terms

There are some special terms in auditing. I want to give you a short definition of them as i have to use them in this article. I've copied this definitions from the manual for Solaris Auditing.

**Audit events:** A security-related system action that is audited. For ease of selection, events are grouped into audit classes. **Audit Class:** A grouping of audit events. Audit classes provide a way to select a group of events to be audited.

**Audit policy:** A set of auditing options that you can enable or disable at your site. These options include whether to record certain kinds of audit data. The options also include whether to suspend auditable actions when the audit trail is full.

### Configuring basic auditing

You have to search for a place in your filesystem. It's a good practice to use an own filesystem, as auditing will eat away your filesystem space until there is nothing left and this is a bad idea for the root file system. But in this example i will omit this step.

At first login as root. Okay, you need a place to store the audit logs. It's important to change the rights of the directory to assure only root can access it.

```
mkdir /var/audit/aragorn-sol
```

```
chmod -R 750 /var/audit/aragorn-sol
```

Then go to /etc/security and edit the file /etc/security/audit\_control. This file controls where what classes of information are logged and where you write the log. For example: The lo is the audit class for all events in regard of logins and logoffs:

```
dir:/var/audit/aragorn-sol
```

```
flags:lo
```

```
minfree:20
```

Okay, configuration is done. But let's have another look the file /etc/security/audit\_startup. The commands in this script control the audit policies and thus the behaviour of the logging and the amount of informations in the log records:

```
#!/usr/bin/echo "Starting BSM services."
```

```
/usr/sbin/auditconfig -setpolicy +cnt
```

```
/usr/sbin/auditconfig -conf
```

```
/usr/sbin/auditconfig -aconf
```

The second line is the most interesting. Without this line the system would stop user interaction when the system is unable to log. You would deactivate this behaviour, when logging is more important than system availability. For the moment we don't change this file.

### Start the auditing

Now activate auditing. You have to reboot after the activation.

```
#!/usr/sbin/bsmconv
```

This script is used to enable the Basic Security Module (BSM).

```
Shall we continue with the conversion now? [y/n] y
```

```
bsmconv: INFO: checking startup file.
```

```
bsmconv: INFO: turning on audit module.
```

```
bsmconv: INFO: initializing device allocation.
```

The Basic Security Module is ready.

If there were any errors, please fix them now.

Configure BSM by editing files located in /etc/security.

Reboot this system now to come up with BSM enabled.

```
# reboot
```

```
Two short checks ... auditd runs ...# svcs | grep "auditd"
```

```
online
```

```
# ls -la
```

```
total 6
```

drwxr-x---	2	root	root	512	Feb 1 23:30	.
drwxr-xr-x	3	root	sys	512	Feb 1 23:18	..

```
-rw-r----- 1 root root 255 Feb 1 23:33 20080201223003.not_terminated.aragorn-sol
```

Okay, now you have completed the configuration. The system has started to write audit logs.

#### Managing the audit logs

Audit logs grows infinitely. To the maximum filesize in the used filesystem or the end of disk capacity ... whatever occurs first. It's a good practice to checkpoint the audit logs in a regular interval. It's quite simple: `audit -n` With this command the actual file gets closed and a new one gets opened. `# cd /var/audit/aragorn-sol/`

```
# ls -l
total 24
-rw-r----- 1 root root 684 Feb 1 23:55 20080201223003.20080201225549.aragorn-sol
-rw-r----- 1 root root 571 Feb 2 00:06 20080201225549.20080201230639.aragorn-sol
-rw-r----- 1 root root 2279 Feb 2 00:10 20080201230834.20080201231010.aragorn-sol
-rw-r----- 1 root root 755 Feb 2 00:12 20080201231010.20080201231245.aragorn-sol
-rw-r----- 1 root root 4274 Feb 2 08:36 20080201231245.20080202073624.aragorn-sol
-rw-r----- 1 root root 200 Feb 2 08:36 20080202073624.not_terminated.aragorn-sol
```

#### Analysing the audit trails

It doesn't make sense to create audit logs without looking at them. You can't look directly at them as this file are binary ones. You need to command to analyse the audit log. One to extract the data out of the log files based on certain rules and one command to translate it into an human readable format.

You use the `auditreduce` command for the first step, and the `praudit` command for the second one. `# cd /var/audit/aragorn-sol`

```
auditreduce * | praudit -s
```

This sequence of commands translate all you audit logs into an human readable form. I've cut out some of the lines for an example: `header,69,2,AUE_ssh,,localhost,2008-02-01 23:49:17.687 +01:00`

```
subject,jmoekamp,jmoekamp,other,jmoekamp,other,720,3447782834,6969 5632 10.211.55.2
```

```
return,success,0
```

```
header,77,2,AUE_su,,localhost,2008-02-01 23:49:55.336 +01:00
```

```
subject,jmoekamp,root,other,jmoekamp,other,729,3447782834,6969 5632 10.211.55.2
```

```
text,root
```

```
return,failure,Authentication failed
```

```
header,69,2,AUE_su,,localhost,2008-02-01 23:50:11.311 +01:00
```

```
subject,jmoekamp,root,root,root,root,730,3447782834,6969 5632 10.211.55.2
```

```
return,success,0
```

What tells this snippet to you: I've logged into my system as the user `jmoekamp`, tried to assume root privileges, failed the first time (due wrong password), tried it again and succeeded.

#### More auditing

Sometimes it's important to know what users have done on you system. For example: Which programs has been executed. With Solaris auditing it's really easy to collect this information.

At first you have to configure auditing to collect this kind of information: `dir:/var/audit/aragorn-sol`

```
flags:lo,ex
```

```
minfree:20
```

```
naflags:lo,ex
```

The `ex` audit class matches to all events in system in regard to the execution of a program. This tells the auditing subsystem to log all `execve()` system calls. But you have to signal this change to the audit subsystem to start the auditing of this events. With `audit -s` you notify the audit daemon to read the `/etc/security/audit_control` file again.

```
header,113,2,AUE_EXECVE,,localhost,2008-02-02 00:10:00.623 +01:00
```

```
path,/usr/bin/ls
```

```
attribute,100555,root,bin,26738688,1380,0
```

```
subject,jmoekamp,root,root,root,root,652,2040289354,12921 71168 10.211.55.2
```

```
return,success,0
```

But this configuration only logs the path of the command, not the command line parameters. You have to configure to log this information. You remember: The audit policy controls the kind of information in the audit logs.

Thus we have to modify the audit policy. With the command `auditconfig -setpolicy +argv` you change the policy. You don't have to activate it, it's immediately effective: `header,124,2,AUE_EXECVE,,localhost,2008-02-02 00:12:49.560 +01:00`

```
path,/usr/bin/ls
```

```
attribute,100555,root,bin,26738688,1380,0
```

```
exec_args,2,ls,-l
```

```
subject,jmoekamp,root,root,root,root,665,2040289354,12921 71168 10.211.55.2
```

```
return,success,0
```

To make this behaviour persistent, you have add the `auditconfig -setpolicy +argv` to the file

Want to learn more?

## Blog Export: c0t0d0s0.org, <http://www.c0t0d0s0.org/>

This is only a really short introduction to the topic. You will find the documentation for this feature at docs.sun.com: Part VII Solaris Auditing of System Administration Guide: Security Services is a good place to start.

Posted by Joerg Moellenkamp at 11:21

When I think someone owend my box, the first thing I should do is a1:1 copy of my harddrive... running this auditing process first could damage important data (important like 'how did the attacker get on my system').

Beside that it's great article. But somehow I get the feeling that Schaeuble will use Solaris soon.  
Anonymous on Feb 2 2008, 15:23

Obviously you should activate auditing before an attacker hacks your system and move them to a different place on a regular schedule. So you can search for the attack vector and for the things done by the attacker.  
Anonymous on Feb 2 2008, 15:33

Good thing would be to use the audit\_syslog plugin. (<http://docs.sun.com/app/docs/doc/816-5175/6mbba7eup?a=view>) This will duplicate the events into syslog.

Syslog messages can then be forwarded to another system.  
Anonymous on Feb 2 2008, 15:42

Yes ... definitely. A central loghost for this messages is a good idea.  
Anonymous on Feb 2 2008, 15:51

Check out this BigAdmin page for a HOWTO on setting up a remote auditlog server using SSH.

[http://www.sun.com/bigadmin/content/submitted/bsm\\_audit.jsp](http://www.sun.com/bigadmin/content/submitted/bsm_audit.jsp)

See method #4 for the remote option (SSH + RBAC).

-Mike.  
Anonymous on Apr 17 2008, 14:20