Auditing NFS Events

Efstratios Karatzas
gpf@FreeBSD.org

FreeBSD Developer Summit
Karlsruhe Institute of Technology
Karlsruhe, Germany
October 8, 2010

Audit's Limitations

TrustedBSD's Audit implementation is widely used by sys/admins to keep an eye on their servers via run-time monitoring, post-mortem analysis, etc.

However, Audit is limited on the following aspects:

- Audit is focused on gathering data from local system calls.
- Audit assumes that a running kernel thread will be involved in at most one security event at a time.

Motivation

Why should we extend the functionality of Audit?

- There are plenty of kernel subsystems (such as firewalls or NFS) that generate security log data which are, at the moment, ignored by Audit.
- FreeBSD is widely used as a file server and current implementation only supports auditing of local filesystem activity, i.e. No logging of NFS server activity.
- Logging of NFS activity requested by sys/admins on multiple occasions.

Suggested GsoC Project Idea

How should we extend Audit?

► In order to successfuly audit other kernel subsystems, Audit must support handling of simultaneous audit records per running kernel thread.

i.e. two Audit Records are needed when an open(2) is used over NFS and this action triggers a Firewall Event.

► Actually extend other kernel subsystems to use the Audit framework and log security relevant information.

GSoC 2010 – Audit Kernel Events

Deliverables of this project in a nutshell:

► Audit Support for NFS RPCs that get serviced by your FreeBSD fileserver.

This includes support for both the current NFS implementation in sys/nfs* as well as the experimental NFS implementation in sys/fs/nfs* (NFS protocols v2,3 & 4).

► Handling of multiple audit records per kernel thread via use of a treelike data structure that is kept with each kernel thread.

Demo

This is the audit record for the NFSv3 RPC "WRITE":

```
header,181,11,nfsrv_write(),0,Mon Aug  9 20:21:08 2010, + 697 msec
argument,2,0x8,flags
path,/usr/home/gpf/Desktop/nfs/my_test/file
attribute,646,1002,0,104,216583,857980
text,192.168.1.10:639
protocol,NFSv3
subject,-1,1002,0,1002,0,1560,0,0,0.0.0.0
return,success,0
trailer,181
```

vnode* to file path? (1)

Task: Acquiring a full filename from just a vnode pointer; necessary in order to make sense out of NFS activity.

Pre-existing solution:

► vn_fullpath(9) – use the name cache to generate a full pathname.

vnode* to file path? (2)

Problems with vn_fullpath():

- Name Cache Hints are lost every time the server reboots.
- Contents of the cache are replaced in a semi-lru fashion.

Therefore, this KPI is deemed unreliable for use in a security submodule such as Audit.

vnode* to file path? (3)

Proposed Solution:

- Store the directory *node number (hint) inside the filehandle during filehandle creation, VOP_VPTOFH(9).
- Collect the *node number from the filehandle using VFS_FHHINT(9).
- Feed the vnode* and the directory hint to vn_fullpath_nocache(9).
- vn_fullpath_nocache(9) will then try to generate ***a*** working path for the vnode in question without making use of the name cache.

vn_fullpath_nocache(9)

The hardest part is to link a non-directory vnode with a parent directory - VOP_GETPARENT(9) is used.

- Use the directory parent hint to make this easier.
- Perform an exhaustive search on the filesystem.
- Use filesystem specific mechanisms to facilitate the seach. i.e. The parent znode_id is stored inside each znode in ZFS.

Problems:

- Hardlinks with multiple paths, temporary files with no path at all.

Todo List

Stuff left to do:

- Review code alongside mentor and merge with HEAD.
- Port VOP_GETPARENT(9) to other filesystems.
- Update Audit code at sys/fs/nfs* as new functionality is being added to the experimental NFS server.

Audit & GSoC (1)

Distributed Audit Daemon

- Alexey Mikhailov, GSoC 2007 – unsuccessful
- Sergio Ligregni, GSoC 2010 – successful

Application-Specific Audit Trails

- Ilias Marinos, GSoC 2009 – successful
- The same mechanisms can be extended to jails (work in progress)

Audit & GSoC (2)

Audit Log Analyzer

- Dongmei Liu, GSoC 2007 – unsuccessful

Audit Kernel Events

- Diego Giagio (Firewalls), GSoC 2008 – unsuccessful
- His work will be picked up by me :-)
- Efstratios Karatzas (NFS), GSoC 2010 – successful

The End

Thank you for your time!

For more information on my project, please visit the project's wiki page
http://wiki.freebsd.org/SOC2010EfstratiosKaratzas

Special Thanks to:
Robert Watson
Rick Macklem
FreeBSD Project & Community
Google & Carol Smith