AUDIT CONFIGURATION AND PERFORMANCE NOTES

Audit is a monitoring framework that records actions made by user or kernel. It can filter records by system call, user or file.

For our needs, we use the framework to monitor system calls.

SHORT SUMMARY OF CONCEPT

Client Side:

At boot time, the audit daemon starts monitoring. Periodically, we use a script to parse the records produced by the daemon. Then, we produce a list of actions of our interest (e.g. file opened, bytes read, bytes written). This list gets encrypted by a random and consistent hash function and gets sent to the server via ftp. In the mean time, the audit daemon doesn't stop running.

Server Side:

The data gets collected and we can output some statistics.

AUDIT CONFIGURATION

Software Requirements:

- Installed python 3
- Installed ftpdlib (~$ pip install pyftpdlib)

System Requirements:

- ~270MB of RAM space at worst case (this can be reduced if needed)

Installation:

Debian/Ubuntu : apt-get install auditd audispd-plugins

Red Hat/CentOS/Fedora : usually already installed (package: audit and audit-libs)

Configuration:

~# python3 <system_file_name>.py (This script needs root permissions to run (#))

There is need for **(2)** files to change in order to configure the daemon properly.

1) /etc/audit/auditd.conf : This file contains configuration information specific to the audit daemon.
2) /etc/audit/audit.rules : This file contains our rules used to monitor the system calls.

These changes are done **by the script** every time the system boots, so there is no need for manual changes.

Configured variables worth mentioning:

backlog: This is the number of maximum audit messages that can be queued before written on disk. Here is set to 30.000 messages. Every message is of size ~9KB, thus the required memory space is 270MB when the queue is at maximum size.

backlog_wait_time: This is the time that kernel waits for the queue above to be drained. In new kernels there is no need to wait, so we set it to 0. This gives best auditing performance as the kernel doesn't lose any records and most importantly user doesn't suffer from freezed system.

Some statistics of audit performance in my machine:

System Specs:

- Memory : 4GB
- CPU : Intel i5 – 7200U @ 2.5GHz
- OS Type : Ubuntu 16.04 LTS – 64 bit

TIME (hours)	Memory Usage (%)	CPU Usage (%)
0	2	0.1
0.5	2.6	0.1
1	2.3	0.2
2	1.7	0.1
3	2.7	0.1
4	2.3	0.2
5	2	0.1
6	1.5	0.1
7	2.6	0.2
8	2.1	0.1
9	1.8	0.1
10	2.7	0.1
11	2	0.1
12	1.7	0.1

Graphic Representation: