## HTCondor-CE Audit Log

### Overview

In order to satisfy OSG security and WLCG traceability requirements, an HTCondor-CE setup needs an audit log detailing users, their actions, and their jobs. The audit log must be easily searchable for certain types of queries.

We will provide this capability with a combination of a new audit log file and the existing event log and job history files.

### Audit Requirements

The audit information must include authentication and authorization information for each user-initiated event in the system (e.g. job submission, job removal, job ad modification). This includes source IP address, authentication methods attempted (both failed and successful), authentication identity (e.g. X.509 distinguished name and VOMS extensions), mapped HTCondor identity, authorization success/failure, and action requested. This information must be recorded even if the action is not performed, due to authentication or authorization failure.

The audit stream should be available in a grep-friendly format. It should be easy to take one line in the audit stream and find all related lines. Queries like the following should be possible:

* Print all the users who were on a given worker node from this CE.
* Print all the running jobs at a given point-in-time.
* Print all the actions of a specific user / DN.
* Print all the jobs from a given submit host IP.
* List of all executable names from a specific user.
* Print all the users with job proxies valid at time X.

Audit information should be retained for a set minimum length of time (e.g. keep all data up to 90 days old).

### Existing Logs

Some of HTCondor’s existing logging already provides some of the information we need. We will take advantage of these logs where appropriate.

The history file (and history directory) contains descriptions of all jobs that have left the job queue. Together with the job queue, this provides information about all jobs, past and present. This includes job owner, executable name, input and output filenames, and proxy identity information. Any information here that we need to rely on will have to be protected from editing by the user while in the queue.

The event log provides a list of interesting events about each job while it is in the queue. This includes where and when it executed, whether it was successful, and how and when it left the queue.

We may need to write a tool that provides audit information from the event log and history file in a grep-friendly format.

### New Audit Log

For information that’s not reliably captured in existing logs, we will create a new audit log. Each time a user connects to an HTCondor dameon to issue a relevant command (e.g. modify the job queue in some fashion), all relevant network and security information will be recorded in the audit log. Additional information specific to that command may also be recorded.

Not all commands need to be recorded in the audit log. The commands of primary interest are ones that modify the job queue or any data associated with submitted jobs. These include job submission, editing attributes of jobs, holding/releasing/removing/vacating jobs, and transferring job files (including updating X.509 proxies). Additional commands may be identified later. Commands known to originate from trusted daemons (i.e. local shadow or gridmanager) need not be logged.

Each new connection will be assigned a unique identifier. Reuse of identifiers after several days or a restart of the daemon is permissible. All audit log entries will begin with a timestamp and the id of the relevant connection. As described above, the following information will also be recorded, when available:

* Command id/name
* Source IP address
* Authentication methods attempted, both failed and successful
* Authenticated identity and HTCondor identity it maps to
* Whether the client is authorized

Additional information specific to the type of command may also be logged. For example, when new jobs are submitted, the job ids will be logged. All information for a given connection may be split across multiple lines. Since each line contains the connection id, related lines can easily be associated with each other, even if interspersed with other information.

The audit log will be implemented as a new dprintf() debug category, D\_AUDIT. The dprintf() system will be configured to write D\_AUDIT messages only to the audit log, which will be specified via a configuration parameter, probably AUDIT\_LOG. Daemon-core will log basic network and security information for all relevant commands. The command callouts in the schedd will log additional command-specific information. Security information and the connection id can be obtained from the Sock object for the connection.

OSG requires that audit information be retained for at least 90 days. This will require supporting time-based rotation of daemon logs.

## Development Plan

* [2 days] Identify commands to be logged, including additional information
* [1 day] Add D\_AUDIT debug category and AUDIT\_LOG config params
* [1 day] Add and expose connection and security information in CEDAR
* [2 days] Log network and security audit information
* [2 days] Log command-specific audit information
* [1 day] Make identity attributes in job ad immutable
* [2 days] Add time-based daemon log rotation