Files

Introduction

Open-AudIT can now retrieve details about a file or directory of files and monitor these files for changes as per other attributes in the Open-AudIT database.

This feature works out of the box for Linux Open-AudIT servers, but needs a change to the service account name under a Windows Open-AudIT server.

Supported clients are Windows and Linux.

How does it Work?

When a device is Discovered the audit script will be injected with the file (or directory) details. Both audit_linux.sh and audit_windows.vbs will be populated, regardless of the file path. If you're thinking "but putting a Linux path into a Windows machine will break things!", it won't break, you just won't receive any data from that file or directory entry.

Retrieved details will be stored in the "file" database table and are stored like any other attribute. File details will be stored as any other attribute and generate alerts if any of the following attributes change - 'full_name', 'hash', 'inode', 'last_changed'. The table contain a system_id, first_seen, last_seen and current columns as well as the file details.

Attributes recorded are:

Windows - name, size, directory, SHA1 hash (of the file contents), last changed, permissions, owner, version (file permitting).

Linux - name, size, directory, SHA1 hash (of the file contents), last changed, meta data last changed, permissions, owner, group, inode.

You will see a section in both the Windows and Linux audit scripts as below:

```
# DO NOT REMOVE THE LINE BELOW
# Configuration from web UI here
```

Below these lines are where the variables are injected into the script. Following on from our earlier example, the Linux audit script is populated with our directory like so:

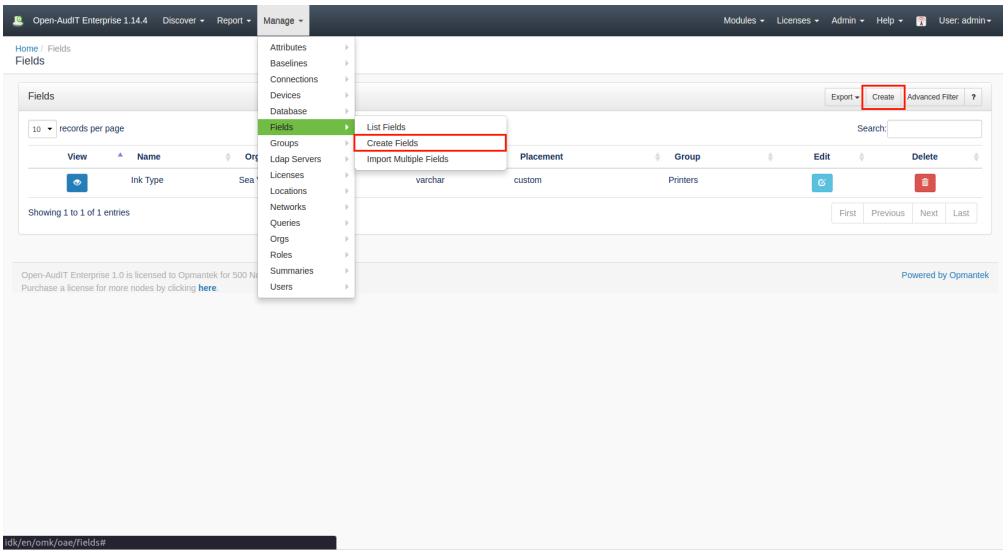
```
# DO NOT REMOVE THE LINE BELOW
# Configuration from web UI here
files[1]="/etc/init.d/"
```

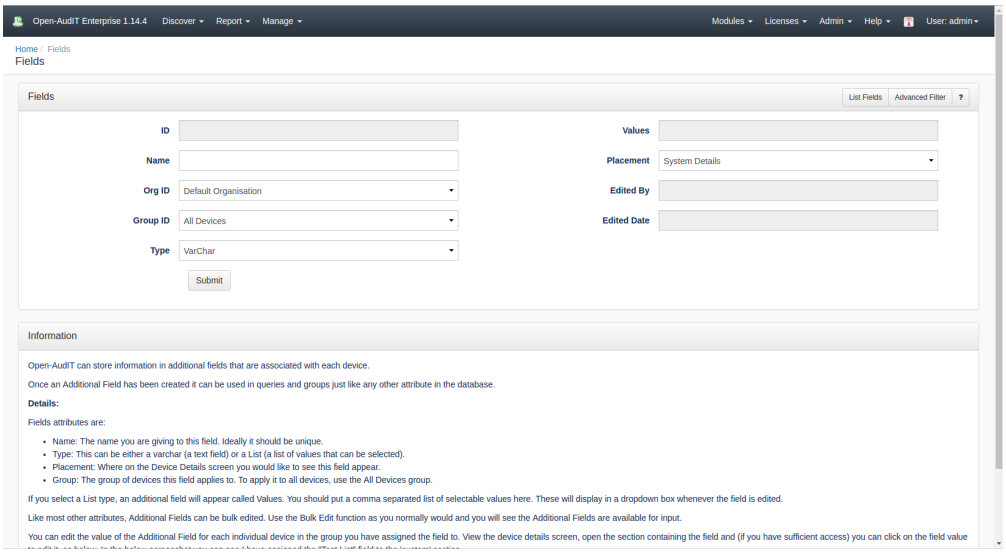
The audit script will use the files[ ] array and retrieve the file details.

If you would like to use the script outside of Discovery we have created another endpoint called "scripts". See the documentation here - Scripts.

Creating File Entries

To create an entry to track either a single file or a directory of files, use Open-AudIT Enterprise and go to menu: Discover-> Files -> Create Files. Create a file by providing values for the path (either the file or a directory with a trailing slash) and an optional description. Click Submit. Once you have created the file you will see it appear in the list at Discover -> Files -> List Files.



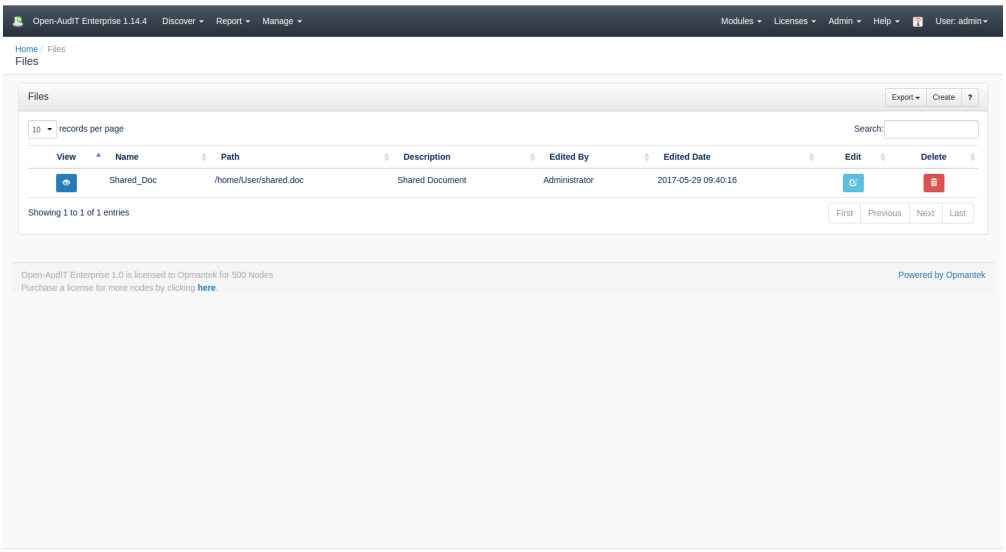


An entry will be create in the Open-AudIT database, in the "files" table.

Viewing File Details

Go to menu: Discover -> Files -> List Files.

You will see a list of files. You can view a file by clicking on the blue view icon. You can also edit or delete your file.



Database Schema

The schema for the database is below. It can also be found in the application if the user has database::read permission by going to menu: Manage -> Database -> List, then clicking on the "files" table.

```
CREATE TABLE `files` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(200) NOT NULL DEFAULT '',
  `org_id` int(10) unsigned NOT NULL DEFAULT '1',
  `path` text NOT NULL,
  `description` text NOT NULL,
  `edited_by` varchar(200) NOT NULL DEFAULT '',
  `edited_date` datetime NOT NULL DEFAULT '2000-01-01 00:00:00',
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Example Database Entry

Files are stored in the database in the "files" table. A typical entry will look as below (for a single file).

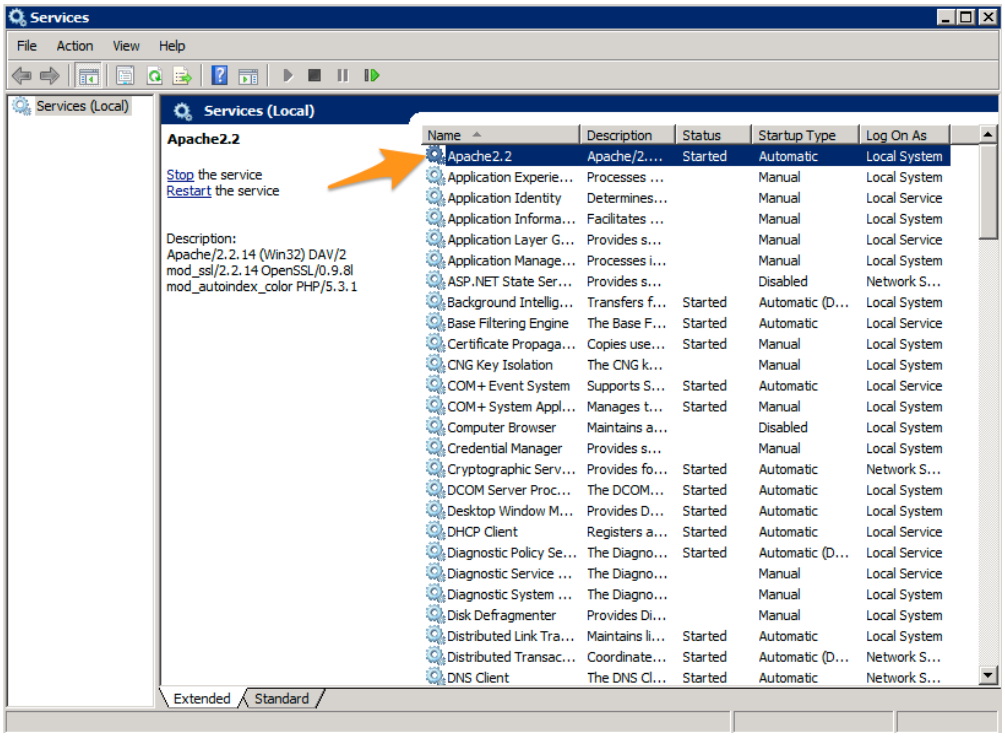
```
              id: 48
       system_id: 8
         current: y
      first_seen: 2016-08-04 00:56:35
       last_seen: 2016-08-04 00:56:35
        files_id: NULL
            name: single
       full_name: /etc/init.d/single
            size: 590
       directory: /etc/init.d
            hash: 27579d05edbd1b71307d2059a6c3370a00823c54
    last_changed: 2014-03-13 11:33:14
meta_last_changed: 2014-08-22 17:42:38
      permission: 755
           owner: root
           group: root
            type:
         version:
           inode: 5374232
```

Enabling the Feature Under Windows

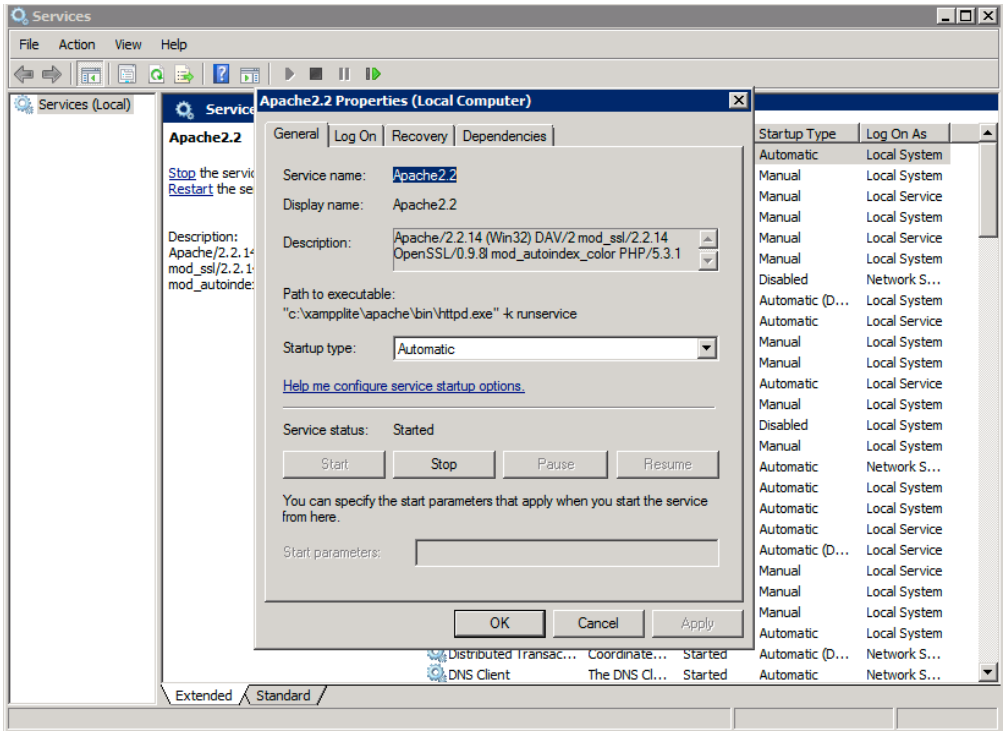
There is no need to do anything if you're running Open-AudIT on a Linux server.

Windows clients are just fine and require no special actions, however.... to enable this feature the audit script must be run locally on the target Windows system. It cannot be run remotely as we do with WMI calls when running the audit script on one Windows machine, while targeting a second Windows machine. To do this we need to copy the audit script to the target Windows machine and then run it. Unfortunately the service account that Apache runs under is the Local System account. This account has no access to remote (network based) resources. To work around this issue the service must be run under another account. It is easiest to just use the local Administrator account, but you can try any account you like as long as it has the required privileges. The Local System account has as much local access as the local Administrator account.

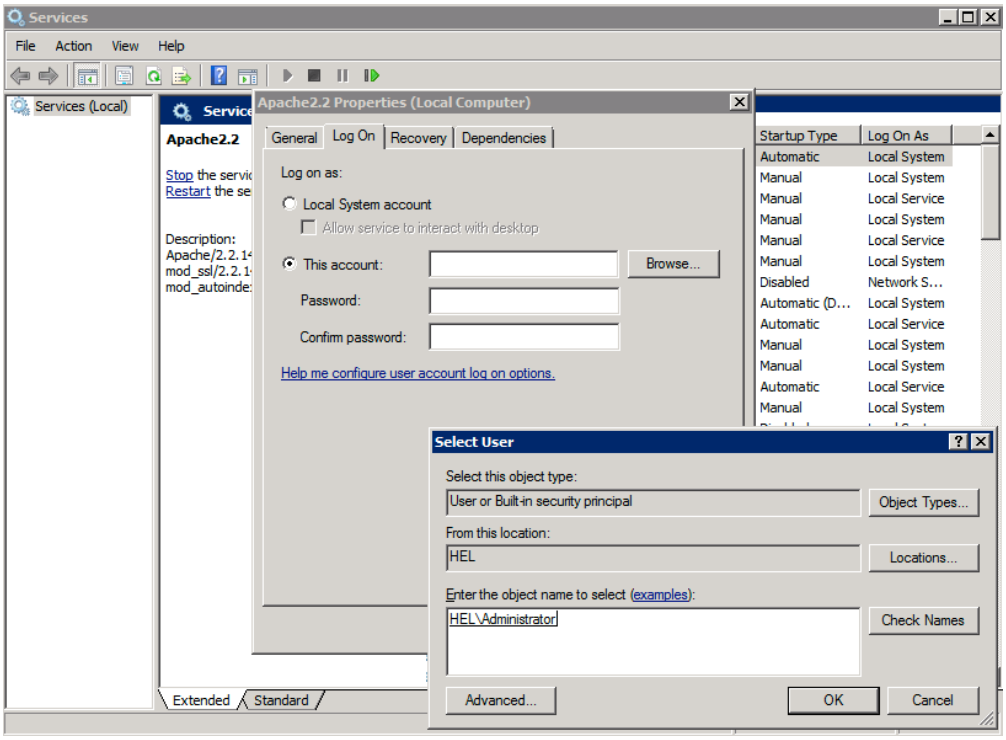
Navigate to the Service list.



Double click the apache 2.2 service.



Click the tab for logon and then click the "this account" option. You might want to click the Browse button and search for the account. Click OK and then restart the service.



Apache will now be running under an account with network access and Open-AudIT will now be able to copy the audit script to the target Windows machine and run it, hence retrieving file details.

API / Web Access?

You can access the /files collection using the normal Open-AudIT JSON based API. Just like any other collection. Please see the API documentation for further details.

API Routes

Request Method	ID	Action	Resulting Function	URL Example	Notes	Example Response
GET	n		collection	/files	Returns a list of files.	files_collection.json
GET	y		read	/files/{id}	Returns a file's details.	files_read.json
PATCH	y		update	/files/{id}	Update an attribute of a file entry.	files_patch.json
POST	n		create	/files	Insert a new file entry.	files_create.json
DELETE	y		delete	/files/{id}	Delete a file entry.	files_delete.json

Web Application Routes

Only available under Open-AudIT Enterprise

Request Method	ID	Action	Resulting Function	URL Example	Notes
GET	n	create	create_form	/files/create	Displays a standard web form for submission to POST /files.
GET	y	update	update_form	/files/{id}/update	Show the script details with the option to update attributes using PATCH to /files/{id}