

How to Schedule Discovery using a script

- [Introduction](#)
- [Argument options](#)
- [Argument Types](#)
- [Debugging](#)
- [Help](#)
- [Scheduling using Cron](#)
- [Logging](#)

NOTE - This has been made largely redundant by the inclusion of [Scheduled Task setup](#) in Open-Audit Enterprise v1.5.1.

Introduction

To enable discovery on a regular basis, you need to be able to schedule the Open-Audit discovery to run at the required frequency, e.g. every day. To do this using a series of discover a subnet using a script can be useful to setup scheduled Discovery runs. These can be created using the cron scheduler on Linux. The script named `discover_subnet_cron.sh` is designed for this purpose and is included with Open-Audit Enterprise. You can supply individual arguments on the command line or set them inside the script. Both ways have benefits.

Argument options

If you set the arguments inside the script:

- Credentials will not appear on the command line or in the cron schedule.
- Multiple copies of the script can be created with different arguments for each.

If you set the arguments on the command line:

- Only a single script is required for (potentially) multiple Discovery runs.

The script arguments can be set on the command line by (for example):

```
./discover_subnet_cron.sh option=value
```

Argument Types

The script has two basic types of arguments - required and optional.

The required arguments are:

- **openaudit_user**, this is the username of a valid Open-Audit administrator level user (set to 'admin' by default). This is not the unix user running the script.
- **openaudit_pass**, the corresponding password for the above user (set to 'password' by default).
- **openaudit_url**, this is usually left at the default supplied value (set to 'http://localhost/open-audit/index.php/discovery/discover_subnet' by default).
- **local_address**, this is the ip address of the Open-Audit server upon which the Discovery is run. This must be an address visible to remote devices if they are to be audited using an audit script (`audit_windows.vbs` or `audit_linux.sh`).
- **subnet**, this is the nmap style subnet (no default). Valid examples are:
 - 192.168.0.1 (a single address)"
 - 192.168.1.2/32 (a single address with mask)"
 - 192.168.3.0/24 (a 24 bit mask - 192.168.3.0 to 192.168.3.255)"
 - 198.168.0-255.1-127 (a range of ip addresses)

The optional arguments are:

- **snmp_community**, the SNMP community string for any devices discovered on this subnet (set to 'public' by default).
- **ssh_user**, the SSH username for any devices discovered on this subnet (no default).
- **ssh_password**, the password for the above SSH user (no default).
- **windows_user**, the Windows username to be used in this discovery run. This should have local administrator rights on any discovered Windows PCs (no default).
- **windows_domain**, the Windows domain for the above user (no default).
- **windows_password**, the Windows password for the above user (no default).
- **debugging**, the command line output level. 0 = none, 1 = debug (set to '1' by default).
- **quiet**, no need to specify a value. using this is the equivalent to `debugging=0`.
- **verbose**, no need to specify a value. using this is the equivalent to `debugging=1`.
- **syslog**, if set, will log to this file (set to `/usr/local/open-audit/other/open-audit.log` by default).

Even though the credentials are optional, not providing them will limit Discovery to only those that are provided. hence, providing no Windows credentials will prevent an audit script from being run upon any Windows computers, etc.

Debugging

If the option is set "debugging=1" or the "verbose" option is provided, command line output will occur. If "debugging=0" or "quiet" are set, no output will occur.

If output is set to occur, a list of variables will be provided, along with any generated log file entries.

A typical example is below:

```
./discover_subnet_cron.sh verbose subnet=192.168.0.1/32 local_address=192.168.0.8
-----
Open-Audit Subnet Discovery cron script
(c) Opmantek, 2014.
-----
ARGUMENTS
-----
Open-Audit User: admin
Open-Audit Password: password
Open-Audit URL: http://localhost/open-audit/index.php/discovery/discover_subnet
Subnet: 192.168.0.1/32
Local Address: 192.168.0.8
Debugging: 1
Syslog: /usr/local/open-audit/other/open-audit.log
Help: n
SNMP Community: public
SSH User:
SSH Password:
Windows User:
Windows Password:
Windows Domain:
DEBUG
-----
Logged: Discovery for 192.168.0.1/32 cron job submission
Logged: Discovery for 192.168.0.1/32 cron job completed
```

Help

If help is invoked with "help" or "help=y", or if an incorrect command line option is provided, output to the console will occur providing an overview of the script and it's options. If an incorrect argument is provided, debugging will also be enabled. The output is below:

```

./discover_subnet_cron.sh verbose help
-----
Open-Audit Subnet Discovery cron script
(c) Opmantek, 2014.
-----
ARGUMENTS
-----
Open-Audit User: admin
Open-Audit Password: password
Open-Audit URL: http://localhost/open-audit/index.php/discovery/discover_subnet
Subnet:
Local Address:
Debugging: 1
Syslog: /usr/local/open-audit/other/open-audit.log
Help: y
SNMP Community: public
SSH User:
SSH Password:
Windows User:
Windows Password:
Windows Domain:
HELP
-----
This script should be used on a Linux based computer to discover hosts in a subnet. This script is designed to
be called by cron or run directly from the command line. Wget is the only prerequisite for this script to
function correctly. Valid command line options are below (items containing * are the defaults) and should take
the format name=value (eg: debugging=1). The special options for help, quiet and verbose do not require an
argument. "./discover_subnet_cron.sh help" is valid.
openaudit_user
    *admin - The Open-Audit user running this script.
openaudit_password
    *password - The password for the above user.
openaudit_url
    *http://localhost/open-audit/index.php/discovery/discover_subnet - The form URL in Open-Audit we are
submitting to.
help
    y - Display this help output.
    *n - Do not display this output.
subnet
    - The format of the subnet is specified in standard Nmap syntax. The following are valid examples:
    - 192.168.0.1 (a single address)
    - 192.168.1.2/32 (a single address with mask)
    - 192.168.3.0/24 (a 24 bit mask - 192.168.3.0 to 192.168.3.255)
    - 198.168.0-255.1-127 (a range of ip addresses)
local_address
    - The external ip of this Open-Audit server.
snmp_community
    *public - The SNMP community string to be used in this discovery run.
ssh_user and ssh_password
    - The SSH credentials to be used in this discovery run.
windows_user and windows_pass and windows_domain
    - The Windows credentials to be used in this discovery run.
syslog
    */usr/local/open-audit/other/open-audit.log - If set the script will log to this file.
debugging
    *1 - If set to 1, will output details on the command line. If set to 0, no output will occur.
verbose
    - Equivalent to debugging=1
quiet
    - Equivalent to debugging=0

```

Example

A valid example to run Discovery on the 192.168.0.1/24 subnet using some defaults would be:

```
./discover_subnet_cron.sh subnet=192.168.0.1/24 ssh_user=root ssh_password=rootpass snmp_community=snmpsecret  
windows_user=administrator windows_password=testpass windows_domain=open-audit.com
```

Scheduling using Cron

Because of file permissions, it may be easiest to set the crontab schedule using root. This is not necessary though and the script can be run (assuming file execute other permission is set) by any valid user, it is advisable that these files be restricted in their access, as they contain passwords.

A valid crontab line to schedule the Discovery job to run at 1:00 am each day would look like:

```
0 1 * * * /usr/local/open-audit/other/discover_subnet_cron.sh subnet=192.168.0.1/24 ssh_user=root  
ssh_password=rootpass snmp_community=snmpsecret windows_user=administrator windows_password=testpass  
windows_domain=open-audit.com
```

As previously stated, the options above may be set in a copy of the script which can be copied and renamed as required.

If you wanted to include multiple subnets you might like to use an intermediate script like `discover_many_subnets.sh`, and schedule that in cron. For example `discover_many_subnets.sh` might include:

```
#!/bin/sh  
SUBNETS="192.168.1.0/24 192.168.2.0/24 192.168.10.0/24 192.168.42.0/24 172.16.1.0/24"  
for SUBNET in $SUBNETS  
do  
    /usr/local/open-audit/other/discover_subnet_cron.sh subnet=$SUBNET ssh_user=root ssh_password=rootpass  
snmp_community=snmpsecret windows_user=administrator windows_password=testpass windows_domain=open-audit.com  
done
```

Then in the crontab schedule the discovery using the intermediate script:

```
0 1 * * * /usr/local/open-audit/other/discover_many_subnets.sh
```

You might like to run infrastructure subnets at night, and user subnets during the day, there are many options available with this flexible solution.

Logging

Once a discovery job has been run, an entry into the standard Open-Audit log will be created.

This is viewable by Menu -> Admin -> Log -> View Log, inside Open-Audit or Menu -> Views -> Log inside Open-Audit Enterprise.

The standard file location is `/usr/local/open-audit/other/open-audit.log`. Typical entries into the log for a discovery run will look like:

```
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job STARTED  
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job COMPLETED successfully
```

If a submission to the Open-Audit server fails, you will see an entry such as:

```
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job STARTED  
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job FAILED wget submission
```

If any required arguments are not provided, a log line will be created and the script will abort. Example:

```
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job STARTED  
Jul 09 15:05:50 desktop 26496 S:discover_subnet_cron U:mark Discovery for 192.168.0.15 cron job FAILED subnet to discover not provided
```

If debugging is enabled, any lines written to the log file will also be printed to the screen.

An example of a complete Discovery run log is below. This includes the cron job submission, the web page acceptance, the discover script, the web page processing of the discover script result and the update of the database.

```
Jul 09 16:39:33 oa.opmantek.com 10853 C:discovery F:process_subnet Deleting credential set for 192.168.0.2 submitted on 2014-07-09 16:39:23.  
Jul 09 16:39:31 oa.opmantek.com 11801 C:discovery F:process_subnet Completed processing 192.168.0.2 (System ID 320).
```

Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	SNMP credential update for 192.168.0.2 (System ID 320).
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	SNMP update for 192.168.0.2 (System ID 320).
Jul 09 16:39:28	oa.opmantek.com	11801	H:snmp_helper	F:get_snmp	192.168.0.2 SNMP v2c scanned.
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	Attempting SNMP discovery on 192.168.0.2.
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	SSH Status: true 192.168.0.2.
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	SNMP Status: true 192.168.0.2.
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	WMI Status: false 192.168.0.2.
Jul 09 16:39:28	oa.opmantek.com	11801	C:discovery	F:process_subnet	Start processing 192.168.0.2.
Jul 09 16:39:28	oa.opmantek.com	23241	S:discover_subnet	U:apache	Discovery for 192.168.0.2 submitted at 2014-07-09 16:39:23 completed
Jul 09 16:39:28	oa.opmantek.com	23241	S:discover_subnet	U:apache	Submitting online 192.168.0.2
Jul 09 16:39:23	oa.opmantek.com	23234	S:discover_subnet_cron	U:marku	Discovery for 192.168.0.2 cron job COMPLETED successfully
Jul 09 16:39:23	oa.opmantek.com	23241	S:discover_subnet	U:apache	Scanning ip address 192.168.0.2
Jul 09 16:39:23	oa.opmantek.com	23241	S:discover_subnet	U:apache	Discovery for 192.168.0.2 submitted at 2014-07-09 16:39:23 starting
Jul 09 16:39:23	oa.opmantek.com	12143	C:discovery	F:discover_subnet	U:Administrator Discovery submitted for 192.168.0.2.
Jul 09 16:39:23	oa.opmantek.com	23234	S:discover_subnet_cron	U:marku	Discovery for 192.168.0.2 cron job STARTED