AutoOpenVAS

Version 0.3

user manual

07.08.2020

automate OpenVAS or integrate it into your own monitoring/scripting solution

dev@muab.org

Table of content

About AutoOpenVAS **3**

1 Setup **4**
1.1 Requirements . . . 4
1.2 Integration . . . 4
1.2.1 Passive . . . 4
1.2.2 Active . . . 4
1.3 Installation . . . 7
1.4 Configuration . . . 7
1.5 Populate inventory . . . 9

2 Usage **11**
2.1 Run AutoOpenVAS . . . 11
2.2 Parameters . . . 12

3 Results **13**
3.1 JSON file . . . 13
3.2 Console . . . 14

4 FAQ **15**

Legal notices **17**
Warranty . . . 17
Licence Agreement . . . 17

About AutoOpenVAS

AutoOpenVAS is an interface for automated processing and managing of Greenbone's Open Vulnerability Assessment Scanner[1] (OpenVAS), a vulnerability scanner including 50.000 individual tests.

- ☑ add/remove targets
- ☑ add/remove tasks
- ☑ scan for machines automatically
- ☑ rotate single tasks
- ☑ extract and process results
- ☑ return severity per machine
- ☒ unicorns

Most recent version is available at GitHub[2].

[1] https://www.greenbone.net/community-edition/
[2] https://github.com/TMagerl/AutoOpenVAS

1 Setup

1.1 Requirements

- Python 3.x
- OpenVAS 9.x
- arping (optional, see section 2.2 at page 12)
- nmap (optional, see section 1.5 at page 9)

1.2 Integration

There are two possible usecases which can be combined.

1.2.1 Passive

Use this mode if you are using any monitoring solution or scripts.

AutoOpenVAS is fed with a list of machines provided by your monitoring solution and returns correlating data which then can be visualized or otherwise processed (see figure 1 at page 5). Please see section 1.5 at page 9 for details how to import jobs.

1.2.2 Active

Use this mode (see figure 2 at page 6) if you just want the jobs to be managed by AutoOpenVAS or you have a highly dynamic network enviroment.

AutoOpenVAS scans a given subnet to populate OpenVAS automatically. See section 1.5 at page 9 for further information.

Hint: A datafile is created also in active mode, so you can still use your monitoring without having to export your machine list.

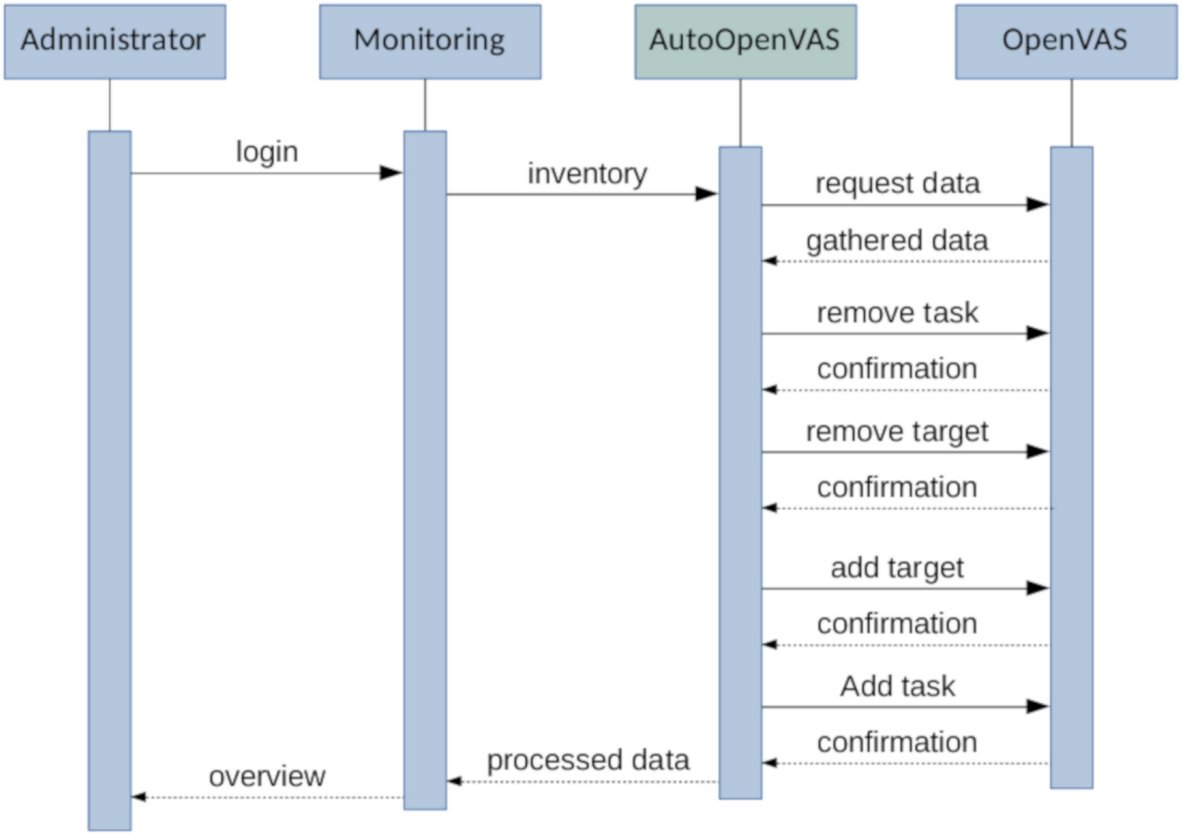

Abbildung 1: passive integration

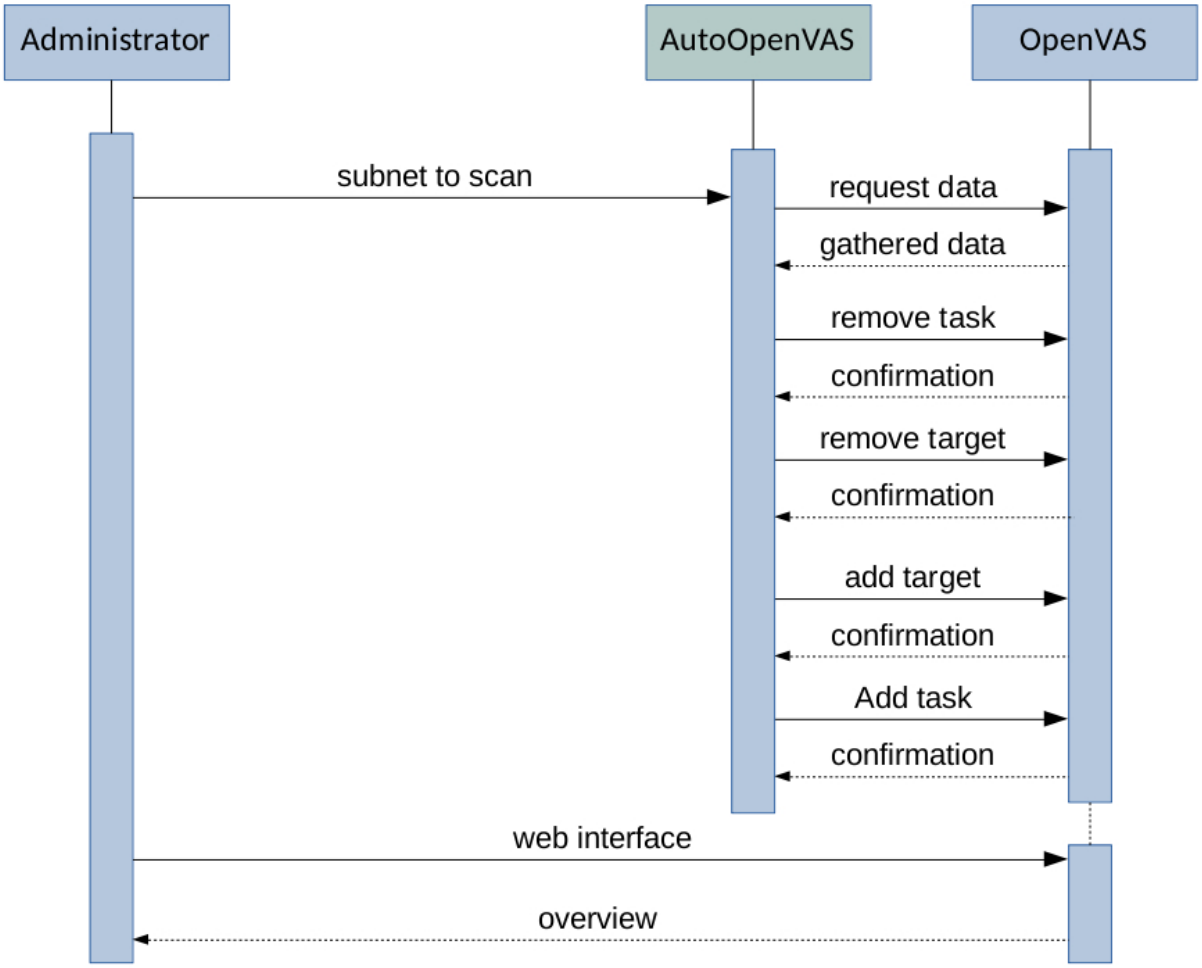

Abbildung 2: active integration

1.3 Installation

Simply copy the AutoOpenVAS folder to any directory, *'/opt/'* is recommended.

arping needs to be installed if you want to verify your machine before a test, see section 2.2 at page 12.

namep needs to be installed if you want to let AutoOpenVAS scan your network for machines, see section 1.2.2 at page 4.

1.4 Configuration

Rename the sample config file (see listing 1 at page 7) in your AutoOpenVAS directory and edit it to your needs.

Hint: Don't forget to grant read privileges.

```
{
   "admin": "admin",
   "passwd": "topsecret",
   "openvas_ip": "127.0.0.1",
   "openvas_web_port": "9392",
   "openvas_omp_port": "9390",
   "default_scan_config_id": "74db13d6-7489-11df-91b9-002264764cea",
   "default_portlist_id": "9ddce1ae-57e7-11e1-b13c-406186ea4fc5",
   "job_source": "/opt/autoopenvas/import.json",
   "data_file": "/opt/autoopenvas/export.json",
   "clean_up_threshold_days": 90
}
```

listing 1: Sample - auto-openvas.conf

- **admin**
  Your admin login for OpenVAS web interface.
- **passwd**
  Your admin password.
- **openvas_ip**
  IP of OpenVAS instance you want to use.
- **openvas_web_port**
  OpenVAS web port (default: 9392)
- **openvas_omp_port**
  OpenVAS omp port (default: 9390)
- **default_scan_config_id**
  id of scan config for new tasks found at

  Configuration -> *Port Lists* -> your choice (see figure 3 at page 8)

- **default_portlist_id**
  id of port list for new tasks found at

  Configuration -> *Port Lists* -> your choice (see figure 3 at page 8)

- **result_file**
  results are stored to this JSON file

- **job_source**
  location of job file, can also be a folder, see section 1.2.2 at page 4

- **clean_up_threshold_days**
  remove anything where no report was created for last x days from data file

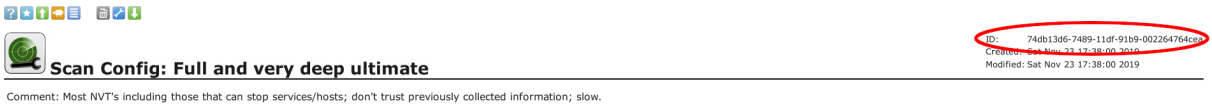

Abbildung 3: OpenVAS web interface

1.5 Populate inventory

scan subnet for machines

If you do not have the option to export an inventory list or you also want to catch *dynamically created machines* (e.g. test-servers) you can use the scan parameter, see section 2.2 at page 12. Any (new) machine found will get a target and task in OpenVAS.

Hint: This method can be used in addition to importing a machine list.

import machine list

job_source (see section 1.4 at page 7) defines the source of the machine inventory you provide.

job_source can also be a folder - a single JSON file per machine is expected then. Might be useful if you have multiple sections in your network with different admins responsible.

Mandatory information is IP and MAC address of the machine.

Hint: You can use the verify switch if there's the chance of changing IP addresses. See section 2.2 at page 12 for further details.

comment is optional and does not affect the process in any way, though it might help for better overview while inspecting the OpenVAS web interface.

skip is optional too and is considered as *false* by default. Helpful if you use the scan option and want to prevent particular machines from being added to OpenVAS. Set to *true* in this case.

Attention: never add URIs, plain IPs only. OpenVAS stores results under the resolved IP which might lead to unexpected results.

Import folder

This is the preferred way, if you have multiple admins who manage their machines on their own. In case you manage your machines centralized it is suggested to jump to section 1.5 at page 10.

Every single machine is configured by an own file, see listing 2 at page 9.

```
{
  "ip": "192.168.x.x",
  "mac": "XX:XX:XX:XX:XX:XX",
  "comment": "maintainer: Hans-Peter",
  "skip": true
}
```

listing 2: sample - input.json

Import file

You can create or generate a single file with all machines.

The example listing 3 at page 10 presets two machines:

- Machine 1 will never be added to OpenVAS, even if found by *scan*.
- Machine 2 will be added, even if (currently) not found. It gets the comment on the OpenVAS web interface.

```
[
  {
    "ip": "192.168.x.1",
    "mac": "XX:XX:XX:XX:XX:XX",
    "skip": true
  },
  {
    "ip": "192.168.x.2",
    "mac": "XX:XX:XX:XX:XX:XX",
    "comment": "maintainer: Hans-Peter"
  }
]
```

listing 3: sample - bulk-input.json

Attention: A list of machines needs to be in brackets.

2 Usage

2.1 Run AutoOpenVAS

You can start AutoOpenVAS from terminal after configuration, though the main purpose is to be run by cronjobs.

For a simple cron job see listing 4 at page 11. This example results in scanning for (new) machines and testing next machine every hour.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
#m h   dom mon dow command

* */1 * * * /opt/AutoOpenVAS/AutoOpenVAS.py -scan 192.168.1.0/24 -run
```

listing 4: simple cron job sample

For a more advanced cron job see listing 5 at page 11. Results in scanning for (new) machines every hour, testing every two hours during work time, hourly over night and every 30 minutes on weekends.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
#m    h      dom mon dow command

 *    8-16/2 *   *   1-5 /opt/AutoOpenVAS/AutoOpenVAS.py -run
 *    17-7/1 *   *   1-5 /opt/AutoOpenVAS/AutoOpenVAS.py -run
 */30 *      *   *   6-7 /opt/AutoOpenVAS/AutoOpenVAS.py -run

 *    */1    *   *   * /opt/AutoOpenVAS/AutoOpenVAS.py -scan 192.168.1.0/24
```

listing 5: advanced cron job sample

Attention: Read and write privileges are needed for import and export paths.

Hint: All actions will be submitted to your local syslog server. See section 2.2 at page 12 for more details.

2.2 Parameters

-v

Use *-v* if you want more details in your syslog.

-vv for insane debug spam.

-print

Use *-print* to print a human readable table to console. See section 3.2 at page 14. Can be combined with *-verify*.

-run

Use *-run* to start the next vulnerability check. The candidates will be rotating, so each time you start AutoOpenVAS with this argument, the next machine in the row will be tested.

After a test is started AutoOpenVAS waits 2 minutes and checks again if task is running. If the task was aborted for any reason it will continue with the next machine.

-scan [subnet]

scan the given subnet for machines.

Populate OpenVAS automatically with all machines of your network. See section 1.2.2 at page 4.

Notation is the network address of the subnet you want to scan including CIDR (e.g. *-scan 192.168.1.0/24*).

Attention: *nmap* has to be installed for this to work.

-verify

verify MAC address before scan with *-verify*. In case a MAC/IP address pair doesn't match the configured (see section 1.5 at page 9) one (any more), the job will be skipped as *failed*.

Attention: *arping* has to be installed for this to work.

3 Results

3.1 JSON file

Hint: If you do not use any monitoring solution or scripts you can skip this part. Simply visit the OpenVAS web interface to get results (see figure 2 at page 6) or use console output section 3.2 at page 14.

All gathered values are stored to the configured *export path* using JSON format. The file name corresponds to the MAC address.

```
{
  "mac": "XX:XX:XX:XX:XX:XX",
  "ip": "192.168.x.x",
  "err": 2,
  "severity": 2.6,
  "trend": "-",
  "stamp": "1970-01-01 00:00:00",
  "link": "https://192.168.x.x:9392/omp?cmd=get_report&report_id=xxxxx"
}
```

listing 6: sample - output.json

MAC and IP

Initially defined (passive mode) or found (active mode) IP and MAC address. Please see the verify MAC option on section 2.2 at page 12.

severity

OpenVAS displays the max severity value (0-10) for each machine. AutoOpenVAS returns the sum of every severity of the machine for a more meaningful value. If the machine was not available during the last test, the severity of the last successful test is returned.

stamp

This is the timestamp of the last successful test.

link

This is a direct link to the latest report of the machine. If there is no report (yet) it points to the task or target if available.

3.2 Console

You can print the results to console by using the *-print* switch.

```
MAC      IP     Verified Severity Last report Comment
XX:XX:.. 192... false    -1   2020-07-28 10:13:25 maintainer: hans
XX:XX:.. 192... true   14.0  2020-08-06 11:04:45 maintainer: peter
XX:XX:.. 192... true   10.0  2020-08-06 10:39:52 maintainer: siggie
```

listing 7: print console output

4 FAQ

Is there any log file?

- AutoOpenVAS logs to your local syslog.
- Consider using a switch to increase log level (see section 2.2 at page 12).

Do I have to configure my machines individually?

- No, see section 1.2.2 at page 4.

Can I configure my machines individually though?

- Yes, see section 1.2.1 at page 4.

Why is the time wrong?

- Times are stored as UTC, so notice your time zone.

Job files (import) cannot be read

- Files have to be strictly in JSON, see section 1.5 at page 9
- Please check privileges.

Data file (export) cannot be read

- Please check privileges.
- Can be removed if corrupted (though the auto run cycle will be reset).

Link opens task instead of report?

- If there is no report (yet), yes, see section 3.1 at page 13.

How can I see if at least one test was performed?

- If *severity* not equal to '-1', yes.

How can I see if OpenVAS tried to test, but failed?

- If *last_attempt* > *last_report*.

Resolving IP addresses leads to strange results?

- Indeed, it is not supported. Please see section 1.5 at page 9

Failed to acquire socket?

- check if OpenVAS is running
- check if OMP port is set correctly
- check if firewall/fail2ban blocks OMP port

clean_up_threshold_days does not always work?

- You might have defined a job in your *import machine list*, section 1.5 at page 9.

Can I exclude particular machines from scan?

- You can add a job for this machine and set *skip*, see section 1.5 at page 9.

Legal notices

Warranty

The software is licensed “as-is”. You bear the risk of using it. I give no express warranties, guarantees or conditions. You may have additional consumer rights under your local laws which this agreement cannot change. To the extent permitted under your local laws, I exclude the implied warranties of merchantability, fitness for a particular purpose and non-infringement.

Licence Agreement

[1] http://www.gnu.org/licenses/gpl-3.0.html