

Installing OpenVAS on Kali Linux



OpenVAS Installation



OpenVAS, an **endpoint scanning** application and web application used to identify and detect vulnerabilities. It is widely used by companies as part of their **risk mitigation solutions** to quickly identify gaps in their production and even development servers or applications. This is not a complete solution, but it can help you fix **common security vulnerabilities** that may not be discovered.

The condition of Greenbone mode is open (APEVALV) from infected chemistry (GVM) of the quality of the storage and the GitHub area. it is used in the Greenbone Security Manager device and is a comprehensive scan. An engine that runs an advanced and constantly updated Network Vulnerability Test Package (NVT).

Prepare Kali Linux for the installation of OpenVAS

Unless you have already done so, make sure that the ***Kali Linux is up to date*** and ***install the latest Kali Linux***. You automatically download the latest rules, create admin users, and start the various services. Depending on bandwidth and computer resources, this may take a while.

- ***sudo apt update*** — or use *sudo apt-get update*

```
(hassen@hannachi)-[~]
$ sudo apt-get update
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.3 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [121 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [261 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [194 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [885 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [33.0 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [16.8 kB]
Fetched 68.8 MB in 33s (2,115 kB/s)
Reading package lists... Done
```

```
sudo apt-get update
```

- ***sudo apt upgrade -y***

```

(hassen@hannachi)-[~]
$ sudo apt upgrade
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done

The following packages were automatically installed and are no longer required:
  libboost-dev libboost1.83-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
  libpython3.12 libpython3.12-dev libxsimd-dev python3-all-dev python3-beniget python3-gast
  python3-pythran python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.

The following NEW packages will be installed:
  libgck-2-2 libgcr-4-4 libjq1 libonig5 python3-pyasyncore

The following packages will be upgraded:
  adwaita-icon-theme autopsy bind9-dnsutils bind9-host bind9-libs binutils binutils-common
  binutils-x86-64-linux-gnu binwalk colord colord-data console-setup console-setup-linux debconf
  debconf-i18n dnsmasq-base exploittb firefox-esr firmware-linux-free fontconfig fontconfig-config
  fonts-lyx gdal-data gdal-plugins geoip-database gir1.2-gstreamer-1.0 gir1.2-nm-1.0 gir1.2-vte-2.91
  glib-networking glib-networking-common glib-networking-services go-l2tp gsettings-desktop-schemas
  gstreamer1.0-gl gstreamer1.0-libav gstreamer1.0-plugins-bad gstreamer1.0-plugins-base
  gstreamer1.0-plugins-good gstreamer1.0-x gvfs gvfs-backends gvfs-common gvfs-daemons gvfs-fuse
  gvfs-libs iputils-ping isc-dhcp-client isc-dhcp-common iso-codes keyboard-configuration kmod
  libadwaita-1-0 libappstream5 libatkmm-1.6-1v5 libaudio2 libavif16 libbinutils libboost-dev
  libbssn-1.0-0 libc-ares2 libcapstone-dev libcapstone4 libcolord2 libcolorhug2 libcompress-raw-lzma-perl
  libctf-nobfd0 libctf0 libdavid7 libdaxctl1 libeac3 libencode-perl libfontconfig1 libgdal34
  libgdal-common libgdal22 libgprofng0 libgstreamer-gli1.0-0 libgstreamer-plugins-bad1.0-0
  libgstreamer-plugins-base1.0-0 libgstreamer1.0-0 libgtk-layer-shell libgtk-2.0-11 libgtk2-common
  libhogweed6 libkmod2 libldb2 liblua5.4-0 libmanette-0.2-0 libmd4c0 libmjpegutils-2.1-0 libmongoc-1.0-0
  libmosquitto libmousepad0 libmpeg2encpp-2.1-0 libmpfr6 libmplex2-2.1-0 libmtdev1 libmujs3 libndctl6
  libnet-dns-perl libnetcdf19 libnettle8 libnettp3-3 libnm0 libnpt0 libnspr4 libnss3 libnvm1

```

```
sudo apt upgrade
```

- ***sudo apt dist-upgrade -y***

```
(hassen@hannachi)-[~]
$ sudo apt dist-upgrade -y
[sudo] password for hassen:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
Calculating upgrade ... Done
The following packages were automatically installed and are no longer required:
  libboost-dev libboost1.83-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
  libpython3.12 libpython3.12-dev libxsimd-dev python3-all-dev python3-beniget python3-gast
  python3-pythran python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.

(hassen@hannachi)-[~]
$
```

sudo apt dist-upgrade -y

Installing OpenVAS on Kali Linux

To install Openvas and its dependencies on our Kali Linux system run the following command:

- ***sudo apt install openvas***

or use

- ***sudo apt install gvm***

```
(hassen@hannachi)-[~]
$ sudo apt install openvas
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'gvm' instead of 'openvas'
The following packages were automatically installed and are no longer required:
  libboost-dev libboost1.83-dev libopenblas-dev libopenblas-pthread-dev libopenblas0 libpython3-all-dev
  libpython3.12 libpython3.12-dev libxsimd-dev python3-all-dev python3-beniget python3-gast
  python3-pythran python3.12-dev xtl-dev
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  greenbone-security-assistant gsad gvm-tools libmicrohttpd12
The following NEW packages will be installed:
  greenbone-security-assistant gsad gvm gvm-tools libmicrohttpd12
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 5,153 kB of archives.
After this operation, 20.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

sudo apt install openvas

The next step is to run the installer, which will configure OpenVAS and download various *network vulnerability tests* (NVT) or signatures. Due to a large number of NVTs (50.000+), the setting process may take some time and consume a lot of data.

Run the following command to start the setup process:

- ***sudo gvm-setup***

```
(hassen@hannachi)-[~]
$ gvm-setup
[-] Error: /usr/bin/gvm-setup must be run as root

(hassen@hannachi)-[~]
$ sudo gvm-setup

[>] Starting PostgreSQL service
[>] Creating GVM's certificate files
[>] Creating PostgreSQL database
[*] Creating database user
[*] Creating database
[*] Creating permissions
CREATE ROLE
[*] Applying permissions
GRANT ROLE
[*] Creating extension uuid-oss
CREATE EXTENSION
[*] Creating extension pgcrypto
CREATE EXTENSION
[*] Creating extension pg-gvm
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password 'ef30a874-739e-425e-9612-615332e2e86d'.
[*] Configure Feed Import Owner
[*] Define Feed Import Owner
[*] Update GVM feeds
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/openvas/feed-update.lock
Acquired lock on /var/lib/openvas/feed-update.lock
[+] Downloading Notus files from
rsync://feed.community.greenbone.net/community/vulnerability-feed/22.04/vt-data/notus/ to /var/lib/notus
```

sudo gvm-setup

The `gvm-setup` command will take a **long time** to download all the vulnerability definitions (*Notus files, NASL files, SCAP data, CRET-Bund data, gvm data*).

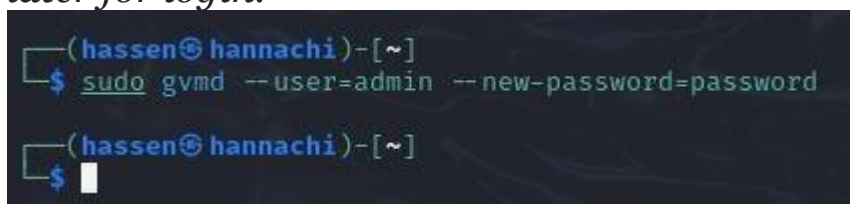
Hint: OpenVAS will also set up an **admin account** and automatically generate a **password** for this account which is displayed in the last section of the setup output.

Password reset

Did you forget to note down the password? You can change the admin password using the following commands:

- **`sudo gvmc --user=admin --new-password=password`**

Note: if you don't rest the automatically generated admin credentials [password], make sure to save a copy as you will need it later for login.



```
(hassen@hannachi)-[~]  
$ sudo gvmc --user=admin --new-password=password  
  
(hassen@hannachi)-[~]  
$
```

update admin user password
`sudo runuser -u _gvm -- gvmc -- create-user=admin2 -- new-password=12345`

To change the password of the existing user

sudo runuser -u _gvm -- gvm -- user=admin -- new-password=new_password

Verify the Installation

You can verify your installation with.

- ***sudo gvm-check-setup***

```
(hassen@hannachi)-[~]
$ sudo gvm-check-setup
gvm-check-setup 23.11.0
Test completeness and readiness of GVM-23.11.0
Step 1: Checking OpenVAS (Scanner) ...
    OK: OpenVAS Scanner is present in version 22.7.9.
    OK: Notus Scanner is present in version 22.6.2.
    OK: Server CA Certificate is present as /var/lib/gvm/CA/servercert.pem.
Checking permissions of /var/lib/openvas/gnupg/*
    OK: _gvm owns all files in /var/lib/openvas/gnupg
    OK: redis-server is present.
    OK: scanner (db_address setting) is configured properly using the redis-
server socket: /var/run/redis-openvas/redis-server.sock
    OK: the mqtt_server_uri is defined in /etc/openvas/openvas.conf
    OK: _gvm owns all files in /var/lib/openvas/plugins
    OK: NVT collection in /var/lib/openvas/plugins contains 88489 NVTs.
    OK: The notus directory /var/lib/notus/products contains 456 NVTs.
Checking that the obsolete redis database has been removed
    OK: No old Redis DB
Starting ospd-openvas service
Waiting for ospd-openvas service
    OK: ospd-openvas service is active.
    OK: ospd-OpenVAS is present in version 22.6.2.
Step 2: Checking GVM Manager ...
    OK: GVM Manager (gvm) is present in version 23.1.0.
Step 3: Checking Certificates ...
    OK: GVM client certificate is valid and present as /var/lib/gvm/CA/clientcert.pem.
    OK: Your GVM certificate infrastructure passed validation.
Step 4: Checking data ...
    OK: SCAP data found in /var/lib/gvm/scap-data.
    OK: CERT data found in /var/lib/gvm/cert-data.
```

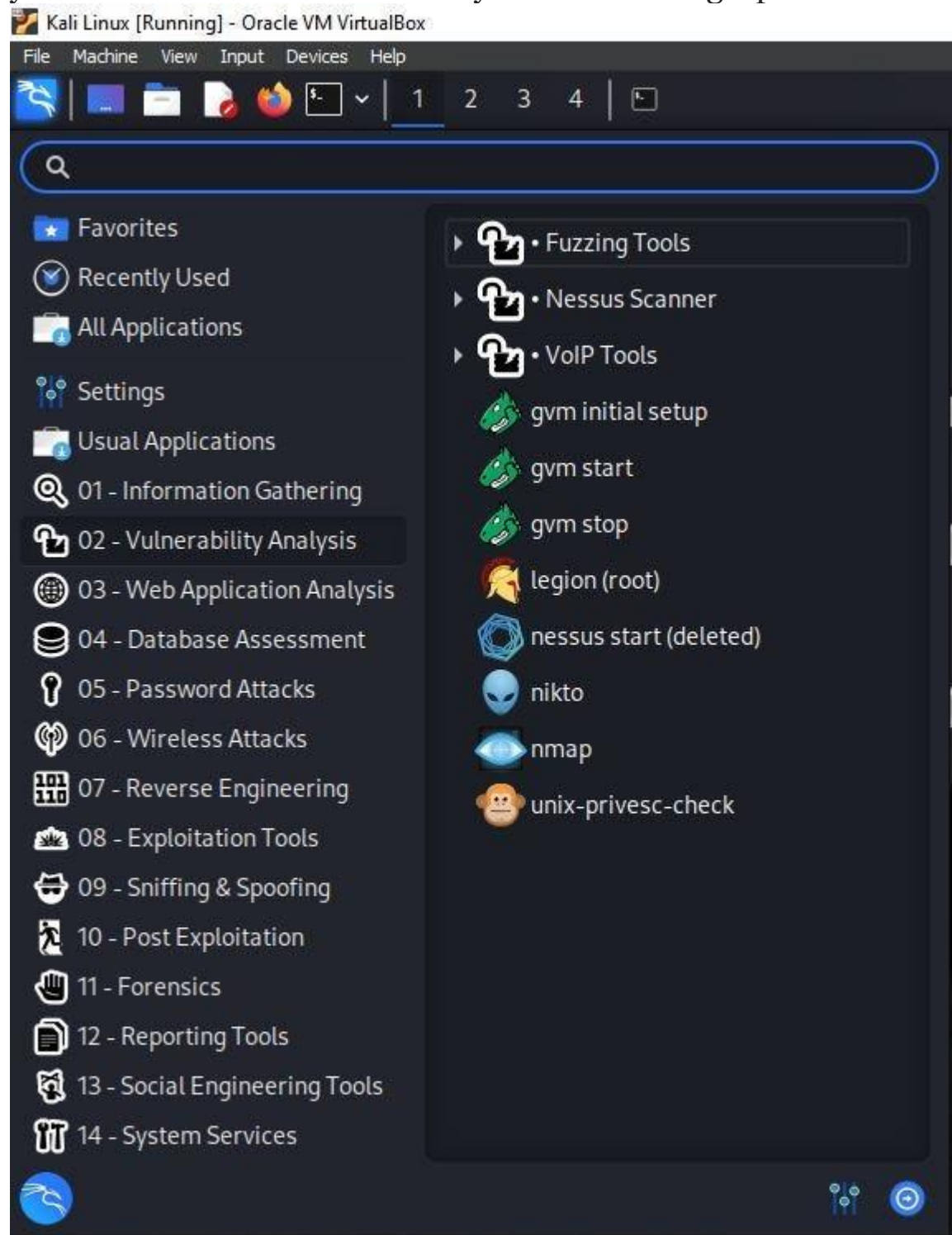
after the process is complete, we should get a confirmation that the installation was completed without error.

```
Step 9: Checking greenbone-security-assistant ...
    OK: greenbone-security-assistant is installed

It seems like your GVM-23.11.0 installation is OK.
```

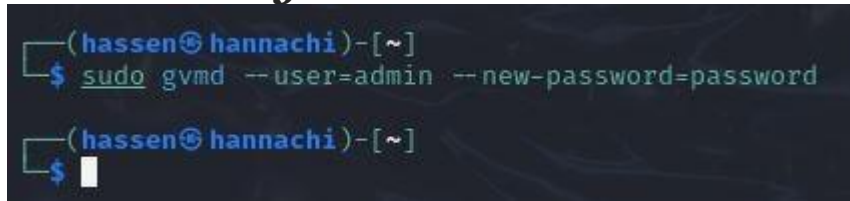
Starting and stopping OpenVAS

Before starting to install the virtual appliance, the last step I have to consider is to start and stop the OpenVAS service. OpenVAS services consume a lot of unnecessary resources, so it is recommended that you disable these services when you are not using OpenVAS.



Run the following command to start the services:

- ***sudo gvm-start***

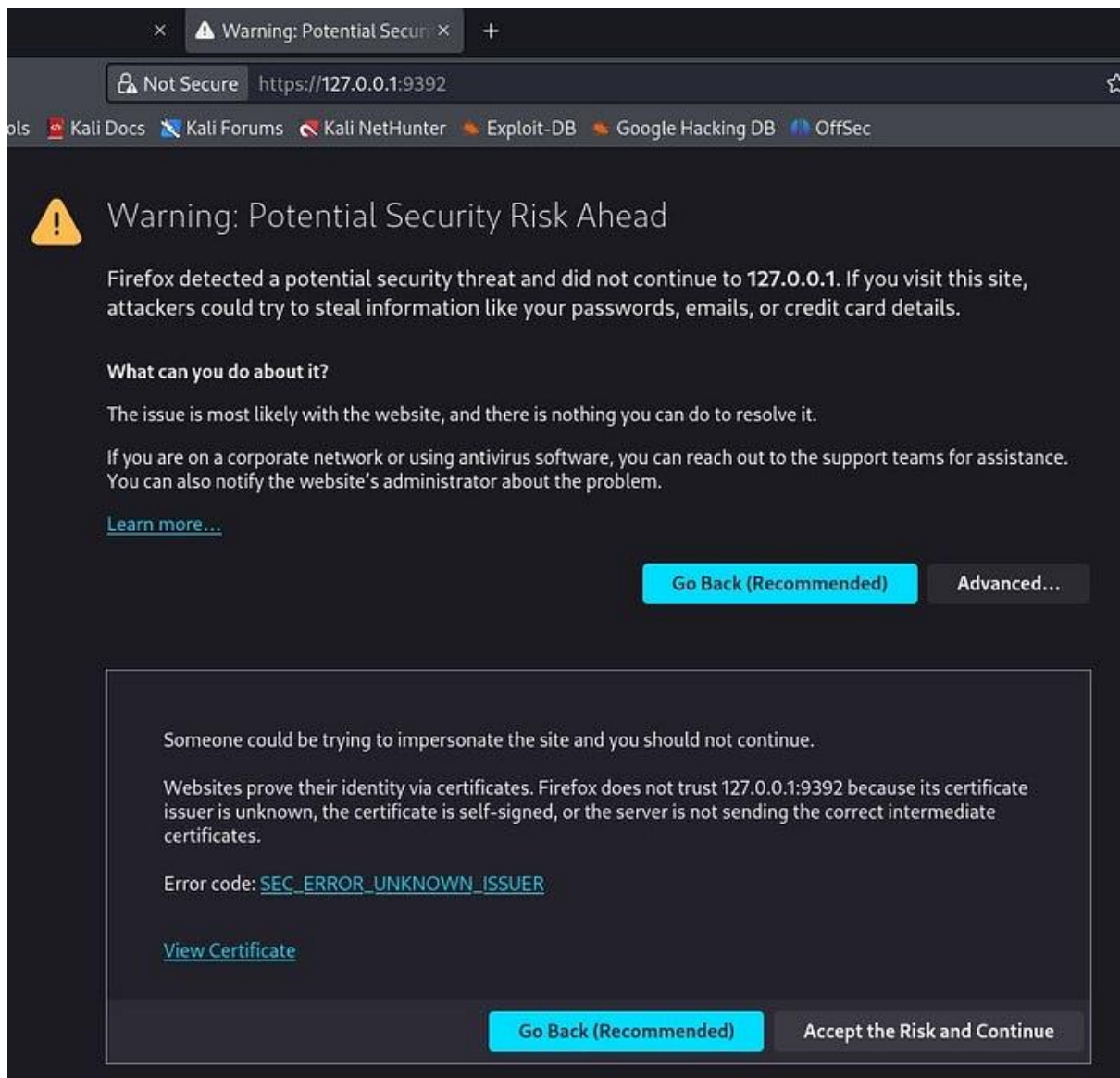
A terminal window with a dark background. The prompt is '(hassen@hannachi)-[~]'. The command 'sudo gvm-start' has been entered and executed. The output shows the command being run as root, with the user 'admin' and password 'password' specified. The prompt returns to '(hassen@hannachi)-[~]'.

```
(hassen@hannachi)-[~]  
$ sudo gvm-start  
[hassen@hannachi ~]$
```

Hint: To stop the OpenVAS services again, run: ***sudo gvm-stop***

After the configuration process is complete, all the necessary OpenVAS processes will start and the web interface will open automatically (In my case I had to open the browser manually). The web interface is *running locally* on *port 9392* and can be accessed through <https://localhost:9392>

First time you want to open this URL you will get a security warning. Click on **Advanced** and **Accept the Risk and Continue**.



The next step is to accept the self-signed certificate warning and use the automatically generated admin credentials (in my case I rest the admin password) to login on to the web interface:



Greenbone

Sign in to your account

Username

admin

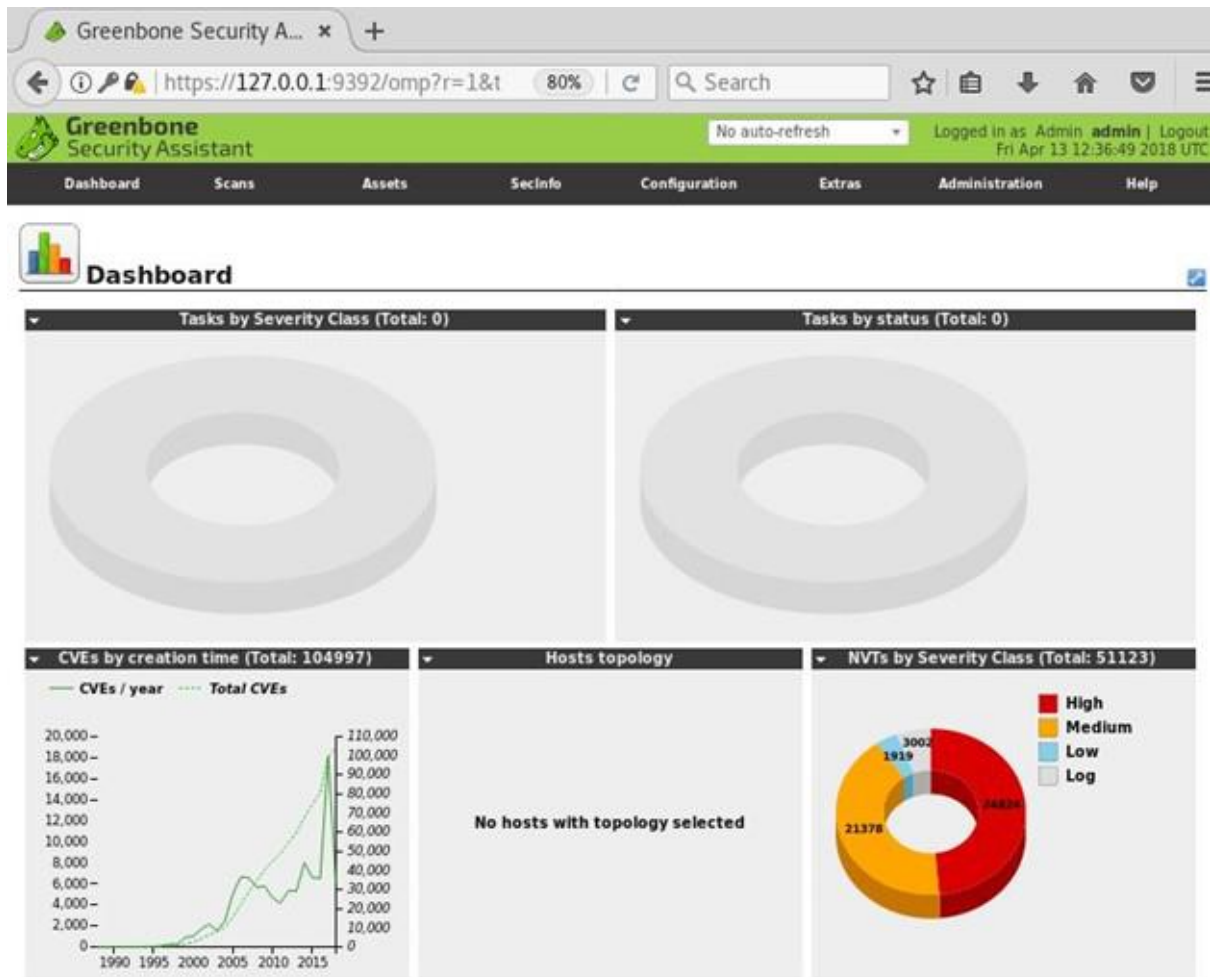
Password

.....

Sign In

Greenbone
Community

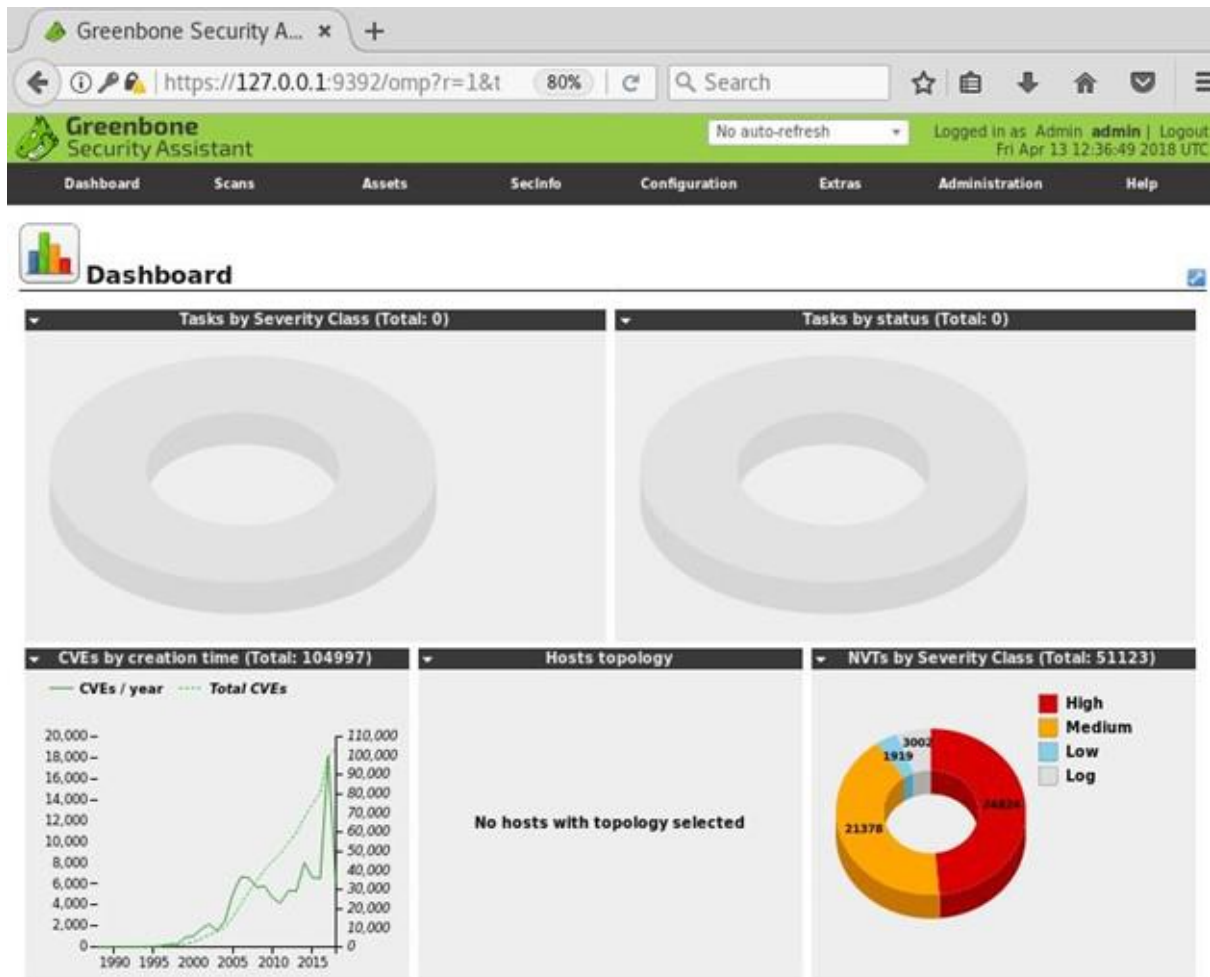
EDITION



Configuration for a new target

Begin by navigating to **Scans > Tasks** and clicking on the **purple magic wand icon** to begin the basic configuration wizard. After successfully navigating to the wizard, you should see a pop-up window similar to the one shown above. You can set up the initial scan of the local host here to make sure everything is set up correctly.

Scanning may take a while. Please allow OpenVAS enough time to complete the scan. You will then see a new dashboard for monitoring and analyzing your completed and ongoing scans, as shown below.

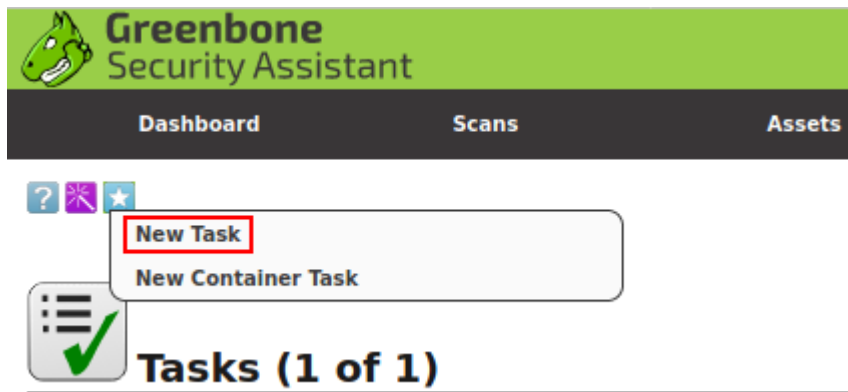


Schedule the scanning process

Now that we know everything is normal, we can take a closer look at OpenVAS and how it works. Expand the car to **scan and> start the task** of creating a scan task for the managed computer.

Creating a Task

To create a custom task, navigate to the star icon in the upper right corner of the taskbar and select New task.



After selecting "New Task" from the drop-down menu, you will see a large pop-up window with many options. We will introduce each option part and its purpose.

New Task

Name:

Comment:

Scan Targets:

Alerts:

Schedule: ☐ Once

Add results to Assets: ☒ yes ☐ no

Apply Overrides: ☒ yes ☐ no

Min QoD: %

Alterable Task: ☐ yes ☒ no

Auto Delete Reports: ☒ Do not automatically delete reports
☐ Automatically delete oldest reports but always keep newest reports

Scanner:

Scan Config:

Network Source Interface:

Order for target hosts:

Maximum concurrently executed NVTs per host:

Maximum concurrently scanned hosts:

For this task, we'll be specializing only in the Name, Scan Targets, and Scanner Type, and Scan Config. In later tasks, we will be focusing on the opposite choices for additional advanced configuration and implementation/automation.

1. **Name:** permits North American country to line the name the scan are going to be referred to as inside OpenVAS
2. **Scan Targets:** The targets to scan, can embrace Hosts, Ports, and Credentials. to make a brand new target you may follow another pop-up, this can be lined later during this task.
3. **Scanner:** The scanner to use by default will use the OpenVAS design but you'll be able to set this to any scanner of your selecting within the settings menu.
4. **Scan Config:** OpenVAS has seven totally different scan sorts you can choose from and can be used supported however you're aggressive or what info you wish to gather from your scan.

Scoping a New Target

To scope a new target, navigate to the star icon next to Scan Targets.

New Target [X]

Name:

Comment:

Hosts: ☒ Manual
☐ From file No file selected.
☐ From host assets (0 hosts)

Exclude Hosts:

Reverse Lookup Only: ☐ Yes ☒ No

Reverse Lookup Unify: ☐ Yes ☒ No

Port List: [★]

Alive Test:

Credentials for authenticated checks

SSH: on port [★]

SMB: [★]

ESXi: [★]

SNMP: [★]

Above is that the menu for configuring a replacement target. the 2 main choices you may have to be compelled to assemble are the Name and therefore the Hosts. This procedure is fairly uncomplicated and different options will solely be employed in advanced vulnerability management solutions. These are going to be lined in later tasks.

Name:

Comment:

Hosts: ☒ Manual
☐ From file No file selected.

Now that we've got our target scoped we are able to still produce our task and start the scan. When the task is created, you'll come to the

scanning management panel, wherever you'll track and execute the task. To run the task, navigate to the run icon within the operation.

Scan Configuration

Prior to launching a vulnerability scan, you should fine-tune the Scan Config that will be used, which can be done under the “Scan Configs” section of the “Configuration” menu. You can clone any of the default Scan Configs and edit its options, disabling any services or checks that you don't require. If you use Nmap to conduct some prior analysis of your target(s), you can save hours of vulnerability scanning time.

Family	NVTs selected	Trend	Select all NVTs	Actions
AIX Local Security Checks	1 of 1		<input type="checkbox"/>	
Amazon Linux Local Security Checks	748 of 748		<input type="checkbox"/>	
Brute force attacks	9 of 9		<input type="checkbox"/>	
Buffer overflow	555 of 555		<input checked="" type="checkbox"/>	
CISCO	638 of 638		<input type="checkbox"/>	
CentOS Local Security Checks	2939 of 2939		<input type="checkbox"/>	
Citrix XenServer Local Security Checks	27 of 27		<input type="checkbox"/>	

Task Configuration

Your credentials, targets, and scan configurations are setup so now you're ready to put everything together and run a vulnerability scan. In OpenVAS, vulnerability scans are conducted as “Tasks”. When you set up a new task, you can further optimize the scan by either

increasing or decreasing the concurrent activities that take place. With our system with 3GB of RAM, we adjusted our task settings as shown below.

New Task

Add results to Assets: ☒ yes ☐ no

Apply Overrides: ☒ yes ☐ no

Min QoD: 70 %

Alterable Task: ☒ yes ☐ no

Auto Delete Reports: ☒ Do not automatically delete reports
☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner: OpenVAS Default

Scan Config: Subnet-86 Full and fast ultimate

Network Source Interface: eth0

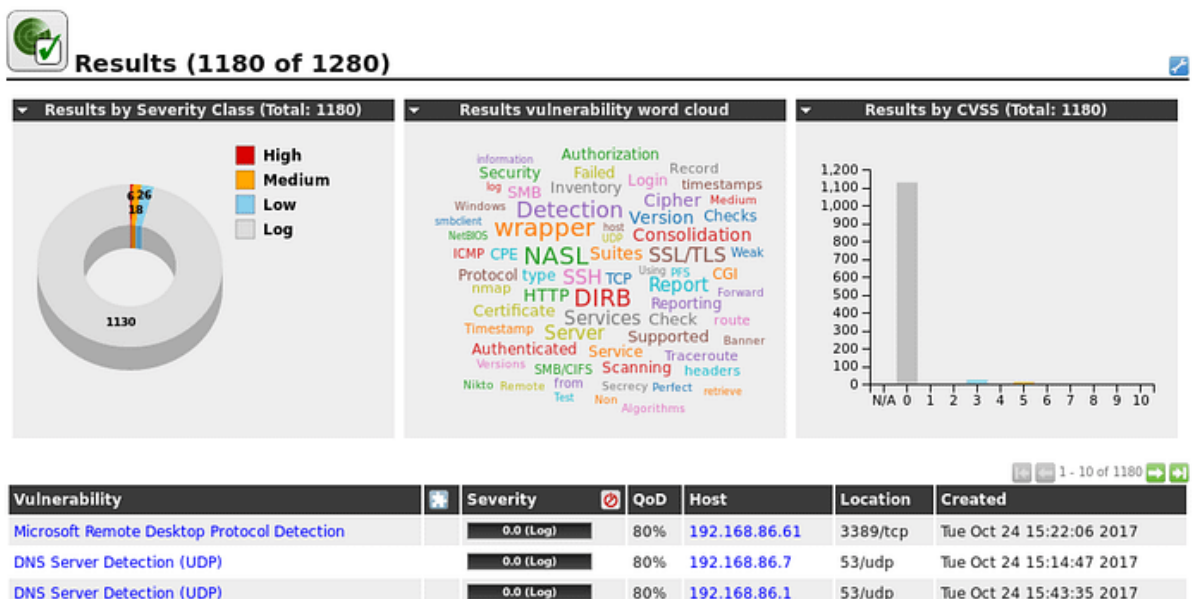
Order for target hosts: Random

Maximum concurrently executed NVTs per host: 3

Maximum concurrently scanned hosts: 15

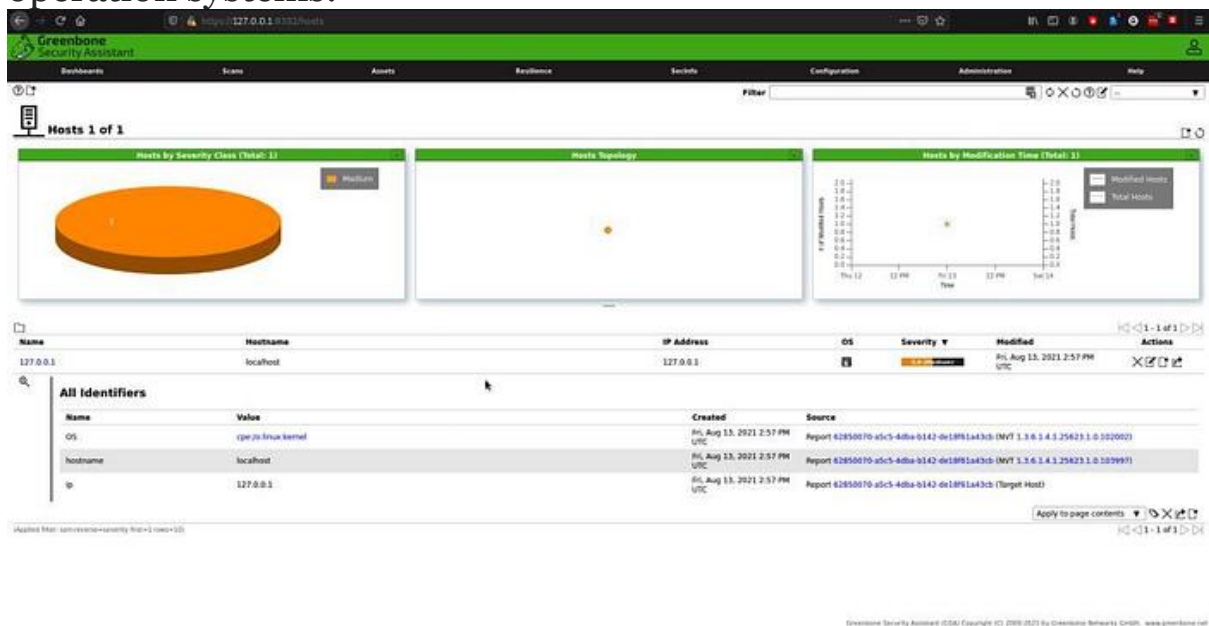
Create

With our more finely-tuned scan settings and target selection, the results of our scan are much more useful.



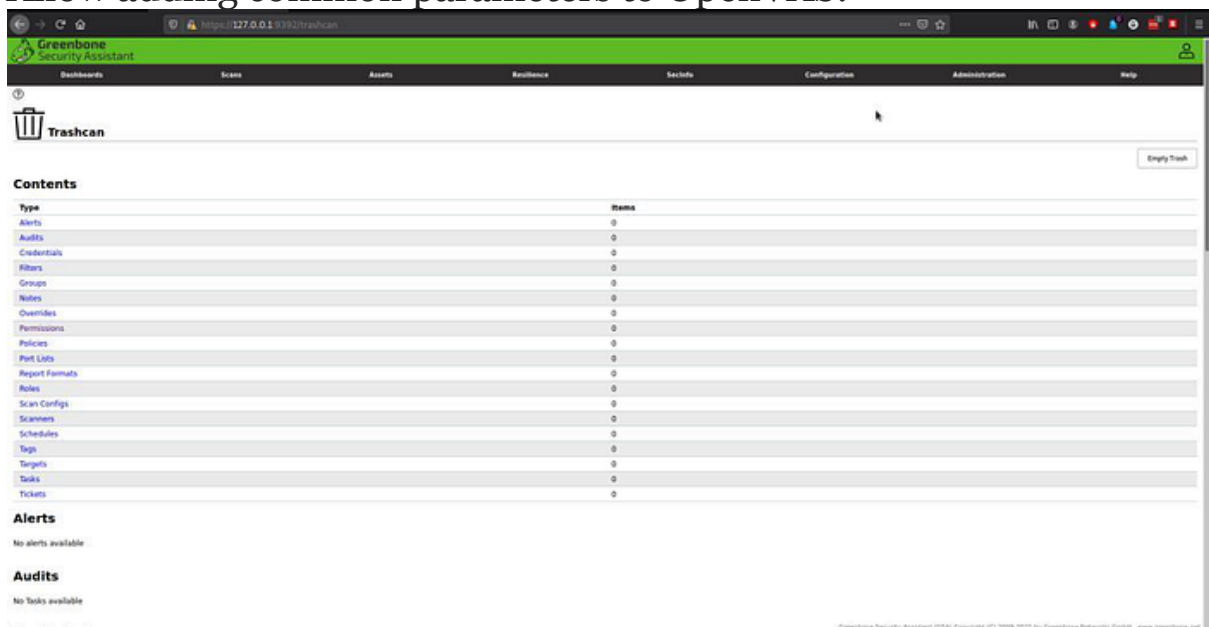
Assets

It permits visualizing the vulnerability of the parts akin to hosts or in operation systems:



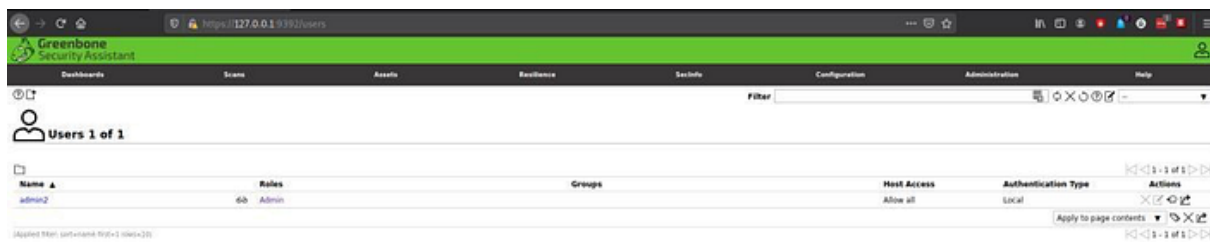
Additional features

Allow adding common parameters to OpenVAS:



Administration

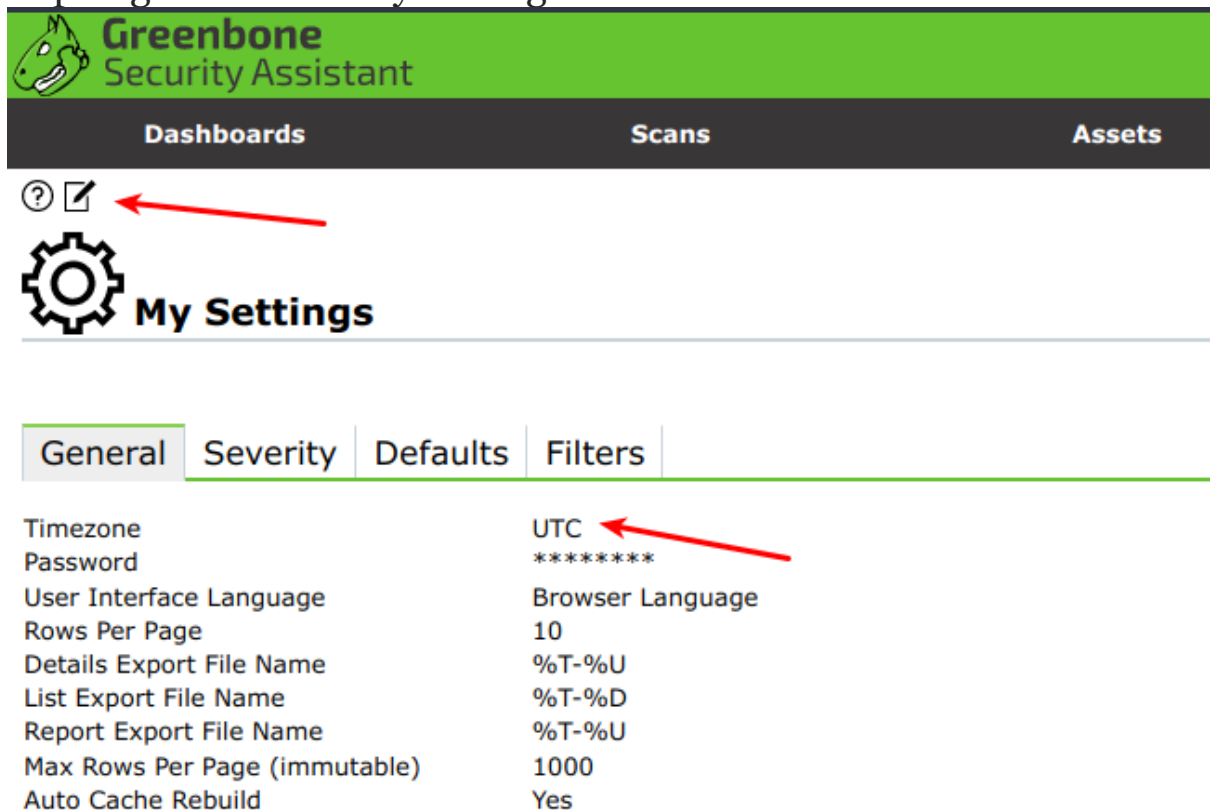
As the name suggests, you can manage passwords, users, etc.:



Change timezone

Note: Recommend setting the timezone as UTC, the report displays UTC time only no matter what timezone you set

Top-Right corner > My Settings



With the wide range of options available in OpenVAS, we were only really able to just scratch the surface in this post but if you take your time and effectively tune your vulnerability scans, you will find that

the bad reputation of OpenVAS and other vulnerability scanners is undeserved. The number of connected devices in our homes and workplaces is increasing all the time and managing them becomes more of a challenge. Making effective use of a vulnerability scanner can make that management at least a little bit easier.