Faraday Client

With Faraday you have five different ways to interact with the server:

- GTK GUI
- ZSH UI
- Web UI
- CLI
- ZSH Web (only available in our commercial versions)

GTK GUI

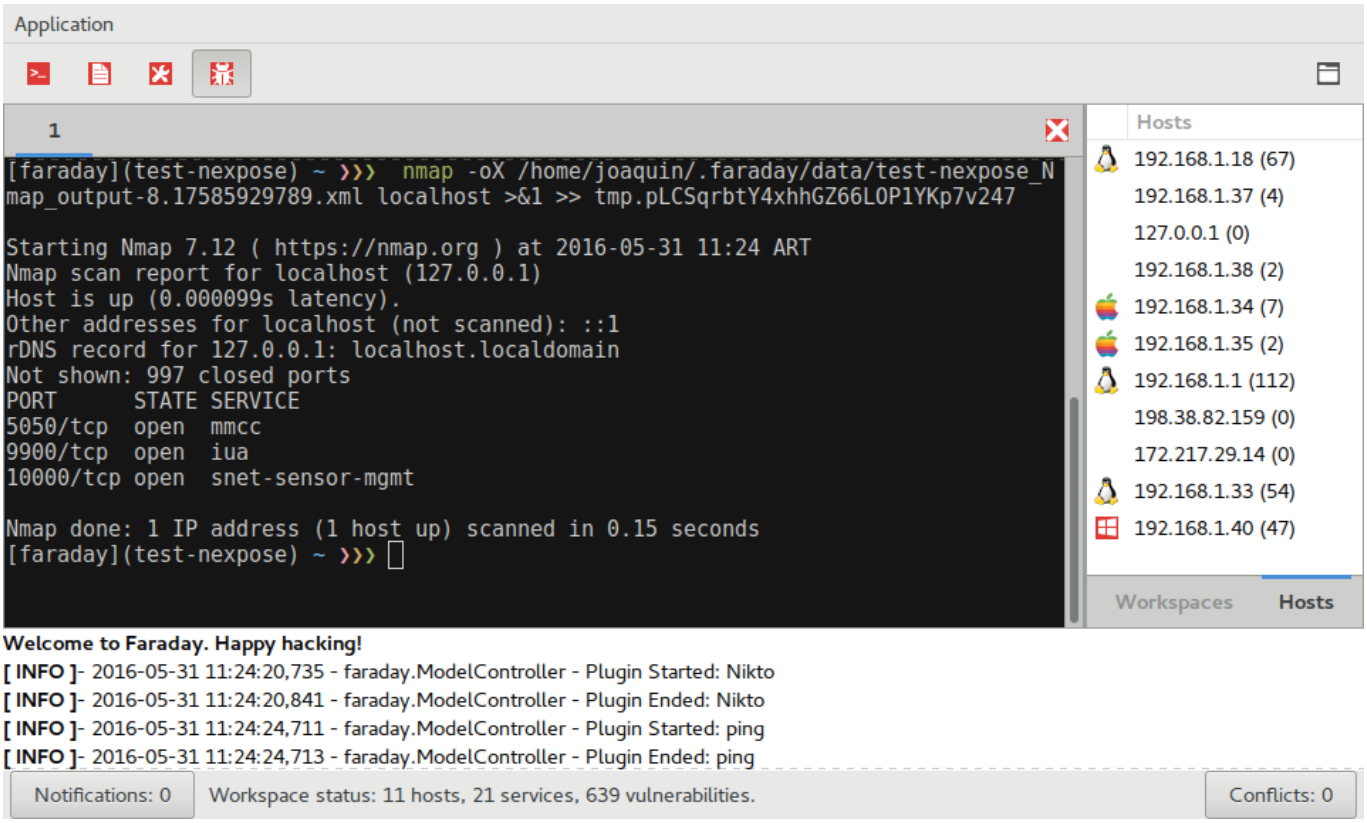
GTK+3 is designed to improve on the deprecated QT interface, so nothing should look out of place if you were already using Faraday. If you weren't, don't worry, it's pretty simple.

To try it, check out our installation manual.

To access Faraday GTK, run the following command in Faraday's folder:

```
$ python faraday.py
```

The main window



You will be presented with a special version of your own ZSH terminal. Just as with GTK, Faraday intercepts every command you execute and checks if there's a plugin available. If there is, Faraday will interpret all the

relevant information like ip adresses, hostnames, services, vulnerabilities, websites, and notes that the command generates.

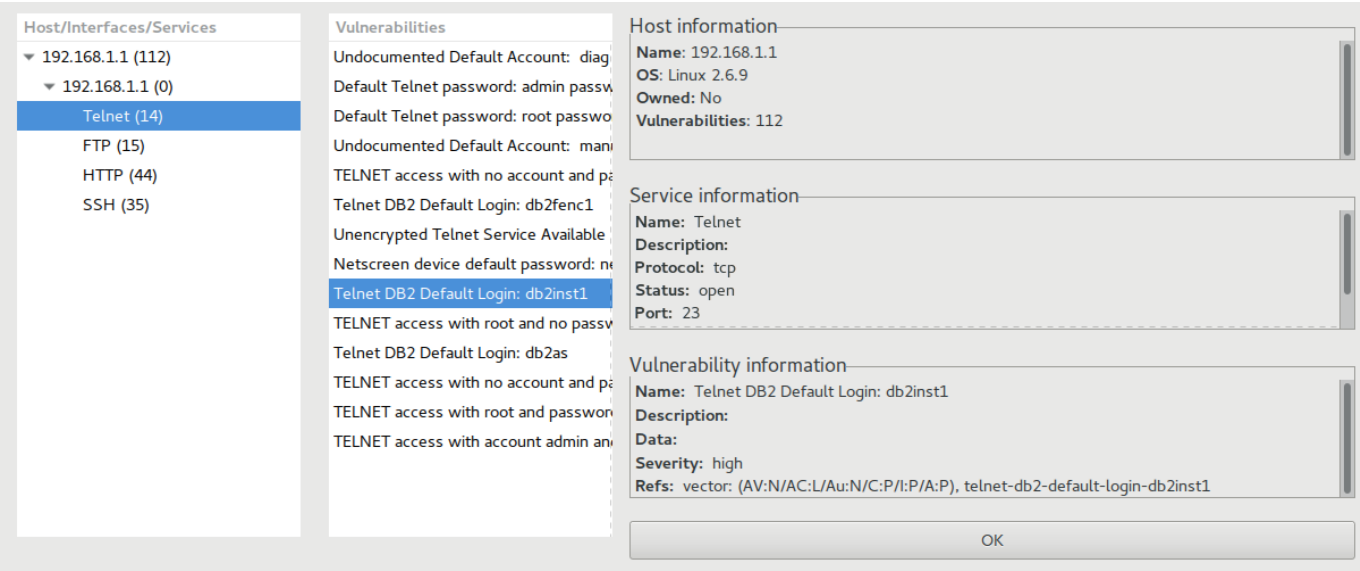
The menu-bar gives you access to the most common options: you can open a new tab to create a new workspace, toggle the log, or set your CouchDB URL in the preferences dialog (and login to the database if you're using our Pro or Corporate versions). At the rightmost border, in the folder icon button you'll be able to import any report by our supported plugins to Faraday.

The sidebar has two tabs, one for workspaces and the other for hosts. The workspaces tab allows you to change workspaces, while the hosts tab shows you all the hosts in your current workspace, plus the amount of vulnerabilities found in each one of them inside parenthesis. Clicking on a host will show you more detailed information, see Host information dialog

The status-bar has information about your workspace and also buttons to access the Notifications dialog and the Conflicts resolution dialog.

The log console works just as you'd expect, showing you what Faraday's doing in the background at all times. For more verbose output, you can run Faraday with the `--debug` flag.

Host information dialog

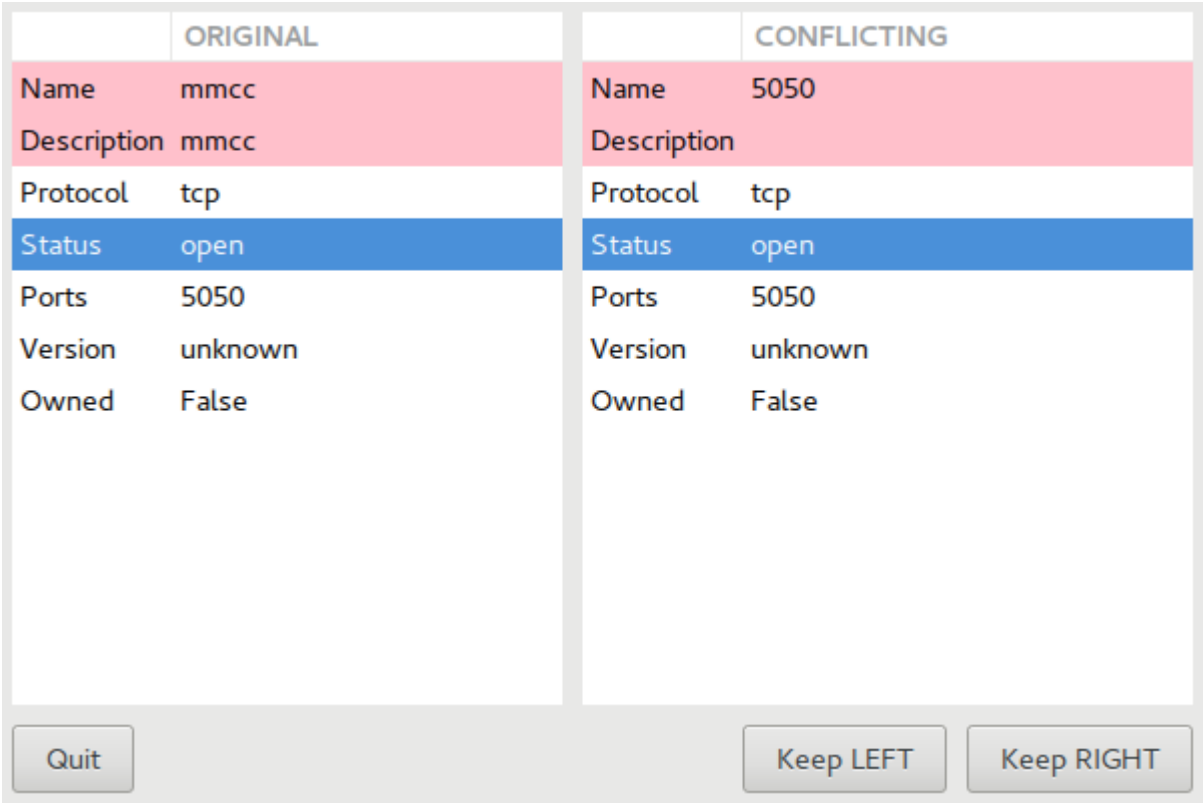


When you click on a host in the Host tab of the sidebar, you'll be presented with a window like the one above. The leftmost tree represents the Host itself, with all its interfaces as children. The interfaces, too, have children, which are the services of each interface. All of these items have the number of vulnerabilities discovered, inside parentheses.

The list of vulnerabilities shows the name of all the vulns found in the selected item of the leftmost tree.

The rightmost side of the windows shows detailed information of the host, the selected item of the leftmost tree (be it a service or an interface) and the selected vulnerability.

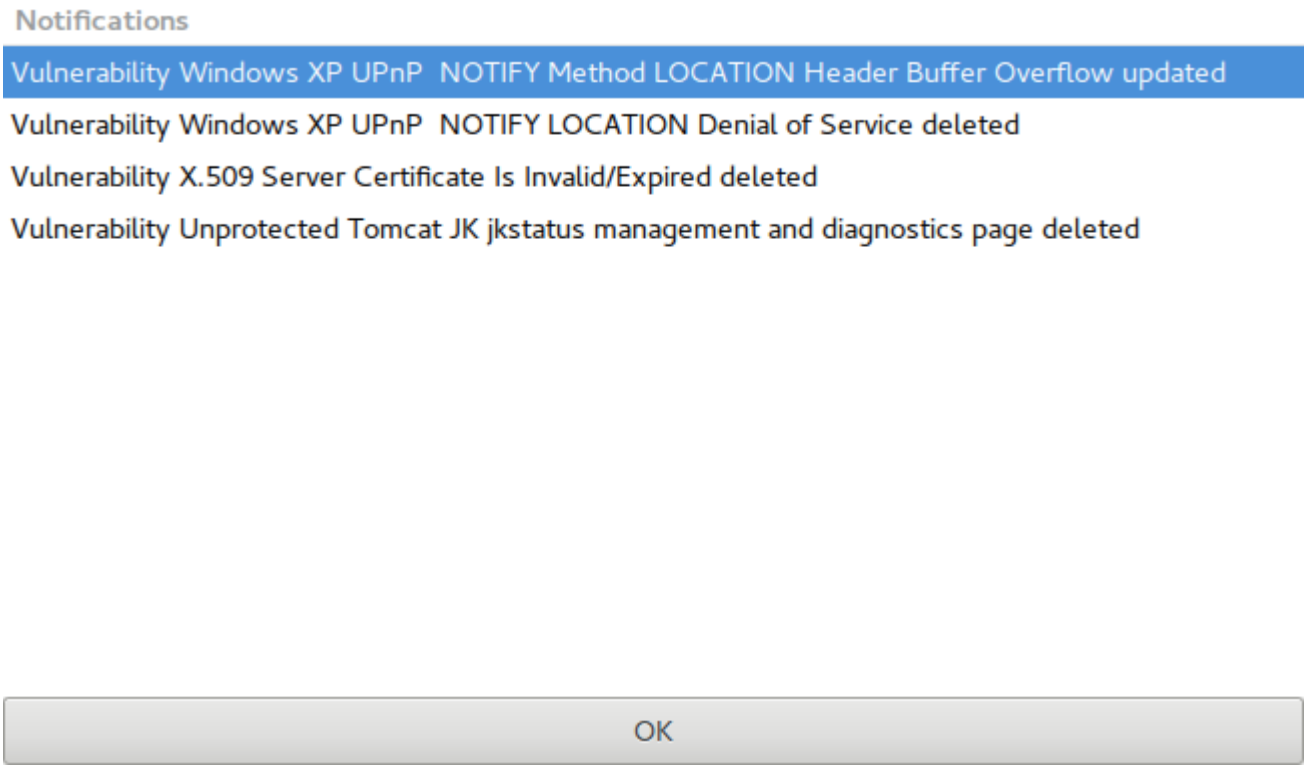
Conflicts resolution dialog



When Faraday finds an object which clashes with one you have already saved, it will inform you there's a conflict. Imagine you have a host marked as a Windows machine, but a tool detects a Linux installation. It's a conflict!

Faraday will show you the two conflicting objects, with its differences highlighted in red. You can edit the information in the objects, and then decide if you want to keep the left or right one.

Notifications dialog



Faraday is a **multi-user** integrated penetration test environment. That's why keeping up with changes coming from your collaborators is so important, and its why the notifications dialog exists.

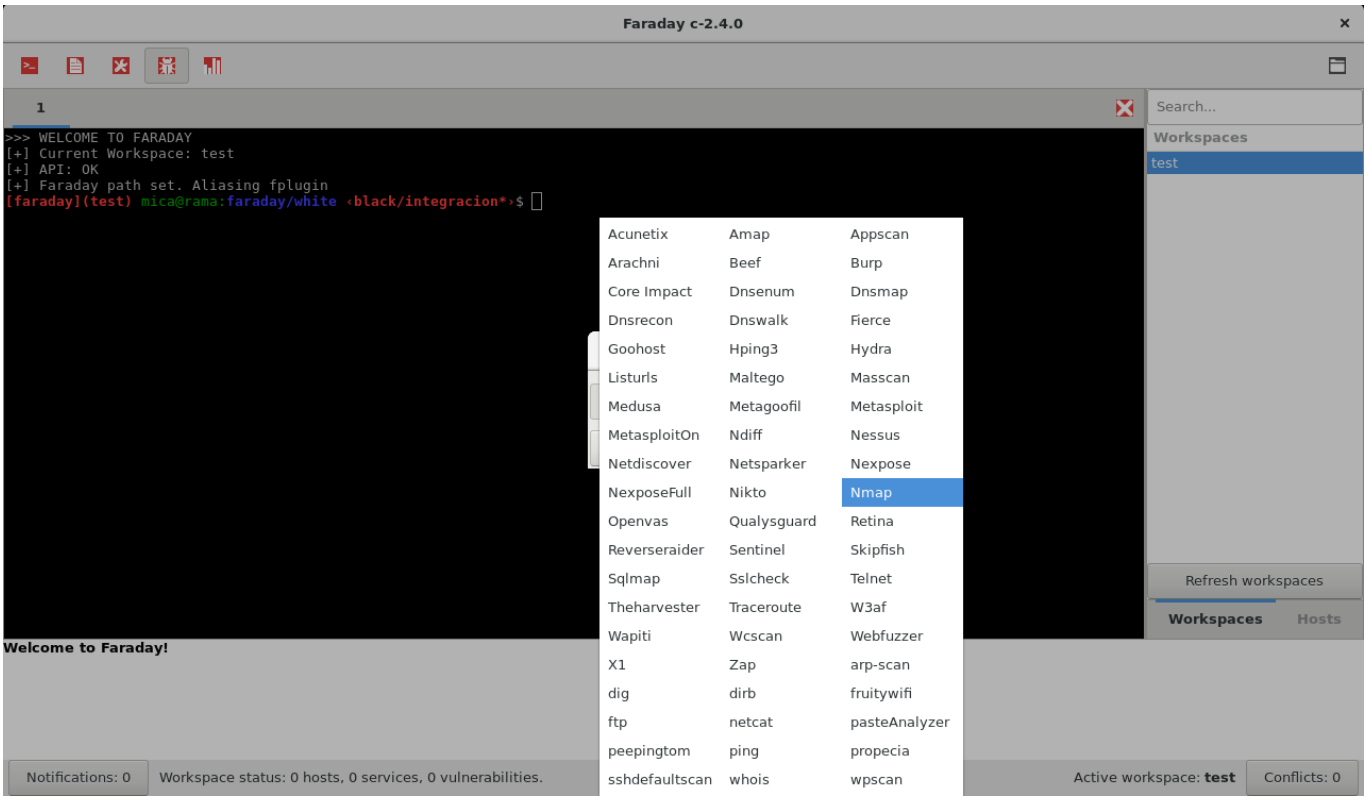
While working, the notifications counter will increase as new changes come from other instances of Faraday clients connected to the same database. If you click on the button, you'll be presented with a list of all the updates, so you are never kept in the dark of what your collaborators are up to.

Adding Reports

If you wish to add a report from a previous scan, you can also do it from the GTK Client.

To do so, click on the Report Button 

A dialog will open, from which you can select the tool that was used to generate the Report:



All the data in the report will be processed and added to the active Workspace, and the console will show a message when the plugin starts and ends.

ZSH UI

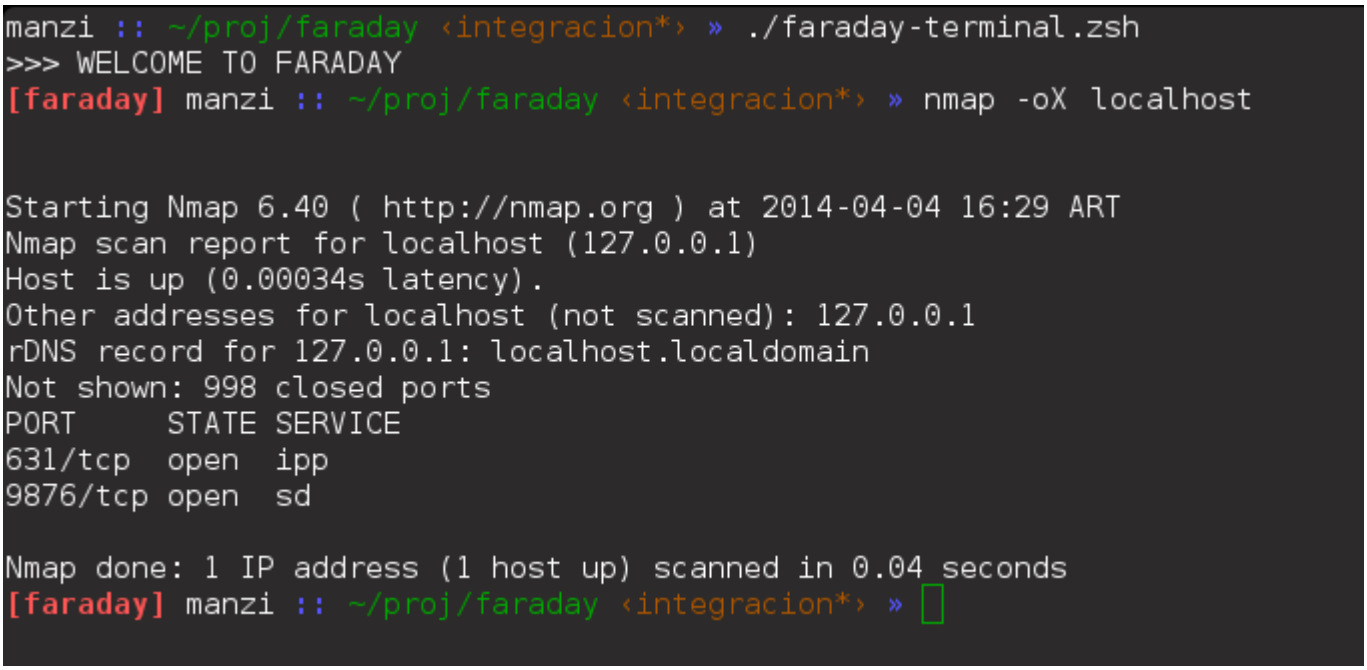
You can even run Faraday in detached mode connecting with a ZSH terminal to it:

First, you need to run Faraday with no GUI:

```
$ python faraday.py --gui=no-gui
```

Now, run Faraday Terminal:

```
$ ./faraday-terminal.zsh
```



To import your reports, drag-and-drop them into:

```
$ ~/.faraday/report/[workspace_name]
```

Replace [workspace_name] with the workspace's name you're working in.

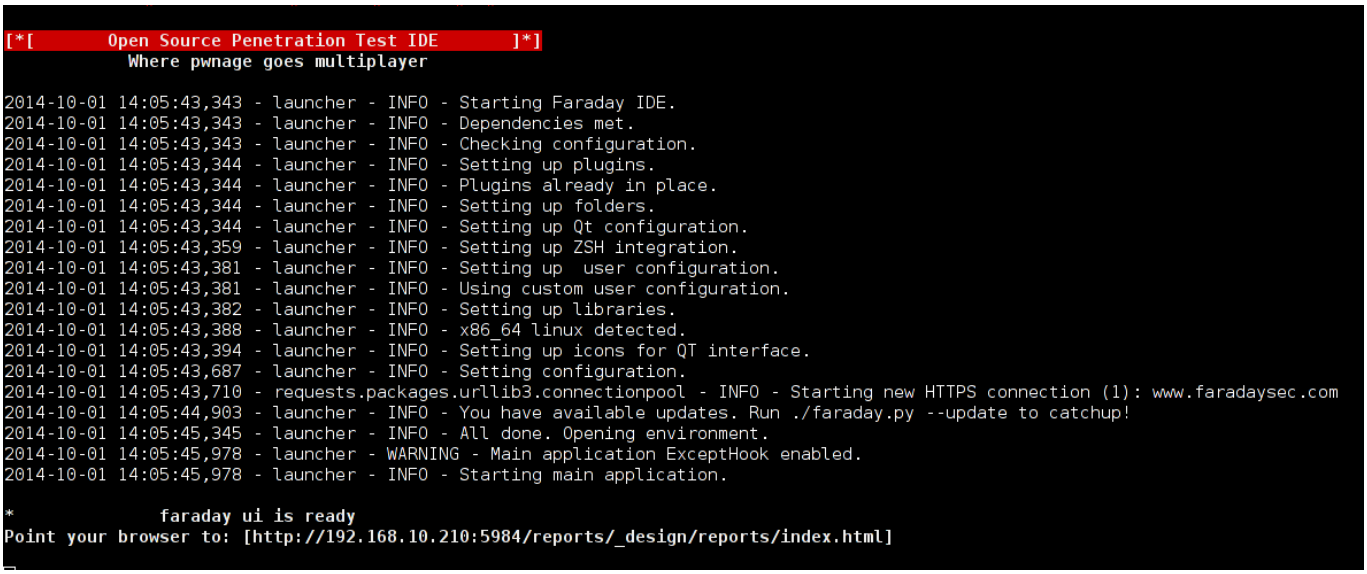
Faraday will parse your reports and upload the information extracted from them. All the available information are viewable through interfaces:

- GTK GUI
- Web UI

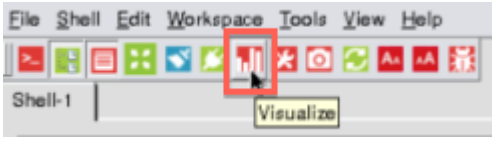
Web UI

In order to access the Web UI point your browser to: `http://[FARADY-SERVER]:PORT/_ui/`

The current URL address is displayed on console log information:



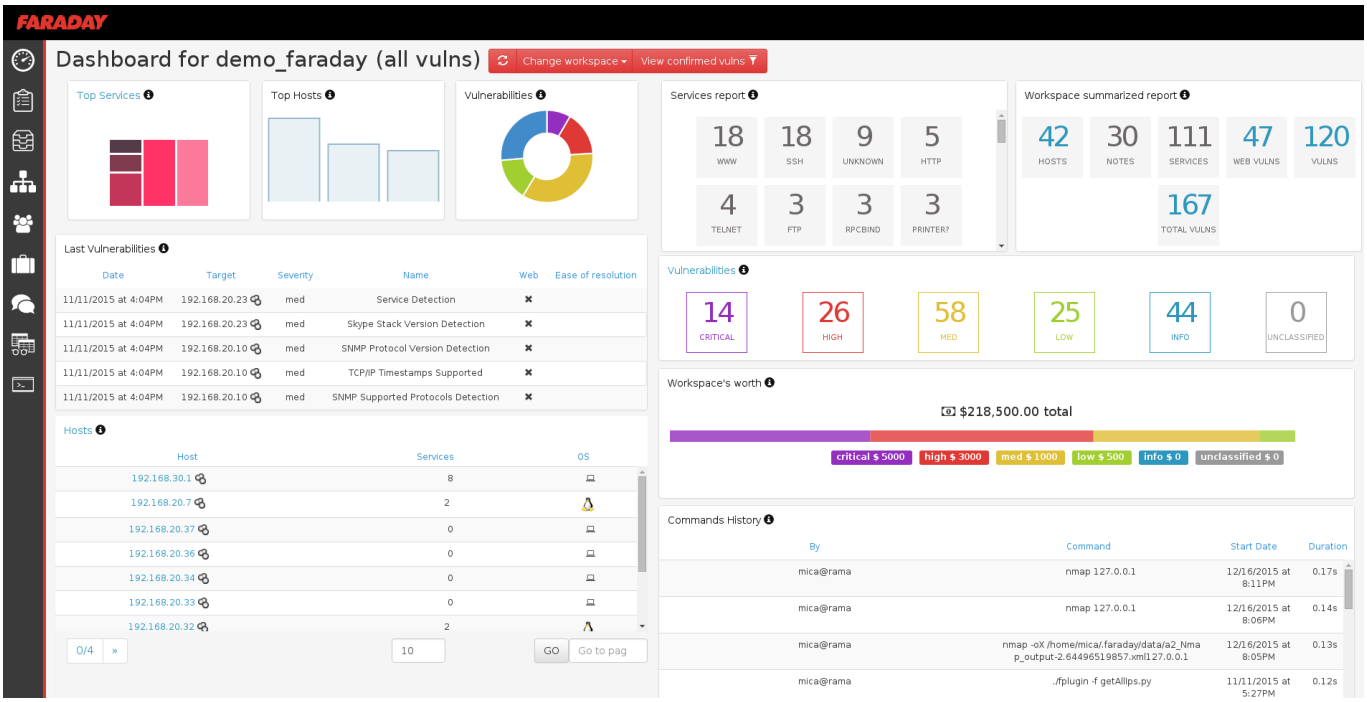
You can also go to the Web UI directly from Faraday GTK by clicking on this icon:



Faraday Community Version

Dashboard

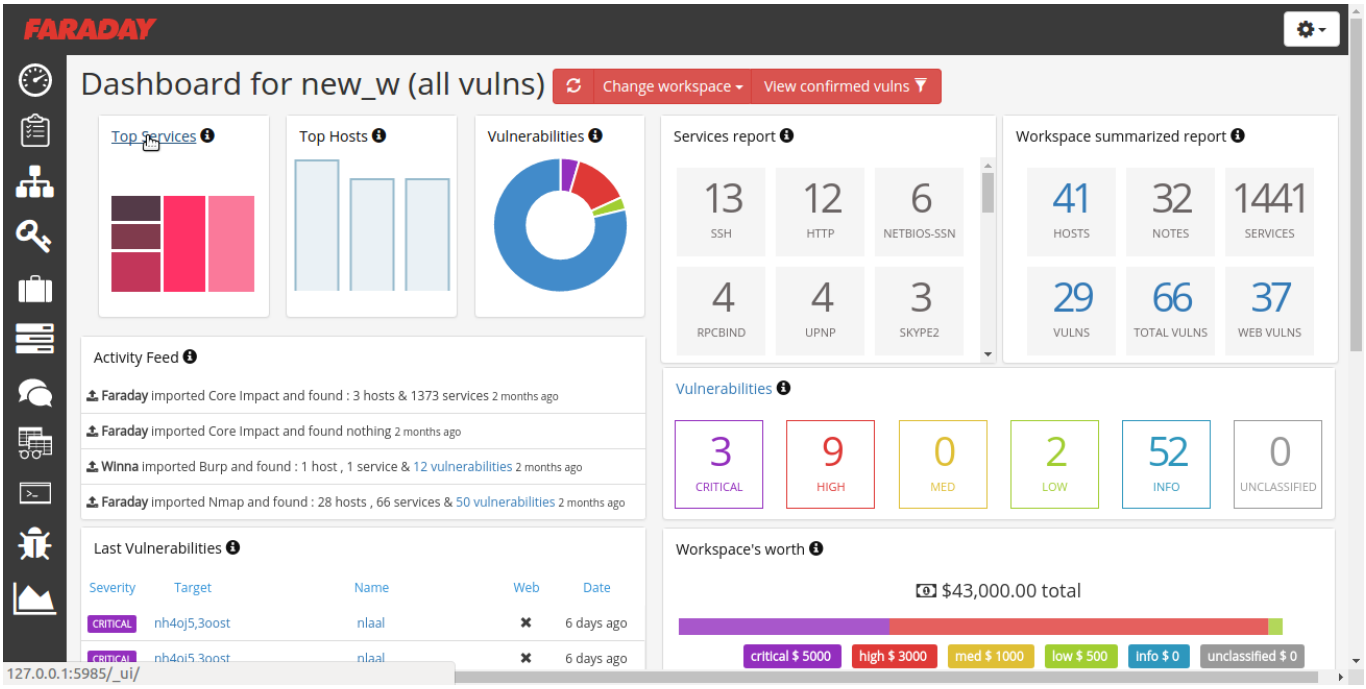
Faraday's dashboard contains a summary of all the data in a Workspace condensed into different boxes. Each box is a visualization a specific aspect of the collected data.

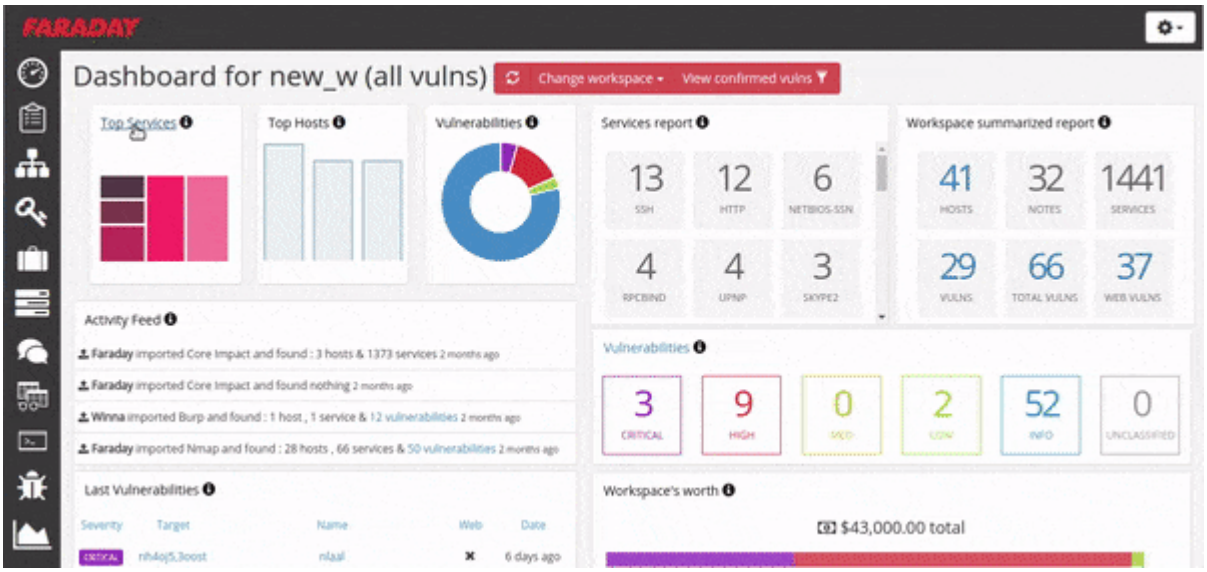


Let's see in more detail some of these boxes:

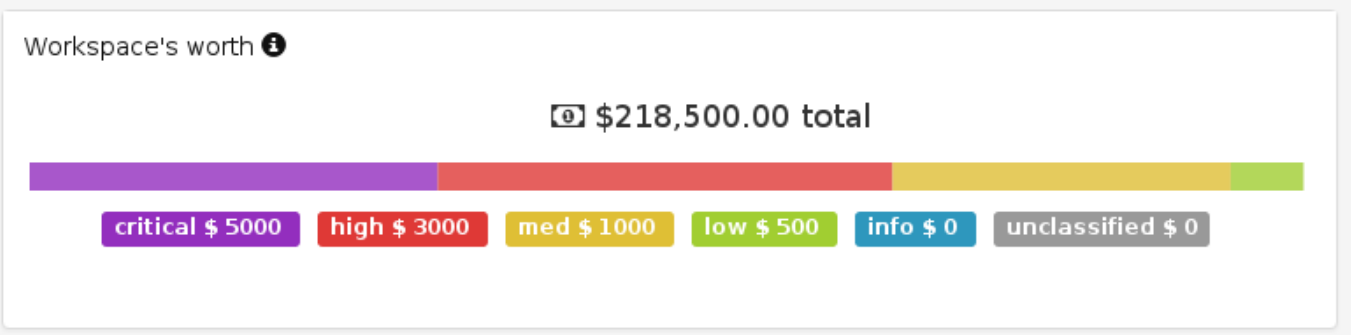
Top Services View

To access a treemap featuring the top services in the Workspace, click on the box title.





Workspace's Worth



This visualization is specially helpful when contributing in a bug bounty program. You can estimate how much your Workspace is worth according to the severity of your vulnerabilities.

You can edit the price of each severity level by clicking on it. The graphic will change as you type.

For instance, let's say that you have a workspace with 6 vulns, one of each severity level. And the price schema is:

- Critical vulns are worth 6 dollars
- High vulns are worth 5 dollars
- Med vulns are worth 4 dollars
- Low vulns are worth 3 dollars
- Info vulns are worth 2 dollars
- Unclassified vulns are worth 1 dollar

Then your Workspace will be worth

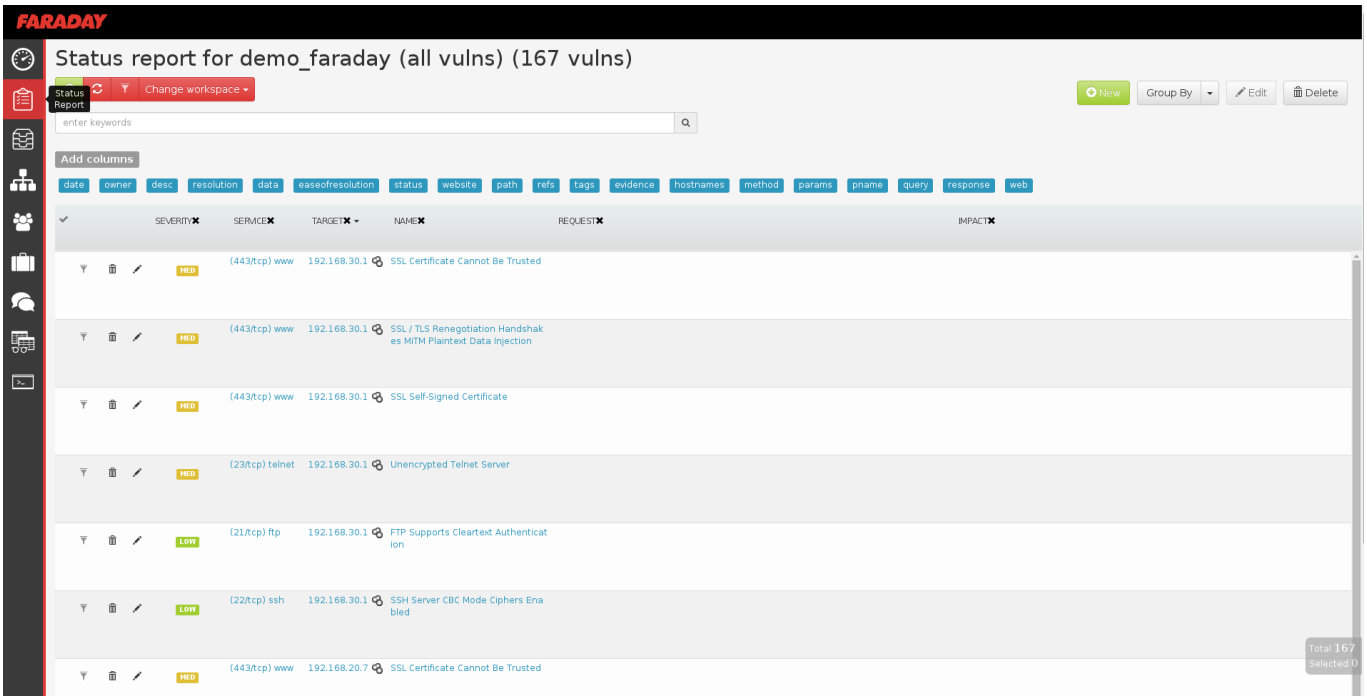
```
6*1+5*1+4*1+3*1+2*1+1*1 = $21
```

The length of the colored bars shows the proportion that severity represents in the final worth of your workspace.

Learn more about using Faraday for [[Bug bounties]].

Vulnerability Status Report

To view a full list of findings you can access the Status Report.



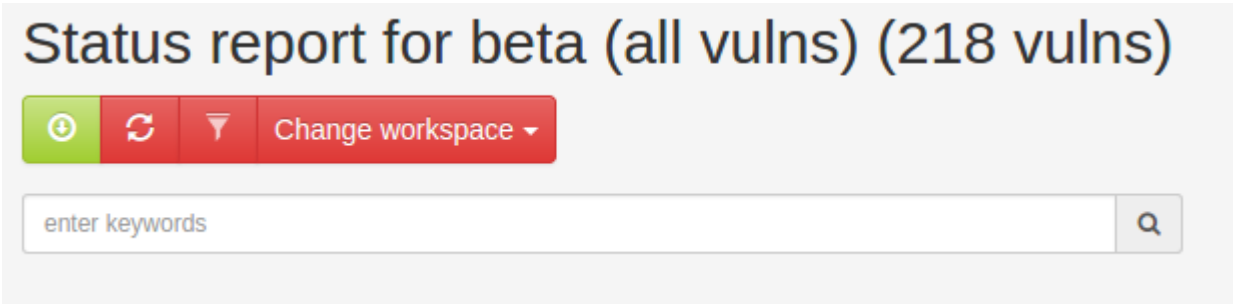
The Status Report provides several options including vulnerability search, filtering and management.

Personalize this view by clicking on the blue buttons to select the columns you wish to see, and remove the ones you don't need with the X's in the table. These changes will be persisted in your browser from session to session, so you only have to apply them once.

Search & Filter

To search, type the keyword in the text-box above the table.

You can find the text filter in both the Status Report and Hosts views. Keep in mind that field values are **case-insensitive**.

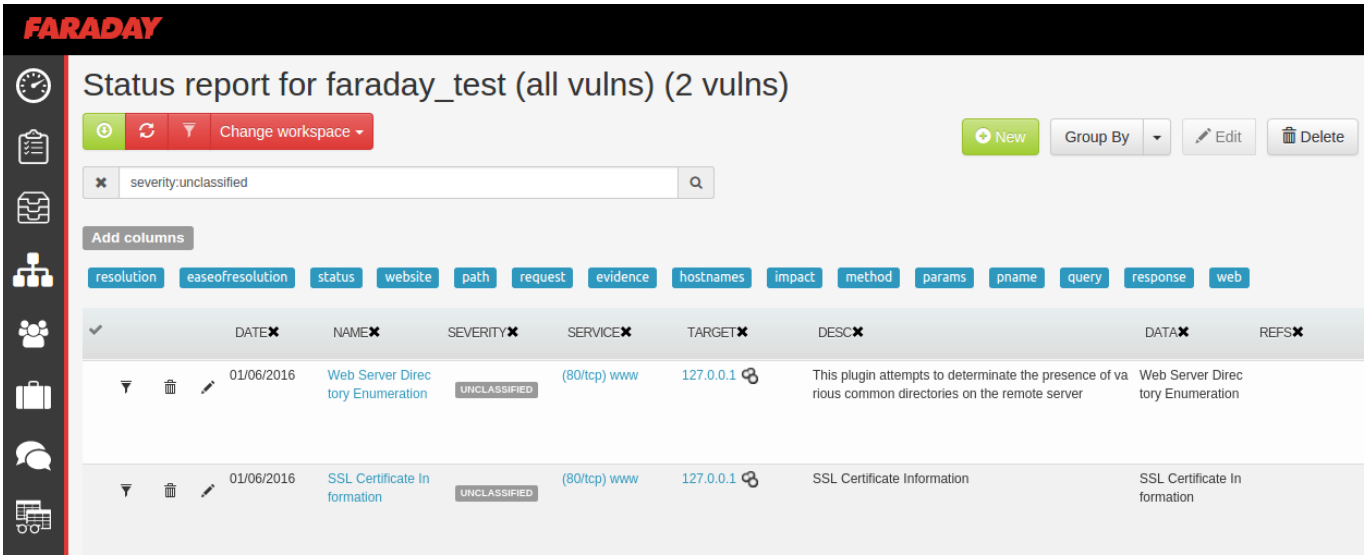


Filter by field

To search by field enter the name of field (e.g. **severity**), continue with a colon (**:**) and finally put in the word that you want to find.

Examples:

- severity:unclassified
- name:Nessus scan info

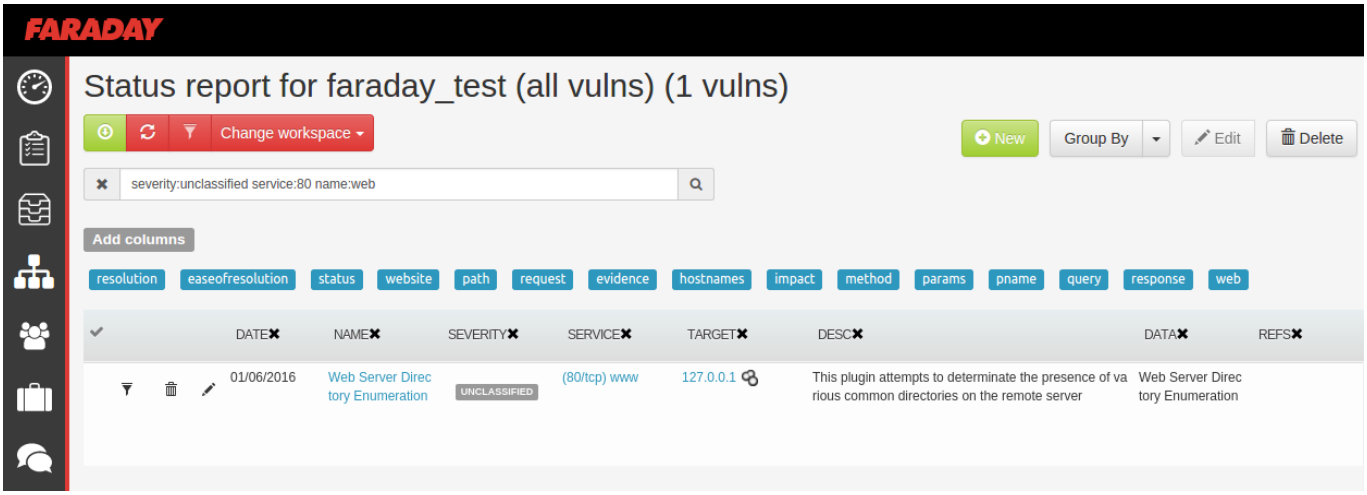


Filter by many fields

To search by many fields, use a *SPACE BAR* to separate each field.

Examples:

- `severity:unclassified target:173.252.100.18`
- `severity:low service:443 target:173.252`

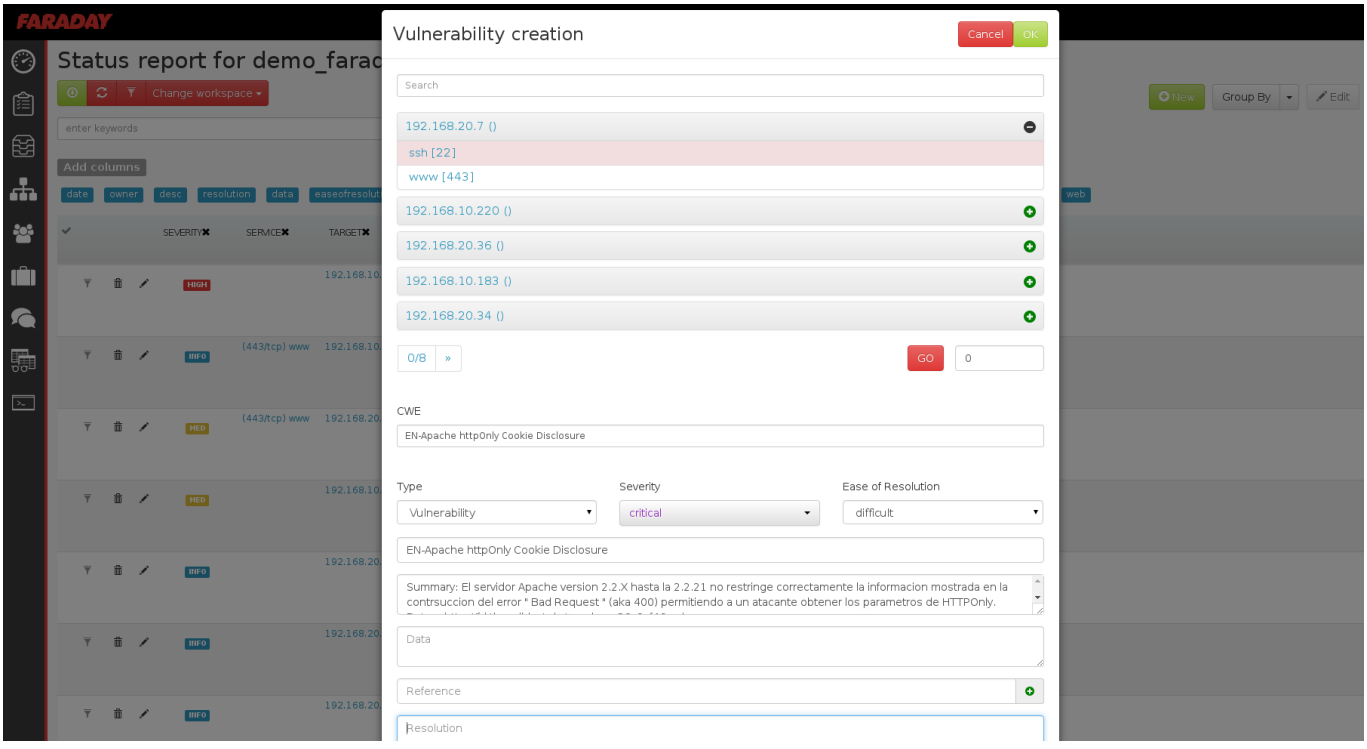


Grouping

To group vulnerabilities by field you can use the **Group By** button. After the vulns are grouped you can select them for easy batch editing.

Vulnerability Creation

To create vulnerabilities manually, you can go to the status report page and click the "New" button at the top right corner. You should see a dialog similar to this:



Make sure you select a host (and a service if the vulnerability applies to it), and the correct type. If you create a web vulnerability, you will have a couple more fields available, such as path, method, request, response and so on.

Faraday Professional & Corporate - Commercial versions

This version includes advanced visualizations, tags, pentest comparison, pentester ranking among others.

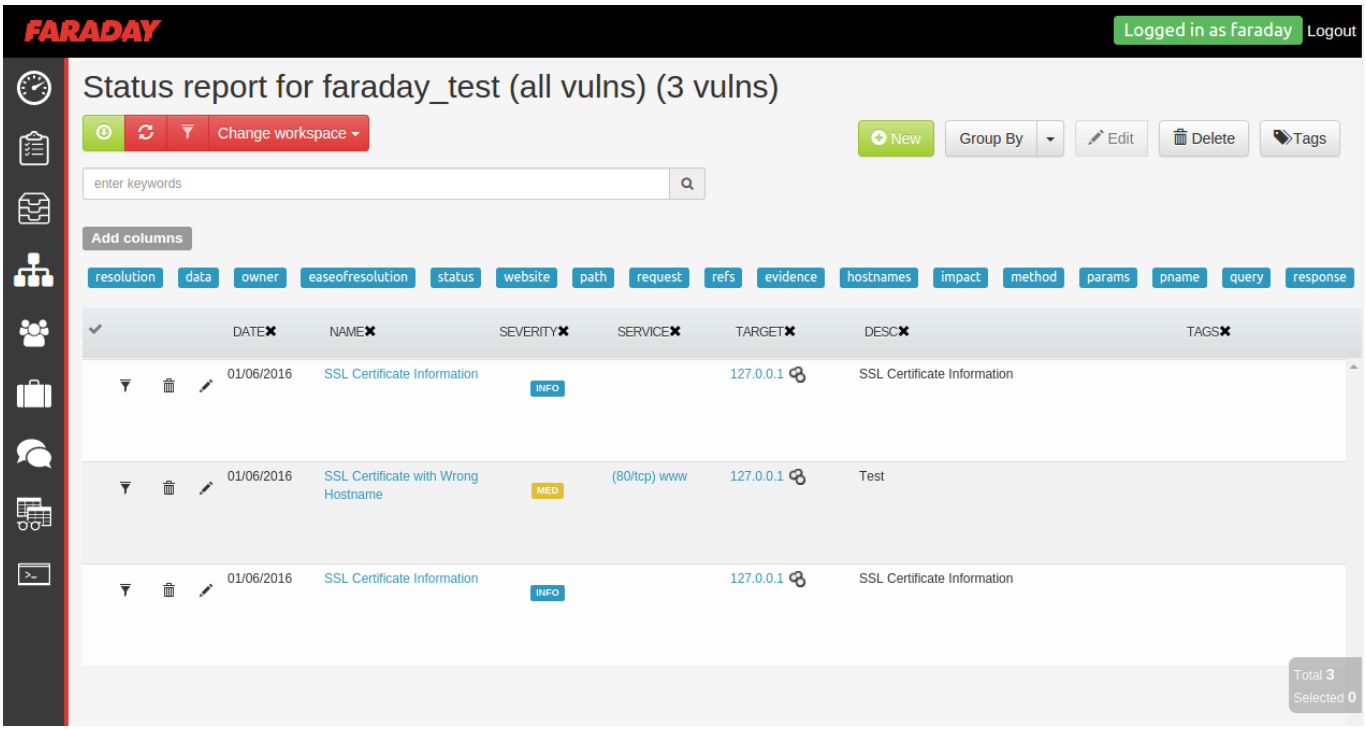
Tags

Tags allow you to organize your vulnerabilities by letting you create and edit categories: "environment", "technology", "state", "language", "projects", whatever! Your team can also see the tags you make.

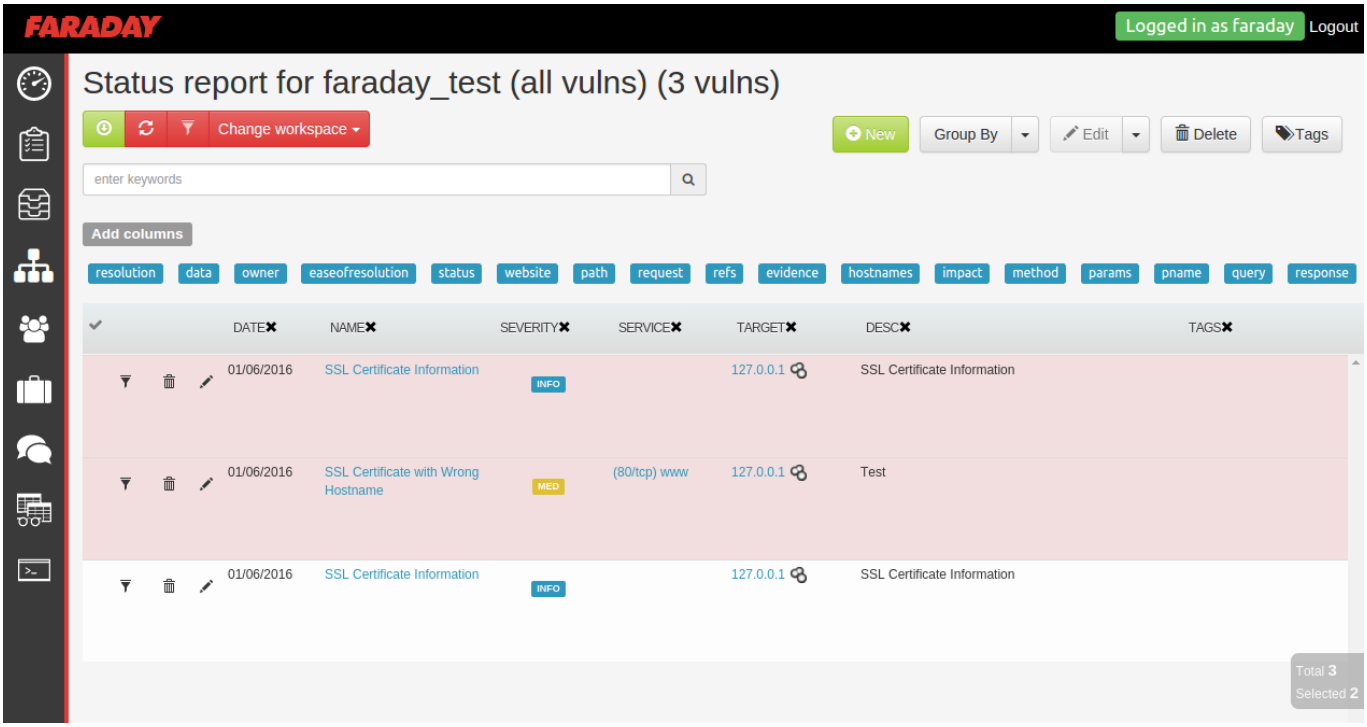
Tags only apply to an individual workspace, so you can use different tags for different projects.

How to tag vulnerabilities

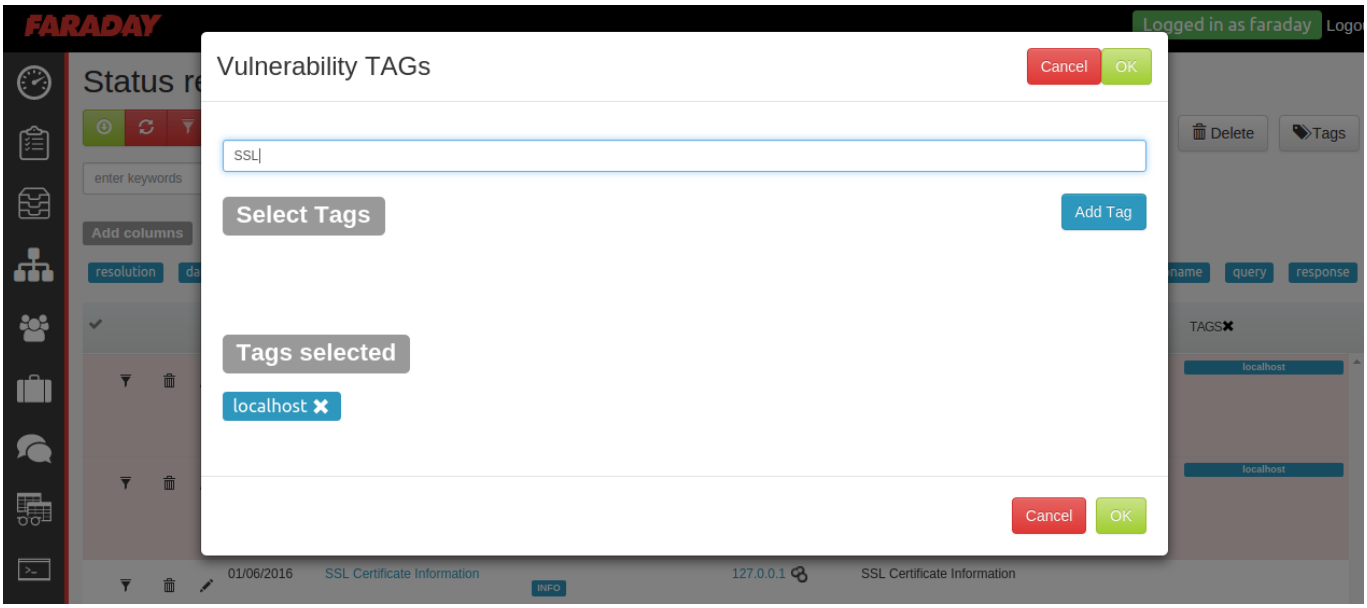
Using the specified username/password from the Faraday server configuration, from the Navigator start the session in Faraday's GUI. Once you have authenticated, click on the “Status Report” icon.



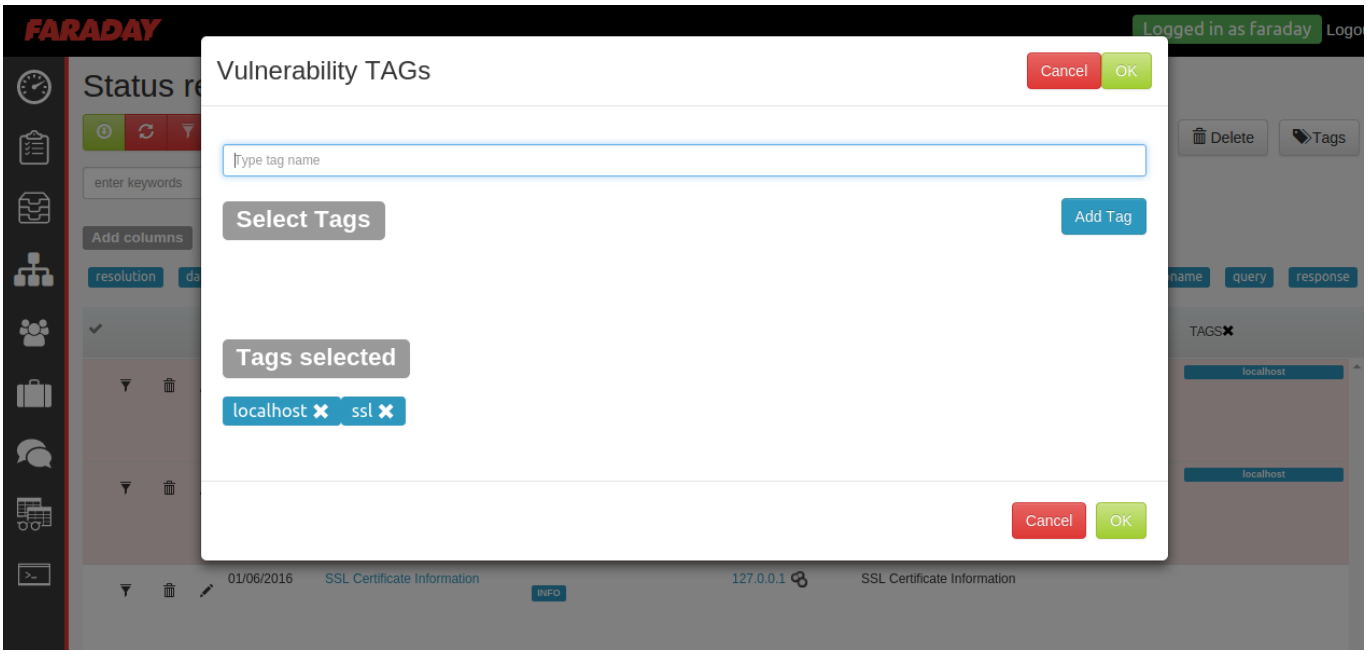
Select the vulnerabilities that you want to tag (for example those that have to do with SSL protocol)



After picking one of the vulnerabilities click on the "Tags" button

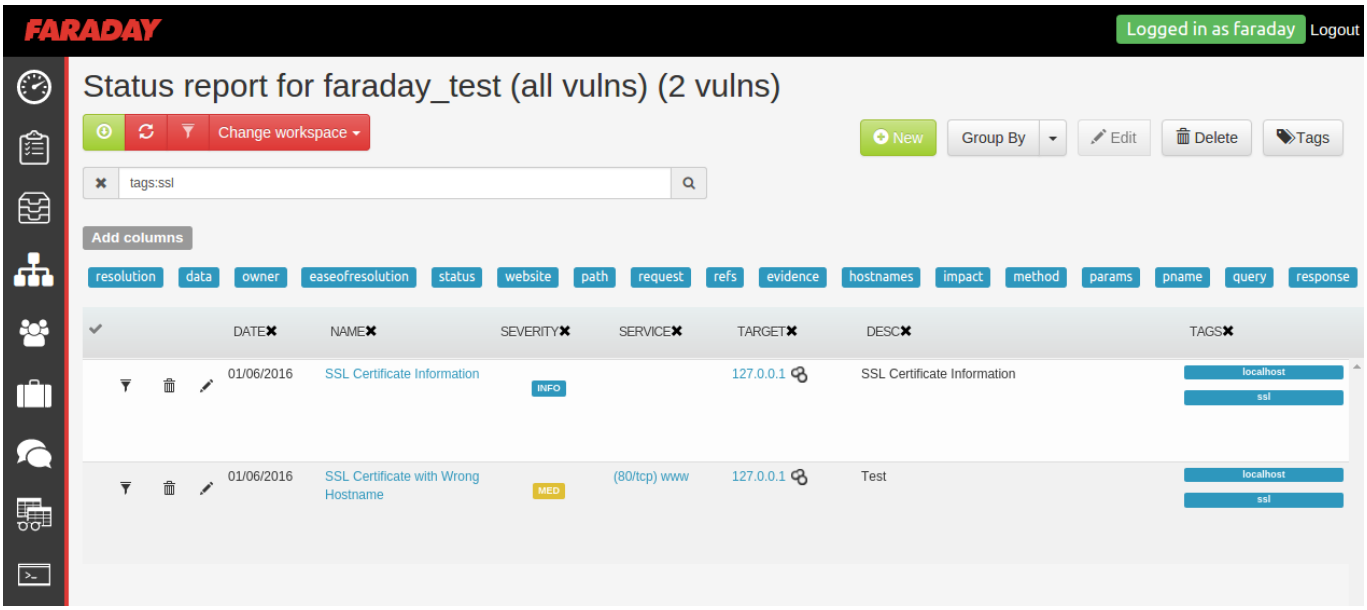


Create the tag that you want (in our case SSL) and click OK



Search tagged vulnerabilities

From the Status Report you will be able to search using the different tags. You can add ! in front of the search criteria in order to invert the result. For example *tag:!example_tag* will result in all vulnerabilities that DON'T have the tag example_tag.



CLI

It's possible to use Faraday in Command-Line Interface (CLI) mode, allowing you to process your reports in batch. So lets say you want to process the XML output of an **nmap** scan located in `/tmp/nmap_scan.xml` and send the results to a workspace called **project_one**. The way to do it using CLI mode would be to run:

```
$ ./faraday.py --cli --workspace project_one --report /tmp/nmap_scan.xml
```

NOTE: the workspace has to already exist for the command to work.

Professional and corporate versions

If you're using a professional or corporate version, you'll probably need to run Faraday as a certain user, with permissions to access your workspaces. You can pass your credentials using a simple json file the contains both your username and password. You have a template in the directory of your Faraday installation called `credentials.json`, but you are allowed to use any path and filename for this json file. The structure is this:

```
{
    "username": "your_user_here",
    "password": "your_password_here"
}
```

And then run Faraday:

```
$ ./faraday.py --cli --workspace project_one --report /tmp/nmap_scan.xml --
creds-file /path/to/file/creds.json
```

ZSH web console

- Commercial version -

It's also possible to use the ZSH interface inside your web browser. The idea of the web-shell is to allow you to work directly from the web using ZSH as a console. You would be connected to your own shell (listening in loopback interface).

To do this, fist you need to install butterfly:

```
$ pip install --user butterfly
```

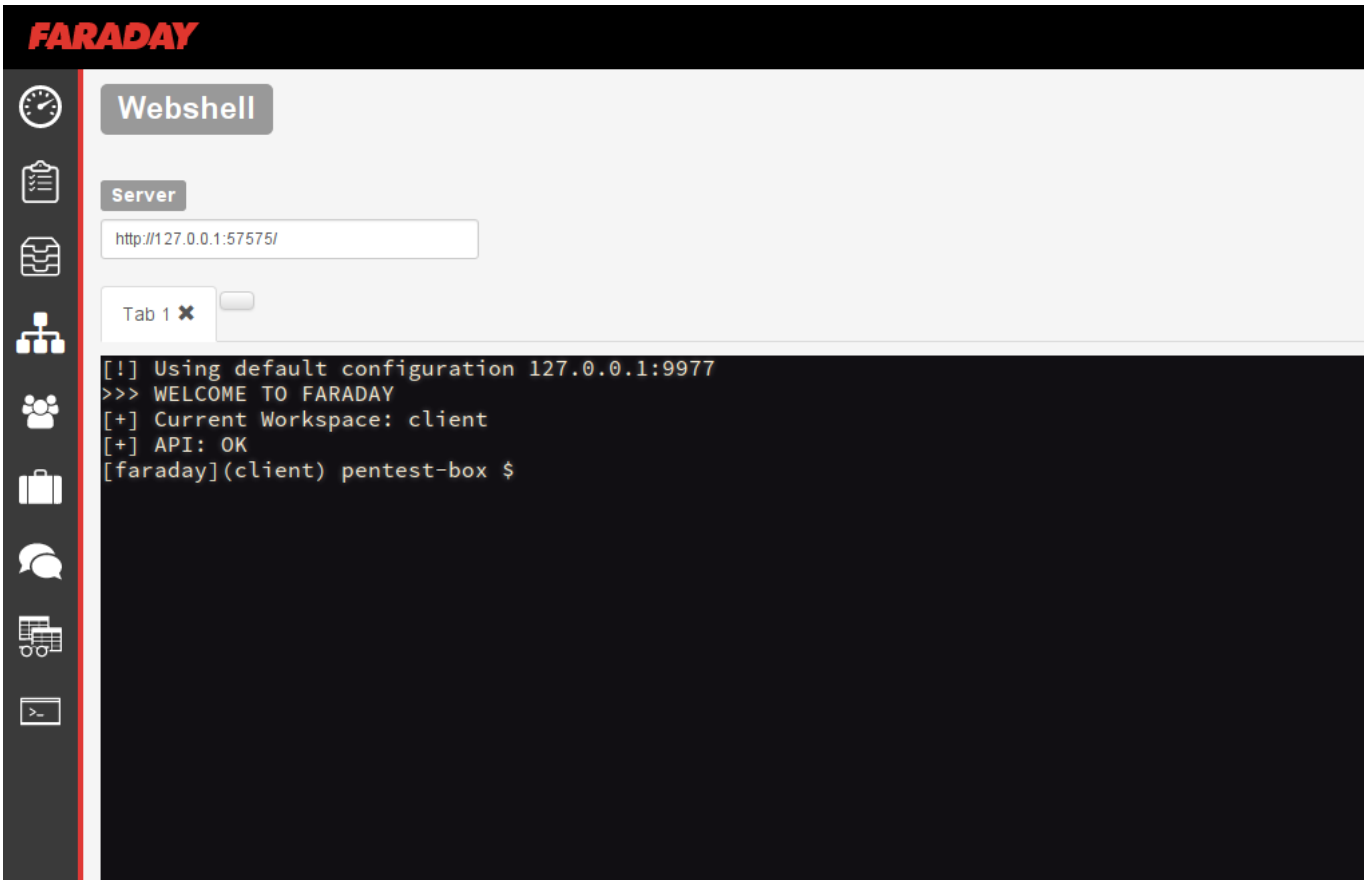
NOTE: If you have both python2 and python3 in your system, it's better to use pip2 instead of pip.

Now, run butterfly:

```
$ butterfly.server.py --unsecure --shell=/bin/zsh --
cmd="path/to/faraday/faraday-terminal.zsh [host] [port]"
```

Of course, you need to set the path of the folder in which you have Faraday (the faraday-terminal.zsh script should be in the root of that folder) and pass the host and port arguments to that script, in case you've changed the Faraday's REST API parameters (remember that you have to run Faraday GTK or Faraday `--gui=no-gui` so that the terminal for ZSH functions properly).

Now, open a new tab from the web-shell icon in the web UI's sidebar, and you should see the zsh shell up and running!


FARADAY
Webshell
Server
http://127.0.0.1:57575/
Tab 1
[!] Using default configuration 127.0.0.1:9977
>>> WELCOME TO FARADAY
[+] Current Workspace: client
[+] API: OK
[faraday](client) pentest-box $