Developing a firmware scraper to download firmware files from various vendors

A python-based solution that can download firmware images from different vendors (e.g., from the corresponding web pages or FTP servers). The meta data of the firmware files should be stored in a MySQL database. Additionally, an EMBArk module with access to all the data should be implemented. From the EMBArk (https://github.com/e-m-b-a/embark) area it should be possible to initiate firmware scans.

Solution details:

A Python Selenium and MySQL based solution will be developed which will have the following features/components:

1. One main script and per vendor modules. This is required because each vendor website will have its own implementation of the firmware download area and this will be changed regularly.

2. Python Selenium based scrapper will be written for each vendor, responsible for downloading the files and grabbing the meta data from the vendor's site. The tool will only download new firmware files that have been added by the vendor. The first execution of the script will initiate an initial download of all available firmware files. Subsequent runs will only download new firmware which was added. This will be achieved by analysing data (e.g., file name, checksum) available in the database and skipping the files that have already been downloaded and processed.

3. Mandatory database metadata fields include the following:
    - Manufacturer (e.g., Siemens, D-Link, AVM, ABB, …)
    - Model/Name (e.g., S7-1500, DIR300, Scalance-X, …)
    - Version (e.g., 1.2.3)
    - Type (e.g., Router, Switch, Firewall, PLC, …)
    - Release Date (if available)
    - Checksum (sha512)
    - EMBA tested (yes/no)
    - EMBA link to report (filesystem)
    - EMBArk link to report (http://...)
    - Firmware download link (vendor link)
    - Firmware filesystem link (to find it on the filesystem)

4. The firmware files itself will be stored in the file system and will be referenced by some index ID in the database (this is primarily needed to setup firmware scans with EMBArk or EMBA).

5. Each vendor will be analysed manually to identify the following areas. These are required to develop the script:

    - URL and method for the firmware download page (web request vs. FTP download)
    - Credential Requirements (Simple Signups, Specific Signups, No Signups)
    - Any Captcha on the page
    - Any honeypot traps

6. If there are credentials required to download the firmware and the credentials are simple ones where a simple sign up is possible, the signup will be done manually as part of the manual analysis using a webmail account dedicated for this work. (This webmail account will be handed over to the project owner at the end of the project)

7. The tool will try to imitate human like behaviour (to a limit) while scraping the web page, so that if the vendor site has scraper/crawler detection logic implemented, it can be skipped. If needed, a Tor proxy could also be used. This will be achieved by adding random delays, random view time, avoiding honeypot traps through manual analysis.

8. The json configuration file can be setup to represent the polling interval for each of the vendor scraper and when the execution script is run it will go and schedule each of the vendor scripts individually according to the polling interval defined in the config.

9. Finally, an EMBArk extension accessing and managing the metadata and the firmware files including analyzing them should be implemented. EMBArk should have access to the firmware meta data from the database. Additionally, it should be possible to initiate scans from the EMBArk web environment. At the end the results should be available in the current EMBArk dashboard and the detail reports should be available via the reporting dashboard. The setup of this module should be optional, and the setup should be handled via an automated script.

Public solutions:

The following solutions are known but mostly outdated. They can be used as a first reference:

- https://github.com/firmadyne/scraper
- https://github.com/TheRaphael0000/SwisscomFirmwares
- https://github.com/MikimotoH/Sitecom_Harvester
- https://github.com/MikimotoH/DLink_Harvester
- https://github.com/MikimotoH/Asus_Harvest
- https://github.com/MikimotoH/netgear_downloadcenter
- https://github.com/MikimotoH/Avm_FritzBox_Harvester
- https://github.com/MikimotoH/Belkin_Harvest
- https://github.com/JustAnotherCoderOnASaturday/Firmware-Scrapers

Company Contact: Michael Messner – michael.messner@siemens-energy.com