ja-netfilter 2022.2.0

A javaagent framework

Usage

- download from the releases page
- add `-javaagent:/absolute/path/to/ja-netfilter.jar` argument (**Change to your actual path**)
    - add as an argument of the `java` command. eg: `java -javaagent:/absolute/path/to/ja-netfilter.jar -jar executable_jar_file.jar`
    - some apps support the `JVM Options file`, you can add as a line of the `JVM Options file`.
    - **WARNING: DO NOT put some unnecessary whitespace characters!**
- or execute `java -jar /path/to/ja-netfilter.jar` to use `attach mode`.
- for **Java 17** you have to add at least these `JVM Options`:

```
--add-opens=java.base/jdk.internal.org.objectweb.asm=ALL-UNNAMED
--add-opens=java.base/jdk.internal.org.objectweb.asm.tree=ALL-UNNAMED
```

- edit your plugin config files: `${lower plugin name}.conf` file in the `config` dir where `ja-netfilter.jar` is located.
- the `config`, `logs` and `plugins` directories can be specified through **the javaagent args**.
    - eg: `-javaagent:/path/to/ja-netfilter.jar=appName`, your config, logs and plugins directories will be `config-appname`, `logs-appname` and `plugins-appname`.
    - if no javaagent args, they default to `config`, `logs` and `plugins`.
    - this mechanism will avoid extraneous and bloated `config`, `logs` and `plugins`.
- run your java application and enjoy~

Config file format

```
[ABC]
# for the specified section name

# for example
[URL]
EQUAL,https://someurl

[DNS]
EQUAL,somedomain

# EQUAL       Use `equals` to compare
# EQUAL_IC    Use `equals` to compare, ignore case
# KEYWORD     Use `contains` to compare
# KEYWORD_IC  Use `contains` to compare, ignore case
# PREFIX      Use `startsWith` to compare
# PREFIX_IC   Use `startsWith` to compare, ignore case
# SUFFIX      Use `endsWith` to compare
# SUFFIX_IC   Use `endsWith` to compare, ignore case
# REGEXP      Use regular expressions to match
```

Debug info

- the `ja-netfilter` will **NOT** output debugging information by default
- add environment variable `JANF_DEBUG=1` (log level) and start to enable it
- or add system property `-Djanf.debug=1` (log level) to enable it
- log level: `NONE=0` , `DEBUG=1` , `INFO=2` , `WARN=3` , `ERROR=4`

Debug output

- the `ja-netfilter` will output debugging information to the `console` by default
- add environment variable `JANF_OUTPUT=value` and start to change output medium
- or add system property `-Djanf.output=value` to change output medium
- output medium value: [`NONE=0` , `CONSOLE=1` , `FILE=2` , `CONSOLE+FILE=3` , `WITH_PID=4`]
- eg: `console` + `file` + `pid file name` = 1 + 2 + 4 = 7, so the `-Djanf.output=7`

Plugin system

- for developer:
  - view the scaffold project written for the plugin system
  - compile your plugin and publish it
  - just use your imagination~
- for user:
  - download the jar file of the plugin
  - put it in the subdirectory called `plugins` where the ja-netfilter.jar file is located

- enjoy the new capabilities brought by the plugin
- if the file suffix is `.disabled.jar`, the plugin will be disabled