BADAPI Github Keyword Alerts

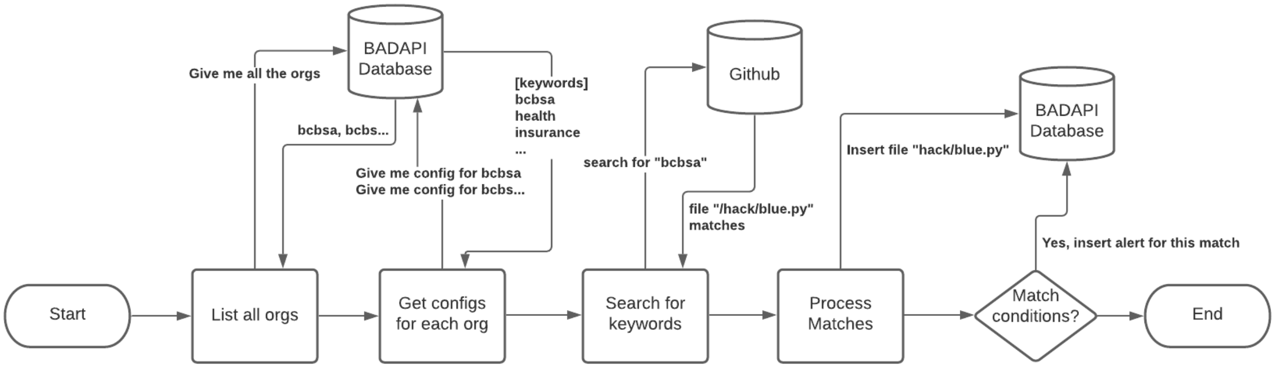
Description

The GitHub Keyword Alerts service will search GitHub for keywords defined in a configuration file (githubsearch.config) and generate alerts when a match to the condition(s) defined in that configuration file is found.

The generated alerts are retrieved using the bimeta client command “get github.alerts”.

Program Flow

Below is a simplified representation of the program flow.



Setting up the configuration file

A sample configuration file will look as follows:

org - The org section is optional and is not used by the service.

keywords – One or more sections beginning with “keywords” is used to denote search terms. A set of these can be used to create rules; for example: keywords, keywords.domains, keywords.highfidelity. As a note “keywords.self” is synomous with “keywords”

keywords.conditions – This section is reserved to create YARA style rules to indicate what is a match. In this example, if the GitHub repo has two different words from the “keywords” section or one word from the “keywords.domain” section; and, no words from the negative section is found, a match is indicated.

Filename: githubsearch.config

```
[org]
blue cross blue shield association

[keywords]
association
health
bcbsa|fullword

[keywords.domains]
bcbsafep.com

[negative]
testing

[keywords.conditions]
2 in $keywords.self or 1 in $keywords.domains and none in $negative
```

Usage

Invoking –help on the command will provide the following:

```
./badapi_client get github.alerts –help
usage: badapi_client get github.alerts [<flags>]

Get gibhub search alerts

Flags:
      --help         Show context-sensitive help (also try --help-long and --help-man).
      --config="/home/michael/Developer/go/src/blueintel/cmd/badapi_client/bimeta_config.toml"
                     Filename of the configuration file. Default is bimeta_config.toml in the executable directory.
      --format=json  Format in which to output the results (csv|json)
      --header=yes   Include header for csv output (yes|no)?
  -o, --org=ORG      Get alerts for this org; defaults to user org
  -f, --from=FROM    Starting date for alerts. Format '2006-01-02 15:04:06'
  -t, --to=TO        Ending date for alerts; defaults to now. Format '2006-01-02 15:04:06'
  -v, --viewed       Include previously viewed alerts
```

This table will further describe the options:

help	Displays the above usage information
config	Sets the location of the configuration file for badapi_client to retrieve the apikey
format	Not currently implemented. Default format is currently fixed at json
org	Defaults to the api user's org. Can be used to specify any org the api user has delegation rights to. Only alerts meant for the specified org will be returned
from	Starting date for the alerts
to	Ending date for the alerts
viewed	Once alerts are viewed, they are marked as viewed by the api user and no longer retrieved again unless the viewed option is given. It essence it's a show me the one's I've already seen.

Output "fields" are as follows:

ID	ID of the alert
RepoURL	URL of the github repository
Keywords	The keywords which were matched by this repo
Timestamp	Time/date the alert was generated
Matches – File	The file within the repo this particular match is from
Matches – Fragments	List of text fragment(s) which matched the keyword(s)

Sample output:

```
{
  "ID": "5f3cddd80823fa40efd1c490",
  "RepoURL": "https://github.com/Suuresh189/KeyWordUsingInvoke",
  "Keywords": [
    "bcbsafep.com"
  ],
  "Timestamp": "2020-08-19T04:07:52-04:00",
  "Matches": [
    {
      "File": "objectmap.properties",
      "Fragments": [
        "fepdirect.url=name\\=http\\://fepdirectf2.bcbsafep.com/fepdirect/login.do\r\n\r\nfepdirect.login.UserName"
      ]
    }
  ]
}
{
  "ID": "5f3cddd80823fa40efd1c492",
  "RepoURL": "https://github.com/manishdube/Automation_framework",
  "Keywords": [
    "fepblue.org"
  ],
  "Timestamp": "2020-08-19T04:07:52-04:00",
  "Matches": [
    {
      "File": "BCBSA_SIT_CI_tobeDeleted/sit/Program.fs",
      "Fragments": [
        " \"stage.fepblue.org\"\n\n    let liveHtmlReporter  = reporter :?\u003e LiveHtmlReporter\n    liveHtmlReporter"
      ]
    },
    {
      "File": "BCBSA_Canopy_CI/sit/Program.fs",
      "Fragments": [
        " \"stage.fepblue.org\"\n\n    let liveHtmlReporter  = reporter :?\u003e LiveHtmlReporter\n    liveHtmlReporter"
      ]
    }
  ]
}
```