dox

Abstract

Docker API proxy : allow filtered (regexp) access to Docker API thru docker.sock + GUI with HTML embedded GO templates

dox -

Docker API proxy : allow filtered (regexp) access to Docker API thru docker.sock + GUI with HTML embedded GO templates

DRAFT - incomplete doc

DOX : Docker API Proxy

Requisites

- a linux server
- an apache server with php enabled
- gcc
- bash
- golang
- root access
- should I mention a running dockerd

Installation example

- `git clone` to `/opt/dox`
- compile demux.c : `gcc -odemux demux.c`
- compile gotemp.go : `go build golang.go`
- apache : alias `/dox/` (URL) to `/opt/dox/publish/dox/` (FS)
- create proxying user : `sudo adduser dockerlogs`
- add user to `docker` group \
- `sudo visudo -f /etc/sudoers.d/dockerlogs`

```
Cmnd_Alias DCOM = /opt/dox/bin/dox *
User_Alias DUSERS = www-data, %dockerlogs
DUSERS ALL = (dockerlogs) NOPASSWD: DCOM
```

- change `BDIR` value in `/opt/dox/bin/dox` to match installation directory
- add/remove regexps in `dox-regexp.txt` to allow/disallow docker WS accesses

Request processing

Supposing the request is to be sent at https://docker-host.bzh/dox, like in https://docker-host.bzh/dox/version * the apache `.htaccess` rewrite the request to `dox.php` * the actual Docket WS path is extracted * and passed to `dox` bash script. The `-x` parameter tells `dox` bash script to output the correct content-type * dox check if the WS is allowed * if yes call the actual Docker WS * if the WS is tagged as "stream" it uses the demux tool to output a plain text file (for logs for example)

Particularity

There are a few differences between the genuine API and the proxied ones.

GUI

- lightweight PHP app structure with a DoxApp application singleton
- inline GO template for JSON to HTML conversion
- `doxplus.php` receives a WS request and a GO Template. It runs the `dox` bash script and uses the result to run `gotemp` against the template, returning plain HTML to be dealt by jQuery
- same requisites as previous + golang