SpliceProxy

Transparent proxy for http and https sites. Doesn't implement ssl bumping but routes **SSL** queries in function of **server name indication**. Standard **http** requests uses the **Host header**.

The list of autorized servers need to be provided via a whitelist.

Unauthorized requests can be routed to another server or to a hosted basic page.

Installation

Download the binary corresponding to your platform.

Configure via the yaml file (see bellow).

Start:

```
spliceproxy -c config.yaml
```

To install as a service on a systemd linux disribution (i.e centos/redhat/ubuntu):

```
cp sliceproxy /usr/local/bin/
cp ./systemd/spliceproxy.service /etc/systemd/system/
cp ./config.yaml /etc/spliceproxy.yaml
systemctl start spliceproxy
```

This would be with default pathes. It can be customised to your requirements.

Configuration

Basic configuration is done trought a yaml file passed as argument. By default the file is named "config.yaml" and searched in the running path.

```
timeout: 10

buffer: 1024

catchall:
    http: 127.0.0.1:8080
    https: 127.0.0.1:8443

listen:
    https:
        - 0.0.0.0:443
    http:
        - 0.0.0.0:80

alloweddomains:
    - example.com
    - github.com
```

In this example configuraiton:

- Timeout: All requests have a 10 seconds. This can be tuned to your websites. Long running queries will need a longer timeout.
- Buffer: The maximum size of http headers to look at.
- CatchAll: indicates where to redirect the users when access is denied
- Listen: host and port where to listen to requests. These would be registered in your internal dns. The distinction between http and https is there to indicate how to detect the targeted site. Another solution is to redirect all http/https requests of your router to the proxy.
- AllowedDomains: the list of domain name autorised via the proxy. Subdomains will be autorised

Host access denied page in application

The application needs its own certiticate to provide SSL denied page.

```
openssl genrsa -out /etc/spliceproxy.key 2048
openssl req -new -x509 -sha256 -key /etc/spliceproxy.key -out /etc/spliceproxy.crt -days 3650
```

In the configuration:

```
catchall:
    server: true
    key: /etc/spliceproxy.key
    cert: /etc/spliceproxy.pem
    http: 127.0.0.1:8080
    https: 127.0.0.1:8443
```

With this configuration the server will:

- serve the access denied page
- use the provided certificate (key and crt pair)
- listen for http requests on localhost (127.0.0.1) on port 8080
- listen for https requests on localhost (127.0.0.1) on port 8443

License

See License