Securing SWAG

Securing SWAG

SWAG is a reverse proxy that allows you to expose your self-hosted apps to the world, but that comes with risks.

We can mitigate some risks by fine-tuning SWAG and how we access it:

- Prevent accessing some apps via the internet while exposing others.
- Set up brute-force protection via Crowdsec/Fail2Ban.
- Set up geoblock to whitelist/blacklist countries via DBIP/Maxmind.
- Prevent your apps from appearing in search results.
- Set up SSO via Authelia/Authentik.
- Monitor SWAG via a dashboard.
- Access your apps through Wireguard instead of exposing them.

Requirements

- A working instance of SWAG.

Internal Applications

Only expose apps you want to share with others and must expose, keep the rest internal and use WireGuard to access them.

Requirements

- Split DNS - the source IP on requests needs to be local for allow/deny to work properly.

Create a file called `nginx/internal.conf` with the following configuration:

```
allow 192.168.1.0/24; # Replace with your LAN subnet
deny all;
```

Utilize the LAN filter in your configuration by adding the following line inside the server block for every application you want to protect.

```
    include /config/nginx/internal.conf;
```

Example:

```
server {
    listen 443 ssl;
    listen [::]:443 ssl;

    server_name collabora.*;
    include /config/nginx/ssl.conf;
    client_max_body_size 0;
    include /config/nginx/internal.conf;

    location / {
        include /config/nginx/proxy.conf;
        include /config/nginx/resolver.conf;
        set $upstream_app collabora;
        set $upstream_port 9980;
        set $upstream_proto https;
        proxy_pass $upstream_proto://$upstream_app:$upstream_port;
    }
}
```

Repeat the process for all internal applications.

Brute-Force Protection

Crowdsec and Fail2Ban can prevent brute-force attacks by monitoring the logs of apps and banning IPs that fail multiple login attempts.

SWAG comes with Fail2Ban pre-configured with a few basic protections, you can fine-tune it specifically for your apps, or disable it and set up Crowdsec instead.

Crowdsec

Crowdsec is a free, open-source and collaborative IPS; it's like fail2ban but you share your bans with all of the other users to try and pre-emptively block malicious hosts.

Follow this blog post to set it up in SWAG.

Fail2Ban

Fail2Ban is an intrusion prevention software that protects external applications from brute-force attacks. Attackers that fail to login to your applications a certain number of times will get blocked from accessing all of your applications. Fail2Ban looks for failed login attempts in log files, counts the failed attempts in a short period, and bans the IP address of the attacker.

The following is an example of setting up Nextcloud in Fail2Ban, configure other apps in the same way.

Mount the application logs to SWAG's container by adding a volume for the log to the compose yaml:

```
      - /path/to/nextcloud/logs:/nextcloud:ro
```

Note that you should only mount the parent folders, log files can rotate.

Recreate the container with the log mount, then create a file called `nextcloud.local` under `fail2ban/filter.d`:

```
[Definition]
failregex=^.*Login failed: '?.*'? \(Remote IP: '?<ADDR>'?\).*$
          ^.*\"remoteAddr\":\"<ADDR>\".*Trusted domain error.*$
ignoreregex =
```

Before making your own Fail2Ban filter for apps search online for existing one, most chances someone already made it.

The filter containes a pattern by which failed login attempts are matched. Test the pattern by failing to login to nextcloud and look for the entry corresponding to your failed attempt.

```
{"reqId":"k5j5H7K3eskXt3hCLSc4i","level":2,"time":"2020-10-14T22:56:14+00:00","remoteAddr":"1.2.3.4","user":"--",
"app":"no app in context","method":"POST","url":"/login","message":"Login failed: username (Remote IP: 5.5.5.5)",
"userAgent":"Mozilla/5.0 (Linux; Android 11; Pixel 5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/5.6.7.8 Mobile
Safari/537.36","version":"19.0.4.2"}
```

Test the pattern in `nextcloud.local` by running the following command on the docker host:

```
docker exec swag fail2ban-regex /nextcloud/nextcloud.log /config/fail2ban/filter.d/nextcloud.local
```

If the pattern works, you will see matches corresponding to the amount of failed login attempts:

```
Lines: 92377 lines, 0 ignored, 2 matched, 92375 missed
[processed in 7.51 sec]
```

The final step is to activate the jail, add the following to `fail2ban/jail.local`:

```
[nextcloud]
```

```
enabled = true
port    = http,https
filter  = nextcloud
logpath = /nextcloud/nextcloud.log
action  = iptables-allports[name=nextcloud]
```

The logpath is slightly different for applications that have multiple log files with dates:

```
[jellyfin]
enabled  = true
filter   = jellyfin
port     = http,https
logpath  = /jellyfin/log*.log
action   =  iptables-allports[name=jellyfin]
```

Repeat the process for every app you expose, you can find Fail2Ban configurations for most applications on the internet.

If you need to unban an IP address that was blocked, run the following command on the docker host:

```
docker exec swag fail2ban-client unban <ip address>
```

Geoblock

Geoblock significantly reduces the attack surface of SWAG by restricting access based on countries.

Follow the instructions of one of the following mods to set it up:

- DBIP mod
- Maxmind mod

DBIP doesn't require an account, but Maxmind might be more accurate in some cases.

Search Results

You can prevent apps from appearing in search engine results and being crawled by web crawlers.

Note that not all search engines and web crawlers respect this tag, but it significantly reduces the amount.

Add the following to `ssl.conf` to enable it on **all** apps:

```
add_header X-Robots-Tag "noindex, nofollow, nosnippet, noarchive";
```

To disable on a specific app, add the following line to the app's proxy-conf inside the server tag:

```
add_header X-Robots-Tag "";
```

SSO

Setting up SSO will provide an additional layer of security and protect you against login bypass exploits in apps.

- Authelia
- Authentik

Note that api endpoints shouldn't have SSO for them to function properly.

Monitor

Use monitoring solutions such as SWAG Dashboard to keep an eye on the traffic going through SWAG and check for suspicious activity such as:

- Many hits from a country unrelated to your users.

- Many requests to a specific page or static file.
- Referers that shouldn't refer to your domain.
- Many hits on status codes that are not 2xx.

VPN

The most effective security you can implement is to stop exposing your apps entirely, and instead access them via WireGuard.

Requirements

- A working instance of WireGuard.
- Split DNS - the source IP on requests needs to be local for SWAG to work without being exposed.
- DNS Validation - allows you to get an SSL certificate without port forwarding.

Once you've set up wireguard, split DNS, and DNS validation, you can remove the port forwarding on your router and remove your domain's public DNS records on your public DNS provider (not the local DNS).