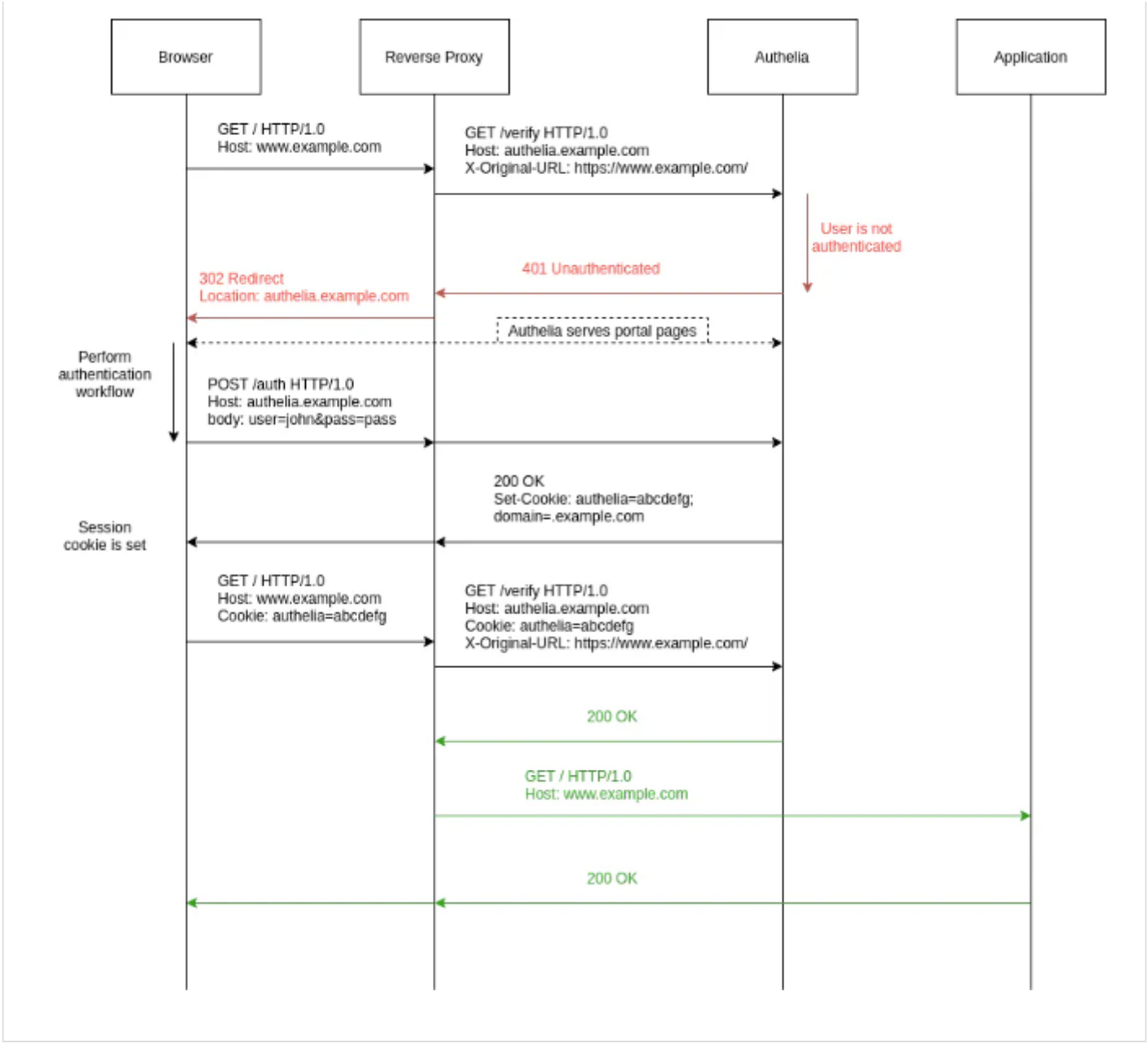
SSO Details

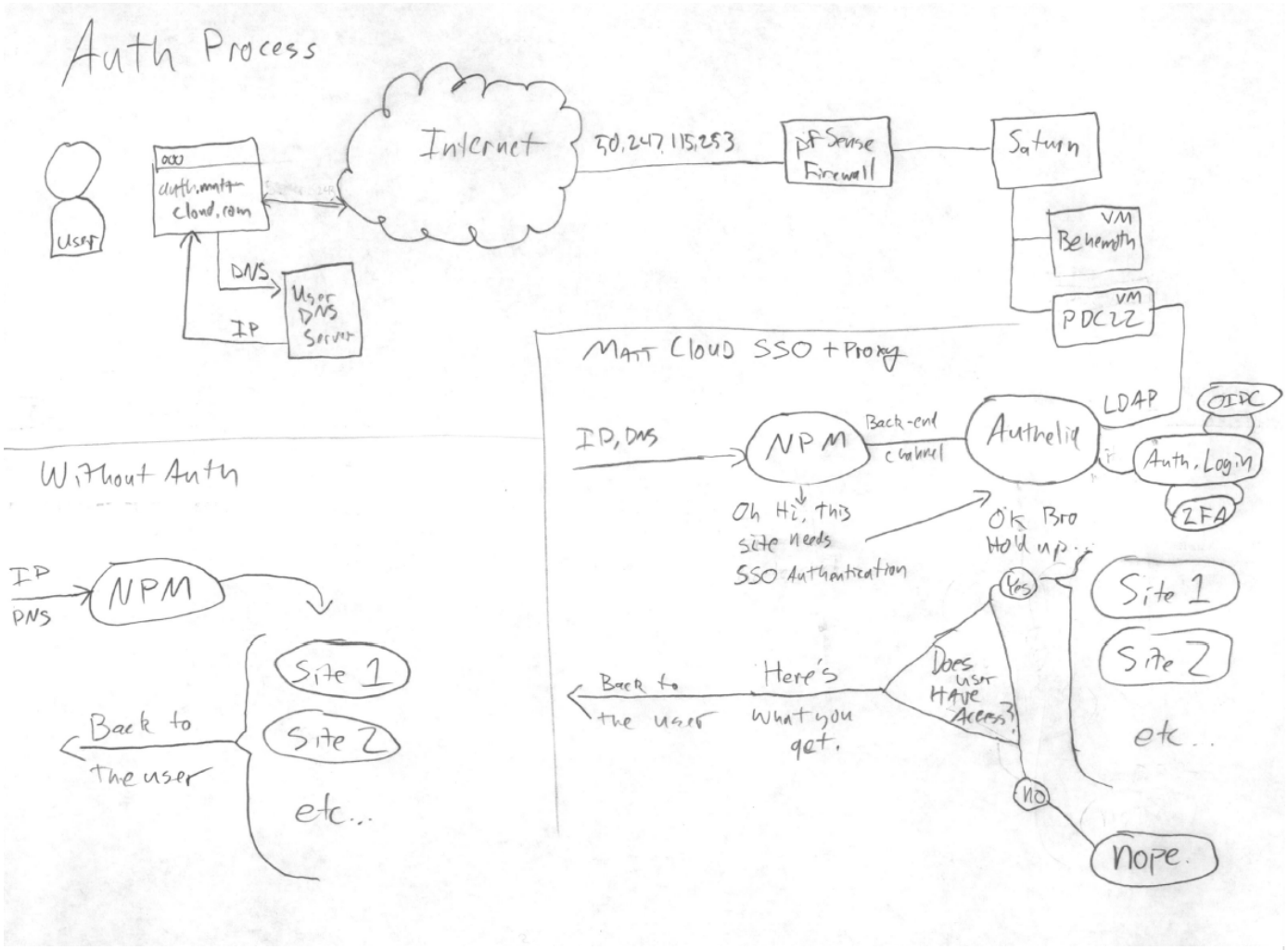
The SSO is handled by Authelia, and a lot of my configs are based on this page. It was pretty difficult to get this going, but now that it is I will give an overview of how it works. It is integrated with the proxy server so that when SSO protected domains are requested my proxy server hands the request off to Authelia for authentication. Authelia sets a cookie when you first log in, so if you've already logged in somewhere, it will just pass you through to the website. I also was able to cobble together OIDC integration with Authelia, and that is integrated with my audiobook site as well as this book stack site.

Here's a diagram of how the proxy, SSO, and containers interact. The **site** items with ovals are the docker containers with open ports that the proxy server would normally forward unfettered. On sites that need SSO, I add a big pile of code to the custom part of the NPM site config that tells it to finish up with Authelia. Then, Authelia does it's stuff based on the config file. The **nope** in the oval is the *forbidden* page that you see when you are logged into SSO and try to access a site that Authelia says **nope** to.

Here is Authelia's diagram of how they do stuff



Here is my diagram of how it works on Matt-Cloud.



For the terribly curious, here is a stripped out docker-compose.yaml and configuration.yml for authelia and the nginx configs.

docker-compose.yaml

```
services:


  authelia:
    container_name: authelia
    image: 'authelia/authelia'
    restart: always
    network_mode: bridge
    ports:
      - 9091:9091
    volumes:
```

```
    - ./authelia-config:/config
    - ./authelia-secrets:/secrets

  authelia-redis:
    container_name: authelia-redis
    image: bitnami/redis:latest
    volumes:
      - ./authelia-redis:/bitnami/
    environment:
      REDIS_PASSWORD: "lolwtfbbqlolwtfbbq"
    restart: always
    network_mode: bridge
    ports:
      - 6379:6379

  authelia-db:
    image: postgres
    container_name: authelia-db
    restart: always
    network_mode: bridge
    volumes:
      - ./authelia-db:/var/lib/postgresql/data
    environment:
      POSTGRES_DB: authelia
      POSTGRES_USER: authelia
      POSTGRES_PASSWORD: lolwtfbbqlolwtfbbq
    ports:
      - 5432:5432
```

configuration.yml

```
---
###############################################################################
#                   Authelia Configuration                    #
###############################################################################
```

```
###############

theme: dark
jwt_secret: "lolwtfbbqlolwtfbbq"
default_redirection_url: https://auth.domain.com/

server:
  host: 0.0.0.0
  port: 9091
  disable_healthcheck: false

log:
  level: info

totp:
  disable: false
  issuer: 'auth.domain.com'
  algorithm: 'sha1'
  digits: 6
  period: 30
  skew: 1
  secret_size: 32
  allowed_algorithms:
    - 'SHA1'
  allowed_digits:
    - 6
  allowed_periods:
    - 30
  disable_reuse_security_policy: false

authentication_backend:
  ldap:
    address: 'ldap://pdc.domain.local:389'
    implementation: 'activedirectory'
    base_dn: 'OU=users,DC=domain,DC=local'
    users_filter:
(&(|({username_attribute}={input})({mail_attribute}={input}))(objectCategory=person)(objectClass=user
)(!userAccountControl:1.2.840.113556.1.4.803:=2)(!pwdLastSet=0))
    groups_filter: (&(member:1.2.840.113556.1.4.1941:={dn})(objectClass=group)(objectCategory=group))
```

```
    group_name_attribute: cn
    mail_attribute: mail
    display_name_attribute: displayname
    user: 'CN=LDAP Service,OU=Service,OU=users,DC=domain,DC=local'
    password: 'lolwtfbbq'
  disable_reset_password: true

access_control:
  ## just some simple example rules
  default_policy: deny
  rules:
    ## bypass rule
    - domain:
        - "auth.domain.com"
      policy: bypass
    - domain:
        - "*.domain.com"
      resources:
        - "^/api([/?].*)?$"
      policy: bypass
    ## 2fa domain
    - domain:
        - "2fa.domain.com"
      policy: two_factor
      subject:
        - "group:2FA-Users"
    # Normal protection
    - domain:
        - "secure.domain.com"
      policy: one_factor
      subject: "group:secure-users"

session:
  name: authelia_session
  domain: domain.com
  same_site: lax
  secret: "lolwtfbbqlolwtfbbq"
  expiration: 30d
  inactivity: 1d
```

```
  remember_me_duration: 6M
  redis:
    host: 0.0.0.0
    port: 6379
    password: "lolwtfbbqlolwtfbbq"
    database_index: 0
    maximum_active_connections: 10
    minimum_idle_connections: 0


regulation:
  max_retries: 3
  find_time: 10m
  ban_time: 12h


storage:
  encryption_key: 'lolwtfbbqlolwtfbbq'
  postgres:
    address: 'tcp://0.0.0.0:5432'
    database: 'authelia'
    schema: 'public'
    username: 'authelia'
    password: 'lolwtfbbqlolwtfbbq'


notifier:
  disable_startup_check: false
  smtp:
    username: authelia@domain.com
    password: "lolwtfbbqlolwtfbbq"
    host: mail.domain.com
    port: 587
    sender: authelia@domain.com
    identifier: authelia.domain.local
    subject: "[Authelia] {title}"
    startup_check_address: user@domain.com
    disable_require_tls: false
    disable_html_emails: true
```

```
    tls:
      skip_verify: false
      minimum_version: TLS1.2


identity_providers:
  oidc:
    hmac_secret: lolwtfbbqlolwtfbbq # provide secure secret
    issuer_certificate_chain: |
      -----BEGIN CERTIFICATE-----
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      -----END CERTIFICATE-----
    issuer_private_key: |
      -----BEGIN PRIVATE KEY-----
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      lolwtfbbqlolwtfbbq
      -----END PRIVATE KEY-----
    access_token_lifespan: 1h
    authorize_code_lifespan: 1m
    id_token_lifespan: 1h
    refresh_token_lifespan: 90m
    enable_client_debug_messages: false
```

```
enforce_pkce: public_clients_only
cors:
  endpoints:
    - authorization
    - token
    - revocation
    - introspection
  allowed_origins:
    - https://*.domain.com # adjust to your url
  allowed_origins_from_client_redirect_uris: false
clients:
  - id: domain-oidc
    description: domain
    secret: 'lolwtfbbqlolwtfbbq' # provide secure secret
    sector_identifier: 'auth.domain.com'
    public: false
    authorization_policy: one_factor # may use two_factor to enforce 2FA
    consent_mode: implicit
    pre_configured_consent_duration: 6m
    audience: []
    scopes:
      - openid
      - groups
      - email
      - profile
    redirect_uris: # adjust to your domains
      - https://auth.domain.com/
      - https://auth.domain.com/oauth2/callback
      - https://audiobooks.domain.com/oauth2/callback
      - https://audiobooks.domain.com/auth/login
      - https://audiobooks.domain.com/user-settings
      - https://audiobooks.domain.com
      - https://audiobooks.domain.com/auth/openid/callback
    grant_types:
      - refresh_token
      - authorization_code
      - implicit
    response_types:
      - code
```

```
    - id_token
  response_modes:
    - form_post
    - query
    - fragment
  userinfo_signing_algorithm: none
```

nginx-server

```
location / {
set $upstream_authelia http://172.17.0.1:9091; # This example assumes a Docker deployment
proxy_pass $upstream_authelia;
client_body_buffer_size 128k;

#Timeout if the real server is dead
proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;

# Advanced Proxy Config
send_timeout 5m;
proxy_read_timeout 360;
proxy_send_timeout 360;
proxy_connect_timeout 360;

# Basic Proxy Config
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $http_host;
proxy_set_header X-Forwarded-Uri $request_uri;
proxy_set_header X-Forwarded-Ssl on;
proxy_redirect  http://  $scheme://;
proxy_http_version 1.1;
proxy_set_header Connection "";
proxy_cache_bypass $cookie_session;
proxy_no_cache $cookie_session;
proxy_buffers 64 256k;
```

```
# If behind a reverse proxy, forwards the correct IP, assumes you're using Cloudflare. Adjust IP for your Docker network.
set_real_ip_from 172.17.0.0/16;
set_real_ip_from 10.0.0.0/8;
real_ip_header X-Forwarded-For;
real_ip_recursive on;
}
```

nginx-client

```
location /authelia {
internal;
set $upstream_authelia http://0.0.0.0:9091/api/verify;
proxy_pass_request_body off;
proxy_pass $upstream_authelia;
proxy_set_header Content-Length "";

# Timeout if the real server is dead
proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;
client_body_buffer_size 128k;
proxy_set_header Host $host;
proxy_set_header X-Original-URL $scheme://$http_host$request_uri;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $remote_addr;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $http_host;
proxy_set_header X-Forwarded-Uri $request_uri;
proxy_set_header X-Forwarded-Ssl on;
proxy_redirect  http://  $scheme://;
proxy_http_version 1.1;
proxy_set_header Connection "";
proxy_cache_bypass $cookie_session;
proxy_no_cache $cookie_session;
proxy_buffers 4 32k;

send_timeout 5m;
proxy_read_timeout 240;
proxy_send_timeout 240;
```

```
proxy_connect_timeout 240;
}

location / {
set $upstream_app $forward_scheme://$server:$port;
proxy_pass $upstream_app;

auth_request /authelia;
auth_request_set $target_url https://$http_host$request_uri;
auth_request_set $user $upstream_http_remote_user;
auth_request_set $email $upstream_http_remote_email;
auth_request_set $groups $upstream_http_remote_groups;
proxy_set_header Remote-User $user;
proxy_set_header Remote-Email $email;
proxy_set_header Remote-Groups $groups;

error_page 401 =302 https://auth.domain.com/?rd=$target_url;

client_body_buffer_size 128k;

proxy_next_upstream error timeout invalid_header http_500 http_502 http_503;

send_timeout 5m;
proxy_read_timeout 360;
proxy_send_timeout 360;
proxy_connect_timeout 360;

proxy_set_header Host $host;
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection upgrade;
proxy_set_header Accept-Encoding gzip;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
proxy_set_header X-Forwarded-Host $http_host;
proxy_set_header X-Forwarded-Uri $request_uri;
proxy_set_header X-Forwarded-Ssl on;
proxy_redirect  http://  $scheme://;
proxy_http_version 1.1;
```

```
proxy_set_header Connection "";
proxy_cache_bypass $cookie_session;
proxy_no_cache $cookie_session;
proxy_buffers 64 256k;

set_real_ip_from 172.17.0.0/16;
set_real_ip_from 10.0.0.0/8;
real_ip_header X-Forwarded-For;
real_ip_recursive on;

}
```

Revision #12
Created 2 April 2024 00:50:27 by Matt Anderson
Updated 12 October 2025 23:34:27 by Matt Anderson