Proxy - Configuration for Microsoft Azure

1.What is proxy?

A Proxy Server is generally used to restrict the direct communication form client to server or vice-versa. It acts as a gateway between client and server.

Modern Proxy servers are can provide functionalities like load balancing, security in the form of firewall, web filter, shared network connections and cache data to speed up common request.

If a Proxy server is used, then every web request will pass through the proxy server to internet and the response will reach to the client through the same proxy server.

A proxy server is basically a computer on the internet with its own IP address that client computer knows. When the client send a web request, the request goes to the proxy server first. The proxy serve then makes your web request, collects the response from the Web Server and forwards the Web Page data so that the client can see the web page in the browser.

2.What are the different types of Proxy Servers available?

2.1 Forward Proxy

It send the client request to a web server. Users access forward proxies by directly surfing to a web proxy address or by configuring their internet settings. It is configured to handle requests for a group of clients.

A good example is a web proxy appliance which access web traffic requests from client machines in local networks and proxy them to servers on the internet.

Here, the server thinks all request are made by the forward proxy when in fact it would be from multiple client machines sitting behind the forward proxy.

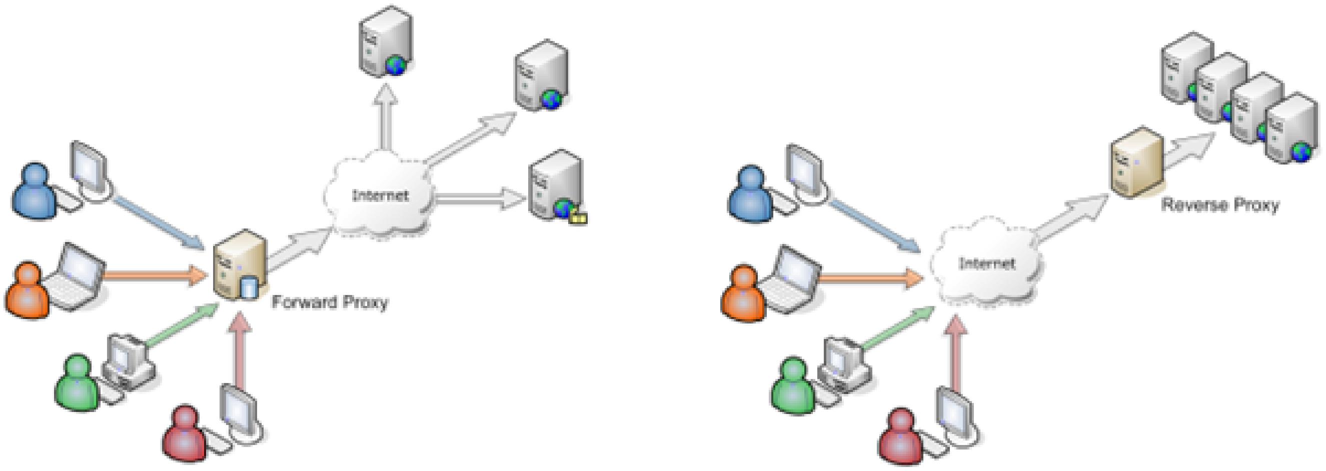
2.2 Reverse Proxy

It handles all requests for resources on destination servers. It is used to handle requests from a group of remote or arbitrary clients to a group of know resources.

An example is a load balancer (a.k.a. application delivery controller) that provides application h high availability and optimization to workloads like as Microsoft Skype, Exchange and SharePoint.

1. It is used to enable indirect access when a website disallows direct connections as a security measure.
2. Also used for load balancing between the servers and to stream internal content to Internet users.
3.

Reverse Proxy is used to disable access to site, for e.g when and ISP to government wishes to block a website. Reverse proxy may also be used to prevent access to immoral, illegal or copyright content.



Picture of Forward and Reverse Proxy Servers

2.3 Transparent Proxy

These are typically found near the exit of a corporate network. These are associated with, or is part of, a gateway server that separates the network from external networks and a firewall that protects the network from outside intrusion and allows data to be scanned for security purposes before delivery to a client on the network.

2.4 Anonymous Proxy

A server that functions as a relay between the user and a destination website. It hides the IP address of the user's machine from the website and may provide encryption on the user side.

2.5 Highly Anonymous Proxy

They hide even the fact that they are being used by clients and present a non-proxy public IP address. So not only do they hide the IP address of the client using them, they also allow access to sites that might block proxy servers.

A **high anonymity** has all the advantages that an **anonymous proxy** has in terms of its privacy and it also has the capacity to conceal that a user is using a **proxy** server to connect to the internet.

Examples: **I2p** and **TOR** (Third generation Onion Routing)

Other available proxies are Socks 4 and 5 proxies DNS proxies e.t.c.

3.What are the proxy provider?

There are variety of proxy servers available as freeware and as paid ones. However most of paid provides will also provide a free trail versions also.

3.1 NGINX and NGINX Plus - https://www.nginx.com/

Nginx is open software for web serving, reverse proxying, caching, load balancing, and more.

It started out as a web server designed for maximum performance and stability. In addition to its HTTP server capabilities, NGINX can also function as a proxy server for email (IMAP, POP2 and SMTP) and a reverse proxy and load balanver for HTTP, TCP and UDP servers.

NGINX is now part of F5.

3.2 Squid - http://www.squid-cache.org/

Squid is a caching proxy for Web supporting HTTP, HTTPS, FTP and more, It reduces bandwidth and improves response times by caching and reusing frequently-requested web pages. It is licensed under GNU GPL.

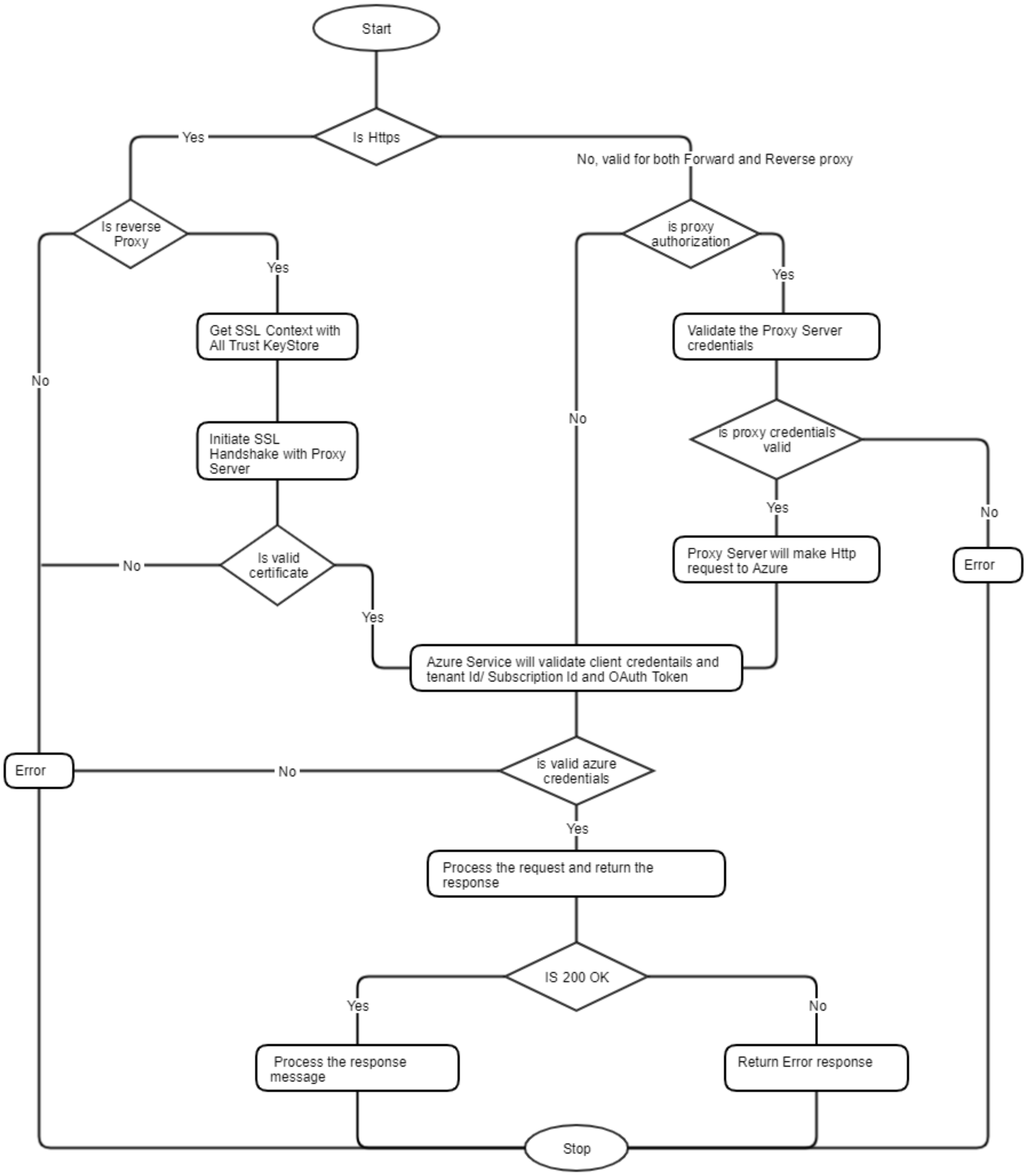
3.3 Privoxy – https://www.privoxy.org/

Privoxy is a free non-caching web proxy with filtering capabilities for enhacing privacy, manipulating cookies and modifying web page data and HTTP headers before the page is rendered by the browser. It is a “privacy enhancing proxy”, filtering web pages and removing advertisements.

3.4 EZproxy - https://www.oclc.org/en/ezproxy.html

EZproxy is a web proxy server used by libraries to give access from outside the library’s computer network to restricted-access websites that authenticate users by IP address.

4. Flow Char of APM IA Azure Extension for Proxy Support


Start
Is Https
Yes
No, valid for both Forward and Reverse proxy
Is reverse Proxy
Yes
No
Get SSL Context with All Trust KeyStore
Initiate SSL Handshake with Proxy Server
Is valid certificate
No
Yes
is proxy authorization
Yes
No
Validate the Proxy Server credentials
is proxy credentials valid
Yes
No
Proxy Server will make Http request to Azure
Error
Azure Service will validate client credentails and tenant Id/ Subscription Id and OAuth Token
is valid azure credentials
No
Error
Yes
Process the request and return the response
IS 200 OK
Yes
No
Process the response message
Return Error response
Stop

5. Nginx as Reverse Proxy configurations

5.1 HTTP without Authorization

```
server {
   listen 8089;
  location /subscriptions/ {
         proxy_pass https://management.azure.com;
        access_log azure_subscription_log.txt;
   }
```

```
      location / {

                  proxy_pass https://login.microsoftonline.com;

                  access_log azure_oauth2_log.txt;

      }

    }
```

Server listens to port 8089 and will allow

- Any connection request contains ‘subscription’ as the part of url
- Any client request for a resource at root ‘/’, mentioned in url

It then forwards the request to the **URL** mentioned at **proxy_pass** directive.

The access logs will be generated at the location mentioned at **access_log** directive.

5.2. HTTP with Authorization

```
   server {

       listen 8088;

       location / {

              auth_basic           "Administrator’s Area";

              auth_basic_user_file /etc/nginx/htpasswd;

              proxy_pass https://login.microsoftonline.com;

              access_log azure_oauth2_log.txt;

       }

      location /subscriptions/ {

             proxy_pass https://management.azure.com;

             access_log azure_subscription_log.txt;

      }

 }
```

Server listens to port 8088 and will allow

- Any client request with proxy credentials for a resource at root ‘/’.
- Any connection request contains ‘subscription’ as the part of url

The above configuration will authenticate any connection request using the username and password mentioned in the file located at ‘**auth_basic _user_file’** directive.

Any connections with wrong credentials will be rejected.

Please note that we have restricted the authorization only to root **‘/’**. This is because the later url, i.e **/subscriptions/**, requires an OAuth token. And the token will be sent as a ‘Authorization’ header in http request.

So, it not feasible to send both proxy authorization and OAuth token using the same ‘Authorization’ header.

- **How to generate authentication credentials for nginx?**

  The following commands are used to create the password file

  sudo htpasswd -c /etc/nginx/htpasswd user1

  Press Enter and type password for user1

  Refer: https://docs.nginx.com/nginx/admin-guide/security-controls/configuring-http-basic-authentication/

5.3 SSL Configuration:

```
server {
    listen 443 ssl default_server;
     ssl_certificate /root/myClientCert/MyCertificate.crt;
     ssl_certificate_key /root/myClientCert/MyKey.key;
     ssl_ciphers EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH;
     ssl_protocols TLSv1.1 TLSv1.2;

    location /subscriptions/ {
                proxy_pass https://management.azure.com;
                access_log azure_subscription_log.txt;
     }

    location / {
                 proxy_pass https://login.microsoftonline.com;
                access_log azure_oauth2_log.txt;
      }
   }
```

Server listens to port 443 and will allow

- Any client request for a resource at root **'/'**.
- Any connection request contains 'subscription' as the part of url

It will then initiate the SSL Handshake for any further communication. If the Client cannot validate the Server Certificate, then the communication will fail.

- **How to generate required server certificate, server keys and client .p12 files?**

Command to generate Server Certificate and Server Key

```
openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out MyCertificate.crt -keyout MyKey.key
```

6. Squid as Forward Proxy

6.1 HTTP without Authorization

Squid normally listens to port **3128** (user can change the port if needed)

Add the following rule allowing access from your local networks. (List internal IP networks)

```
acl localnet src 0.0.0.1-0.255.255.255          # RFC 1122 "this" network (LAN)
```

Comment the CONNECT method configuration to allow proxy to create tunnel with usual HTTP methods

```
#acl CONNECT method CONNECT
```

Allow CONNECT to any port

```
#http_access deny CONNECT !SSL_ports
```

https://en.wikipedia.org/wiki/HTTP_tunnel

6.2 HTTP with Authorization

Squid normally listens to port **3128** (user can change the port if needed)

```
# Basic Authentication
auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid Basic Authentication
auth_param basic credentialsttl 2 hours
acl auth_users proxy_auth REQUIRED
http_access allow auth_users
```

Uncomment the CONNECT method configuration to allow proxy to create tunnel with usual HTTP methods

```
acl CONNECT method CONNECT
```

```
# Deny CONNECT to other than secure SSL ports
http_access deny CONNECT !SSL_ports
```

```
# Generate a pxuser user with pass pxpass for basic authentication
htpasswd -b -c /etc/squid/passwd pxuser pxpass
```

If **httpd-tools** are not installed, then you can install them using the following **yum** command.

```
yum -y install squid squid-helpers httpd-tools
```

7.References

1. https://www.varonis.com/blog/what-is-a-proxy-server/
2. https://whatis.techtarget.com/definition/proxy-server
3. https://docs.nginx.com/nginx/admin-guide/basic-functionality/runtime-control/

TOR (Third-generation Onion Routing)

1. https://whatis.techtarget.com/definition/TOR-third-generation-onion-routing