


Setting up SSL and authentication for Kibana

By default, the communications between Kibana (including the Wazuh app) and the web browser on end-user systems are not encrypted. It's strongly recommended to configure Kibana to use SSL encryption and to enable authentication, next we briefly describe how to do this with a NGINX setup.

NGINX is a popular open-source web server and reverse proxy, known for its high performance, stability, rich feature set, simple configuration, and low resource consumption. Here we will use it as a reverse proxy to provide to the end users an encrypted and authenticated access to Kibana.

 **Note**

Many of the commands described below need to be executed with root user privileges.

Contents

- 1. [NGINX SSL proxy for Kibana \(RPM-based distributions\)](#)
- 2. [NGINX SSL proxy for Kibana \(Debian-based distributions\)](#)

NGINX SSL proxy for Kibana (RPM-based distributions)

1. First, install NGINX:

a. For CentOS:


```
$ cat > /etc/yum.repos.d/nginx.repo <<\EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=0
enabled=1
EOF

$ yum install nginx
```

a. For RHEL:

```
$ cat > /etc/yum.repos.d/nginx.repo <<\EOF
[nginx]
name=nginx repo
baseurl=http://nginx.org/packages/rhel/$releasever/$basearch/
gpgcheck=0
enabled=1
EOF

$ yum install nginx
```

 **Note**

For more information, see [NGINX: Official Red Hat/CentOS packages](#).

2. Install your SSL certificate and private key:

a. If you have a valid **signed certificate**, copy your key file `<ssl_key>` and your certificate file `<ssl_pem>` to their proper locations:

```
$ mkdir -p /etc/pki/tls/certs /etc/pki/tls/private
$ cp <ssl_pem> /etc/pki/tls/certs/kibana-access.pem
$ cp <ssl_key> /etc/pki/tls/private/kibana-access.key
```

b. Otherwise, create a **self-signed certificate**. Remember to set the `Common Name` field to your server name. For instance, if your server is `example.com`, you would do the following:

```
$ mkdir -p /etc/pki/tls/certs /etc/pki/tls/private
$ openssl req -x509 -batch -nodes -days 365 -newkey rsa:2048 -keyout /etc/pki/tls/private/kibana-access.key -out /etc/pki/tls/certs/kibana-access.pem
```

3. Configure NGINX as an HTTPS reverse proxy to Kibana:

```
$ cat > /etc/nginx/conf.d/default.conf <<\EOF
server {
    listen 80;
    listen [::]:80;
    return 301 https://$host$request_uri;
}

server {
    listen 443 default_server;
    listen [::]:443;
    ssl on;
    ssl_certificate /etc/pki/tls/certs/kibana-access.pem;
    ssl_certificate_key /etc/pki/tls/private/kibana-access.key;
    access_log /var/log/nginx/nginx.access.log;
    error_log /var/log/nginx/nginx.error.log;
    location / {
        auth_basic "Restricted";
        auth_basic_user_file /etc/nginx/conf.d/kibana.htpasswd;
        proxy_pass http://localhost:5601/;
    }
}
EOF
```

! Note

We configure nginx in order to encapsulate the IP address of the Kibana server. This configuration allows us to redirect Kibana requests to HTTPS, when you use this configuration it's recommended to edit the file `/etc/kibana/kibana.yml` and set the field `server.host` to `localhost`, then you must restart the Kibana service to apply this change.

4. Allow NGINX to connect to Kibana port if you're using SELinux:

```
$ semanage port -a -t http_port_t -p tcp 5601
```

! Note

We assume that you have `policycoreutils-python` installed to manage SELinux.

Enable authentication by htpasswd

1. Install the package `httpd-tools`:

```
$ yum install httpd-tools
```

2. Generate the `.htpasswd` file. Replace `wazuh` with your chosen username (it must match with *auth_basic_user_file*):

```
$ htpasswd -c /etc/nginx/conf.d/kibana.htpasswd wazuh
```

3. Restart NGINX:

a. For Systemd:

```
$ systemctl restart nginx
```

b. For SysV Init:

```
$ service nginx restart
```

Now try to access the Kibana web interface via HTTPS. It should prompt you for the username and password that you just created.

NGINX SSL proxy for Kibana (Debian-based distributions)

1. Install NGINX:

```
$ apt-get install nginx
```

2. Install your SSL certificate and private key:

a. If you have a valid signed certificate, copy your key file `<ssl_key>` and your certificate file `<ssl_pem>` to their proper locations:

```
$ mkdir -p /etc/ssl/certs /etc/ssl/private
$ cp <ssl_pem> /etc/ssl/certs/kibana-access.pem
$ cp <ssl_key> /etc/ssl/private/kibana-access.key
```

b. Otherwise, create a **self-signed certificate**. Remember to set the `Common Name` field to your server name. For instance, if your server is `example.com`, you would do the following:

```
$ mkdir -p /etc/ssl/certs /etc/ssl/private
$ openssl req -x509 -batch -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/kibana-access.key -
out /etc/ssl/certs/kibana-access.pem
```

3. Configure NGINX as an HTTPS reverse proxy to Kibana:

```
$ cat > /etc/nginx/sites-available/default <<\EOF
server {
    listen 80;
    listen [::]:80;
    return 301 https://$host$request_uri;
}

server {
    listen 443 default_server;
    listen [::]:443;
    ssl on;
    ssl_certificate /etc/ssl/certs/kibana-access.pem;
    ssl_certificate_key /etc/ssl/private/kibana-access.key;
    access_log /var/log/nginx/nginx.access.log;
    error_log /var/log/nginx/nginx.error.log;
    location / {
        auth_basic "Restricted";
        auth_basic_user_file /etc/nginx/conf.d/kibana.htpasswd;
        proxy_pass http://localhost:5601/;
    }
}
EOF
```

Note

We configure nginx in order to encapsulate the IP address of the Kibana server. This configuration allows us to redirect Kibana requests to HTTPS, when you use this configuration it's recommended to edit the file `/etc/kibana/kibana.yml` and set the field `server.host` to `localhost`, then you must restart the Kibana service to apply this change.

Enable authentication by htpasswd

1. Install the package `apache2-utils`:

```
$ apt-get install apache2-utils
```

2. Generate the `.htpasswd` file. Replace `<user>` with your chosen username:

```
$ htpasswd -c /etc/nginx/conf.d/kibana.htpasswd <user>
```

3. Restart NGINX:

- a. For Systemd:

```
$ systemctl restart nginx
```

- b. For SysV Init:

```
$ service nginx restart
```

Now try to access the Kibana web interface via HTTPS. It should prompt you for the username and password that you just created.