


So you want to do some logging. . . (PT. 2 Adding HTTPS)

 blog.iso365down.com/so-you-want-to-do-some-logging-pt-2-adding-https-75f5f78d7c4e

HanSolo71

December 4, 2023

HanSolo71



Best to not just give away our secrets

Building Reverse Proxy

Because Graylog by default uses HTTP and it is important we put the Graylog web and API interfaces behind a HTTPS proxy. For this blog we will be using NGINX as our reverse proxy.

Configuring Graylog

Before installing NGINX we will want to modify our Graylog config file located at `/etc/graylog/server/server.conf`. Find the line `http_bind_address = $ServerIP` and change it to the following config.

```
= .:
```

This now restart the Graylog service.

```
sudo systemctl restart graylog-server
```

Validate Graylog is listening on 127.0.0.1:9000

```
graylog@graylog:~$ sudo netstat -tulpn | grep LISTENTcp
0 0 0 0
127.0.0.1:27017 0.0.0.0:* LISTEN 584/mongodtcp 0
0 0.0.0.0:22 0.0.0.0:* LISTEN 600/sshd:
/usr/sbintcp6 0 0 127.0.0.1:9000 :::*
LISTEN 1299/javatcp6 0 0 :::22 :::*
LISTEN 600/sshd: /usr/sbintcp6 0 0 :::9200 :::*
LISTEN 585/javatcp6 0 0 :::9300 :::*
LISTEN 585/java
```

```
graylog@graylog:~$ curl "http://127.0.0.1:9000"
```

```
<!DOCTYPE >
```

```
<>
```

```
<>
```

```
< = =>
```

```
< = =>
```

```
< =>
```

```
<>Graylog Web Interface</>
```

```
< = =>
```

```
</>
```

```
<>
```

```
< = />
```

```
< = =></>
```

```
< = =></>
```

```
< = =></>
```

```
< = =></>
```

```
< = =></>
```

```
< = =></>
```

Now that Graylog is listening on localhost:9000 again we can install NGINX and move forward with setting up HTTPS.

Installing NGINX

Start by installing NGINX. For Debian 12 this is done with the following command.

```
sudo apt install nginxsudo systemctl daemon-reloadsudo systemctl nginxsudo
systemctl start nginxsudo apt install certbotsudo apt install python3-certbot-
nginx
```

This will do the following actions:

1. Install NGINX
2. Reload systemd configuration manager taking changed configurations from filesystem and regenerating dependency trees
3. Enable the NGINX service so NGINX starts on system boot
4. Start the NGINX service
5. Install the Let's Encrypt certbot
6. Install the Let's Encrypt > NGINX python script library

Configuration NGINX

Now that NGINX is installed and started we will need to configure NGINX to be a reverse proxy for Graylog.

Create a file named *graylog-domain-com.conf* using your favorite text editor under */etc/nginx/conf.d* with the following content.

```
server
{
listen80 default_server;
listen [::]:80 default_server ipv6only=on;
    server_name graylog.domain.com;

        location / {            proxy_set_header Host $http_host;
proxy_set_header X-Forwarded-Host $host;            proxy_set_header X-Forwarded-Server
$host;            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Graylog-Server-URL http:$server_name/;            proxy_pass
http:...;    }}
```

Configuring Certbot



When I hit the certbot request limit after screwing up my command a bunch of times

Before working through this section you will need to ensure the fully qualified domain name (FQDN) you have selected for your Graylog server has a valid **PUBLIC DNS RECORD**. If you do not have a valid public DNS record for your Graylog server this step will fail.

You will also need to be able to create DNS records for this domain as Certbot will require you to create a TXT record for your domain.

If you don't have a public DNS Service you use, now is a great time to buy a domain on namecheap.com and learn some Cloudflare using their free tier.

```
sudo certbot -nginx --preferred-challenges dns -d
```

Follow the instructions given, creating the needed DNS records and validating certbot is changing the NGINX configuration file you expect.

```
graylog@graylog:/etc/letsencrypt$ sudo certbot --nginx --preferred-challenges dns
-d "*.internal.iso365down.com"
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Certificate not yet due for renewal
```

You have an existing certificate that has exactly the same domains or certificate name you requested and isn't close to expiry.
(ref: /etc/letsencrypt/renewal/internal.iso365down.com.conf)

What would you like to do?

- ```
- - - - -
1: Attempt to reinstall this existing certificate
2: Renew & replace the certificate (may be subject to CA rate limits)
- - - - -
```

Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1  
Deploying certificate

Which server blocks would you like to modify?

```
- - - - -
1: File: /etc/nginx/conf.d/graylog-internal-iso365down-com.conf
Addresses: [::]:80 default_server, 80 default_server
Names: graylog.internal.iso365down.com
HTTPS: No
- - - - -
```

Select the appropriate numbers separated by commas and/or spaces, or leave input blank to select all options shown (Enter 'c' to cancel):  
Successfully deployed certificate for \*.internal.iso365down.com to  
/etc/nginx/conf.d/graylog-internal-iso365down-com.conf

Which server blocks would you like to modify?

```
- - - - -
1: File: /etc/nginx/conf.d/graylog-internal-iso365down-com.conf
Addresses: [::]:80 default_server, 443 ssl, [::]:443 ssl, 80 default_server
Names: graylog.internal.iso365down.com
HTTPS: Yes
- - - - -
```

Select the appropriate numbers separated by commas and/or spaces, or leave input blank to select all options shown (Enter 'c' to cancel):  
Congratulations! You have successfully enabled HTTPS on  
https://\*.internal.iso365down.com

```
- - - - - -If
you like Certbot, please consider supporting our work by: * Donating to ISRG /
Let's Encrypt: https://letsencrypt.org/donate * Donating to EFF:
https://eff.org/donate-le- - - - -
- - - - -graylog@graylog:/etc/letsencrypt$
```

If we *cat* our NGINX config file we will see the following sections added.

```

 listen [::]:443 ssl ipv6only=on; # managed by Certbot
 listen 443 ssl; # managed by Certbot
 ssl_certificate /etc/letsencrypt/live/internal.iso365down.com/fullchain.pem; #
managed by Certbot
 ssl_certificate_key /etc/letsencrypt/live/internal.iso365down.com/privkey.pem;
managed by Certbot
 include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
 ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

 }
server
{
 if ($host ~ ^[^.]+\.\.internal\.\.iso365down\.com$) {
 return 301 https://$host$request_uri;
 } # managed by Certbot

 listen 80 default_server; listen [::]:80 default_server
ipv6only=on; server_name graylog.internal.iso365down.com; 404;

```

Validate you can now reach Graylog via HTTPS

```

graylog@graylog:/etc/nginx/conf.d$ curl "https://graylog.internal.iso365down.com"
<!DOCTYPE >
<>
<>
< = =>
< = =>
< =>
<>Graylog Web Interface</>
< = =>

</>
<>
< = />
< = =></>

< = =></>

< = =></>

< = =></>

< = =></>

< = =></>

```

## Setting up Certbot to Auto-renew our Certificates

---

Nothing is worse than a late night alert that a service went down because a SSL cert expired. With Certbot you can automate this task to ensure its never forgotten again.

Start by editing the systems crontab with the following command.

```
sudo crontab -e
```

Add the following line to the bottom of your crontab to run Certbot every day at noon. If the certificate in use by Graylog expires in the next 30 days, Certbot will automatically renew the certificate

```
* * * bin/certbot renew --quiet
```

## Cleaning Up

---

Double check you can still login to the GUI located at <https://graylog.domain.com> now.

## Whats Next?

---

For the last part on setting up, we will be connecting our Graylog server to Active Directory to enable LDAP authentication.