How to enable HTTPS

WAS THIS PAGE HELPFUL? Leave Feedback

CONTENTS

Overview

GroundWork Monitor supports the use of HTTPS using TLS for encrypting web browser connections to Apache, although this feature is not enabled by default. The binaries and libraries necessary to enable HTTPS support are included in the GroundWork Monitor distribution. Also see sections for GDMA Notes.

Scripted HTTPS Support in GroundWork Monitor

To setup HTTPS in GroundWork you will need to run the **setup-https.py** script found in **/usr/local/groundwork/tools/system_setup/**.

> ⚠ You will need to be **root** or run via **sudo** given the nature of some of the tasks performed.

Basic operation

The script will set the system up with a certificate corresponding to the current FQDN of the system. Because daemons are restarted during this process it can take some minutes to complete. If you are planning on doing self signed certificates on a standalone GroundWork system then you can run this script without any flags:

```
# cd /usr/local/groundwork/tools/system_setup/
# ./setup-https.py
GroundWork Monitor Enterprise is now configured for https.
```

Examples

- To not setup a port redirect from 80 to 443 you can use the **--nodirect** flag:

  ```
  ./setup-https.py  --noredirect
  ```

- To use your own provided certificate:

  ```
  /setup-https.py  --certfile path/to/my/cert.pem --certkey path/to/my/cert.key --certca
  path/to/my/certca.pem
  ```

Backup file

A backup of the configuration files modified by the **setup-https.py** script created at **/usr/local/groundwork/backup/DATESTAMP-pre-https-config-backup.tgz**. For example:

```
$ ls -l /usr/local/groundwork/backup/
total 32
-rw-r--r-- 1 root root 29695 Oct 16 11:44 2017-10-16-1144-pre-https-config-backup.tgz
```

Log file

A log of the operation run by the script can be found in the same directory at **/usr/local/groundwork/tools/system_setup/log/setup-https.log**. Refer to this log file if something didn't work as expected. If you need to open a support ticket, attach this file to the ticket.

Reverting to previous configuration

In the event you need to roll back the changes done by this script you can run the **grafbridge-control** script, restore the backup file, and then

restart GroundWork again:

1. Run the **grafbridge-control** script to disable https in the respective config locations:

```
/usr/local/groundwork/grafana/scripts/grafbridge-control -ssl disable
```

Because some APIs are communicated within this step, it is required GroundWork be up and available at the URL configured the previous time it was run (e.g., by the **setup-https.py** script). You MUST have restarted GroundWork prior to running this step again or it will fail. If a restart of GroundWork has occurred but this step produces errors you will need to run the following command to correct it:

```
/usr/local/groundwork/tools/system_setup/scripts/update-graf-ds.py --protocol http
```

2. Restore the backup file found at **/usr/local/groundwork/backup/DATESTAMP-pre-https-config-backup.tgz**.

```
tar xvf /usr/local/groundwork/backup/2017-10-16-1144-pre-https-config-backup.tgz -C /
```

3. Restart GroundWork daemons:

```
/etc/init.d/groundwork restart
```

Usage

```
$ ./setup-https.py --help
usage: setup-https.py [-h] [--create_certs] [--redirect] [--noredirect]
                      [--certfile CERTFILE] [--certkey CERTKEY]
                      [--certca CERTCA] [--servername SERVERNAME]
                      [--josso_servername JOSSO_SERVERNAME]
                      [--java_keystore_pass JAVA_KEYSTORE_PASS]
                      [--extra_vars_file EXTRA_VARS_FILE] [--save] [--print]
                      [--purge_extra_vars] [--info] [--debug]

Tool to drive automated setup of https for GroundWork Monitor Enterprise.
Settings taken from extra-vars.yml if present. Flags take precedence.

optional arguments:
  -h, --help            show this help message and exit
  --create_certs        generate self signed certificates, (default)
  --redirect            listen on port 80 to redirect to port 443. If neither
                        --redirect nor --noredirect is specified. --redirect
                        is assumed.
  --noredirect          do not listen on port 80 to redirect to port 443
  --certfile CERTFILE   path to user supplied certificate
  --certkey CERTKEY     path to user supplied key
  --certca CERTCA       path to user supplied ca certificate
  --servername SERVERNAME
                        servername if different than discovered FQDN
  --josso_servername JOSSO_SERVERNAME
                        servername for josso auth if different than
                        localhost:8888
  --java_keystore_pass JAVA_KEYSTORE_PASS
                        keystore password if different than default, (default:
                        changeit)
  --extra_vars_file EXTRA_VARS_FILE
                        path to extra-vars.yml file if different than default
  --save                Only update the extra-vars.yml file and exit
  --print               print the content of the extra-vars.yml file and exit
  --purge_extra_vars    delete extra-vars.yml file and exit
  --info                set log level to INFO
  --debug               set log level to DEBUG
```

Regenerating certificates or generating new certificates for child servers

Because of recent changes to browsers, such as FireFox and Chrome, certificates are now required to have subject alternative name (SAN) fields. Self signed certificates created with instructions from previous versions of GroundWork will work with 7.2 but will not validate in these browsers because they lack these fields. The OpenSSL cli tool does not prompt for this field to be added so we have wrapped it in a shell tool.

- If you need to regenerate certs or create certs for child servers it is advised to use this tool on the parent so that a single self signed CA is all that needs to be distributed.
- **make_cert.sh** when run without arguments will generate a new CA certificate and a new certificate for the current short hostname signed by that CA. (If they exist it will not overwrite them without the **-f** flag to *force*.) Because certificates are used to convey the identity of the host, it is good practice to reduce ambiguity and use the FQDN of the host that will be used in the browser.

- To create a certificate for the child server you can do this with the **-h** flag:

```
$ ./make_cert.sh -h child.example.com
Generating RSA private key, 4096 bit long modulus
.........++
.....................................................................................................
is 65537 (0x10001)
Signature ok
subject=/C=US/ST=CA/L=San Francisco/OU=IT/CN=child.example.com
Getting CA Private Key
Doing /usr/local/groundwork/common/openssl/certs
groundwork-ca.pem => c0e51f74.0
parent.example.com.pem => 42e41d7a.0
child.example.com.pem => b7536c02.0
```

  The certificates will be created in **/usr/local/groundwork/common/openssl/certs**:

```
$ ls -l/usr/local/groundwork/common/openssl/certs
total 16
lrwxrwxrwx 1 root root   22 Oct 16 17:54 42e41d7a.0 -> parent.example.com.pem
lrwxrwxrwx 1 root root   21 Oct 16 17:54 b7536c02.0 -> child.example.com.pem
lrwxrwxrwx 1 root root   17 Oct 16 17:54 c0e51f74.0 -> groundwork-ca.pem
-rw-r--r-- 1 root root 1911 Oct 16 17:45 child.example.com.pem
-rw-r--r-- 1 root root 1968 Oct 16 17:47 groundwork-ca.pem
-rw-r--r-- 1 root root   17 Oct 16 17:47 groundwork-ca.srl
-rw-r--r-- 1 root root 1988 Oct 16 11:45 parent.example.com.pem
```

  and key will be created in **/usr/local/groundwork/common/openssl/private**:

```
$ ls -l
total 12
-rw-r--r-- 1 root root 3243 Oct 16 17:44 child.example.com.key
-rw-r--r-- 1 root root 3243 Oct 16 11:45 groundwork-ca.key
-rw-r--r-- 1 root root 3243 Oct 16 11:45 parent.example.com.key
```

- This is the same tool used by **setup-https.py** to create certificates.

Notes

- The **grafbridge-control** script requires restarts of GroundWork between changes to the API endpoints. This includes changes to/from HTTPS.
- **make_cert.sh** is used to generate self signed certs. Should be reused on the parent for child server certs.
- Certificates need to have SANs in order to validate in modern browsers.
- Look in the log file if things go bad.

GDMA Notes

- **GDMA Plugins**
  When using HTTPS and downloading GDMA plugins, if you provided your own certificates the hostname used to access the system must exactly match what is in the server's SSL certificate. For more information regarding downloading new GDMA plugins see the Bookshelf document GDMA Advanced. Also see GDMA with HTTPS.
- **GDMA Version**

If you use an old version of GDMA and need to transition to the current version the **--noredirect** flag will setup an incompatible configuration as they leverage the port 80 to 443 redirect in their discovery. You can change this by rerunning the **setup-https.py** script again but passing the **--redirect** flag to override the stored settings from the previous run. Once you have migrated to current versions of the agent you can rerun again with the **--noredirect** flag should you wish to only listen on 443.