Let's Encrypt Setup for FileWave Server (Debian)

What

This Knowledge Base (KB) article discusses the use of a shell script to automate the process of setting up Let's Encrypt SSL certificates for a FileWave server. Let's Encrypt is a free, automated, and open Certificate Authority (CA) that provides SSL certificates required for secure (HTTPS) connections.

When/Why

FileWave administrators would need to set up an SSL certificate for the FileWave server to ensure secure communication. Using Let's Encrypt is a popular choice because it's free and can be automated. However, the setup process has multiple steps and can be prone to human error. This script simplifies the process by automating most of the steps.

> Note that this documented process is for Debian systems. It could be adapted for macOS.

How

The script needs to be run on a FileWave server with root permissions. The server should be Debian 12+ or compatible.

Here are the steps to use the script:

1. Download the script below.
2. Make the script executable by running `chmod +x scriptname.sh`.
3. Run the script as root with --install as the argument.
   Example: `sudo ./scriptname.sh --install`
4. The script will automate the rest of the process. It will:
   - Check if the provided FQDN is resolvable.
   - Backup any existing certificates.
   - Install necessary tools (if not already installed).
   - Request a new certificate from Let's Encrypt.
   - Set up a script to automatically renew the certificate and apply it to the FileWave server.
   - Set up a daily cron job to renew the certificate.
5. Check the output of the script to ensure there were no errors during the process.
6. If you want to remove Let's Encrypt you can do the following and then go in to FileWave Central and replace your certificate with one from a different source.
   Example: `sudo ./scriptname.sh --uninstall`

> Please replace `scriptname.sh` with the actual name of the script.

Download Script: linux-letsencrypt-v2.0.sh

▼ linux-letsencrypt-v2.0.sh (Debian)

```
#!/bin/bash
#26-May-2022 - Removed the old update dep certs file that would break a 14.7.x server updating its dep
profiles
#19-June-2020 - added parameter verification ; made DEP script injection conditional ; added firewall
exceptions ; made sure certificate is injected at initial run ; added cronjob scheduling
#10-July-2023 - Updated the script to replace the old renewal hook if it exists, made the script work with --
uninstall without additional arguments, changed the script to ask for the hostname and email during
installation, added Debian OS check.
#16-Feb-2024 - Updated for Debian server
#08-Mar-2024 - Refinements for Debian use. Switched to SNAP for certbot and more help if errors.
#23-Apr-2024 - Added code to set the cert as trusted

scriptname="$BASH_SOURCE"

# check if script is running on Debian
if ! grep -q 'Debian' /etc/os-release ; then
echo "This script is designed to run on Debian. Exiting."
exit 1
fi
```

```
# root or moot
if [ $(whoami) != "root" ] ; then
echo "root rights required - please rerun as"
echo "sudo ${BASH_SOURCE} --install"
exit 1
fi

#letsencrypt installation for Debian Linux
if [ -z "$1" ]; then
    echo "This script automates the process of obtaining and installing a Let's Encrypt SSL certificate for a
FileWave server on Debian."
    echo "Available arguments:"
    echo "--install: Install a new Let's Encrypt SSL certificate."
    echo "--uninstall: Uninstall the Let's Encrypt SSL certificate and remove associated scripts."
    exit 1
fi

# Check if uninstall switch is provided
if [ "$1" == '--uninstall' ]; then
  echo "Uninstalling..."
  rm -f /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh
  rm -f /etc/cron.daily/letsencrypt-filewave
  snap remove certbot
  echo "Uninstallation complete. The renewal hook script, daily cron job, and Certbot have been removed."
  exit 0
fi

# Proceed with installation
if [ "$1" == '--install' ]; then
  read -p "Enter your fully qualified domain name (hostname): " hostname
  read -p "Enter your email address: " emailaddress

  # verify whether this is a resolvable public hostname ; abort if not
  if ! [ -x "$(command -v nslookup)" ]; then
    echo "Installing 'dnsutils' for Debian."
    apt install -y dnsutils
  fi

  nslookup $hostname 8.8.8.8
  public=$?
  if [ $public -ne 0 ] ; then echo "$hostname not resolvable by 8.8.8.8 (Google) ; exiting." ; exit 0 ; fi

  # Confirm hostname and email address
  echo "Hostname: $hostname"
  echo "Email Address: $emailaddress"
  read -p "Are these correct? (yes/no) " confirm
  if [ "$confirm" != "yes" ]; then
    echo "Aborting due to user confirmation."
    exit 1
  fi

  #Backup existing certs
  BACKUP_DATE=$(date +%Y-%m-%d-%H-%M)
  mkdir /usr/local/filewave/certs/backup-$BACKUP_DATE
  cp -p /usr/local/filewave/certs/server.* /usr/local/filewave/certs/backup-$BACKUP_DATE

  # Update Debian system and Install Certbot if not already installed
  apt update -y
  DEBIAN_FRONTEND=noninteractive apt-get upgrade -y -o Dpkg::Options::="--force-confold"

  if ! [ -x "$(command -v certbot)" ]; then
    echo "Installing SNAP."
        apt install -y snapd
        snap install core
        echo "Installing Certbot."
        apt remove certbot python3-certbot-apache
        snap install --classic certbot
        ln -s /snap/bin/certbot /usr/bin/certbot
    # Test again for the certbot command
```

```
    if ! [ -x "$(command -v certbot)" ]; then
      echo "Certbot installation failed. Exiting."
      exit 1
    fi
  fi

  #request certificate initially
  # I've seen this fail and there is no harm in the command running 2x the first time to be sure it worked.
  certbot -n --agree-tos --standalone certonly -d "$hostname" -m "$emailaddress"

  echo "Certificate for $hostname should be created. If there was an error try running:"
  echo 'certbot -n --agree-tos --standalone certonly -d "'$hostname'" -m "'$emailaddress'"'

# PostgreSQL command to update ios_preferences
  echo "Updating iOS preferences in PostgreSQL..."
  /usr/local/filewave/postgresql/bin/psql -d mdm -U django -c "INSERT INTO ios_preferences (key, value) VALUES
('mdm_cert_trusted', TRUE) ON CONFLICT (key) DO NOTHING; UPDATE ios_preferences SET value = 'true' WHERE key =
'mdm_cert_trusted';"
  if [ $? -eq 0 ]; then
      echo "iOS preferences updated successfully."
  else
      echo "Failed to update iOS preferences in the database."
  fi

  # Ensure renewal-hooks and deploy directories exist
  mkdir -p /etc/letsencrypt/renewal-hooks/deploy

  # Remove old renewal hook if exists
  rm -f /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh

  # Create new renewal hook
  cat>/etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh<<EOT
#!/bin/bash
cp /etc/letsencrypt/live/$hostname/fullchain.pem /usr/local/filewave/certs/server.crt
cp /etc/letsencrypt/live/$hostname/privkey.pem /usr/local/filewave/certs/server.key
chown apache:apache /usr/local/filewave/certs/server.*
echo "Restarting FileWave Server..."
/usr/local/bin/fwcontrol server restart
echo "Updating DEP profile certificates..."
(yes 2>/dev/null) | /usr/local/filewave/python/bin/python /usr/local/filewave/django/manage.pyc
update_dep_profile_certs 2>/dev/null
echo "DEP profile certificates updated."
EOT
  chmod a+x /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh

  echo "injecting certificate into filewave server"
  /etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh
  echo "injection done"

##Install firewalld for configuring port 80
#    if ! [ -x "$(command -v firewalld)" ]; then
#    echo "Installing firewalld."
#    apt install -y firewalld
#    systemctl enable --now firewalld
#    # Test again for the firewalld command
#    if ! [ -x "$(command -v firewalld)" ]; then
#      echo "firewalld installation failed. Exiting."
#      exit 1
#    fi
#  fi

#  echo "making firewall opening for port 80 permanent to allow for automatic renewals"
#  firewall-cmd --add-port 80/tcp --zone=public --permanent
#  firewall-cmd --reload
#  echo "firewall settings modified"

  echo "scheduling automatic renewal of certificate"
  cat>/etc/cron.daily/letsencrypt-filewave<<'EOT'
#!/bin/bash
sleep $[RANDOM % 7200]
```

```
/usr/bin/certbot renew --quiet
EOT
  chmod a+x /etc/cron.daily/letsencrypt-filewave
  echo "scheduling complete"

  echo " "
  echo "-----------------"
  echo "List of files created by this script:"
  echo "/etc/letsencrypt/renewal-hooks/deploy/filewave-server-cert.sh"
  echo "/etc/cron.daily/letsencrypt-filewave"
  echo "Certificate for $hostname should be created. If there was an error try running:"
  echo 'sudo certbot -n --agree-tos --standalone certonly -d "'$hostname'" -m "'$emailaddress'"'
  echo 'And upon success run:'
  echo "sudo certbot renew --force-renewal"
  echo "The main reason this process can fail is if the server is not reachable on TCP 80"
else
  echo "Invalid argument. Please run with --install or --uninstall."
fi
```

Troubleshooting

> Please note that using Let's Encrypt requires your server to be reachable from the internet on port 80, as Let's Encrypt uses the HTTP-01 challenge to verify your server's identity.

If you see errors when it tries to do the below command that registers with Let's Encrypt or if you just see that there is no active cert you can run the command below again. Just replace $hostname with fwjoshlab.filewave.net if that is your server, and $emailaddress with your email. Keep the " marks. It'll tell you if it is successful or already was previously successful.

Once you have success with reistering then running the renew will tell the FileWave Server that the cert is updated.

```
sudo certbot -n --agree-tos --standalone certonly -d "$hostname" -m "$emailaddress"
sudo certbot renew --force-renew
```

If you have noticed that on your https://server:20443 you see 2 options where 1 is to download a certificate then you may have used an old version of this script. The current script handles this. The fix is to do the below in a Terminal session on the server:

```
/usr/local/filewave/postgresql/bin/psql mdm django
insert into ios_preferences values('mdm_cert_trusted', TRUE);
update ios_preferences set value='true' where key='mdm_cert_trusted';
\q
```

An image of what this looks like:

```
admin@ip-172-30-3-220:~$ /usr/local/filewave/postgresql/bin/psql mdm django
psql (12.18)
Type "help" for help.

mdm=# insert into ios_preferences values('mdm_cert_trusted', TRUE);
INSERT 0 1
mdm=# update ios_preferences set value='true' where key='mdm_cert_trusted';

UPDATE 1
mdm=# \q
```

Related Links

- Let's Encrypt Documentation
- GitHub - nycon/filewave-installer: Filewave AIO installer
- Review My Notes: FileWave and Let's Encrypt | Version 12.0 (punkstuff.com)

Revision #7
★Created 8 March 2024 14:38:07 by Josh Levitsky
✎Updated 15 November 2024 14:16:24 by Josh Levitsky