1. Ensure, git a current Python, and PyOpenSSL are installed

```
zypper in git
```

On OpenSUSE 13.2 you will also need (versions of python > 2.7.8 may not require):

```
zypper in python-pyOpenSSL
```

2. In /root, install letsencrypt

```
git clone https://github.com/letsencrypt/letsencrypt
```

3.Configure the site

Create a file `/etc/letsencrypt/<site>.ini` with:

```
    rsa-key-size = 4096
    email = <admin email>
    authenticator = standalone
    webroot-path = <path to web root>
    domain = <hostname to secure>
```

4. Get your certificate

Note: Will require shutting down Apche for this process using standalone

```
cd /root/letsencrypt
systemctl stop apache2.service
./letsencrypt-auto certonly -c /etc/letsencrypt/<site>.ini
```

5. Adjust the server configuration

If step 4 was successful, you'll find the certificates in `/etc/letsencrypt/live/<hostname to secure>/`. Next, adjust the Apache site configuration to use these files:

```
SSLEngine on
SSLCertificateFile /etc/letsencrypt/live/<hostname to secure>/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/<hostname to secure>/privkey.pem
```

> *If this is a new secure site"ServerName" by appending ":443" to IPs resp. domain name (resp. replacing ":80" by ":443").*

A separate ssl log is desirable, adjust the Logfile names (if using "combined" in a CustomLog, substitute with "ssl_combined").

If ssl has not been used before, activate in `/etc/sysconfig/apache2`:

```
APACHE_MODULES="[...] ssl [...]"
```

and

```
APACHE_SERVER_FLAGS="SSL"
```

Now test the apache2 configuration and restart:

```
rcapache2 configtest        apache2ctl configtest
rcapache2 start             apache2ctl start            systemctl start apache2.service
```

6. Renewal of the certificate

If all that was successful, you may wait two months (not more than 89 days) to renew your certifcate by repeating step 4. The apache2 server does not need any new configuration, only a shutdown during the renewal process [standalone].

7. Automatic Renewal

With LetsEncrypt installed, the next good thing is automatic renewal. Here is a sample shell script to run every week:

```
#
#   Cron job to check for expiration of Let's Encrypt Certificate
#
#   Open Source - no copyright
#
#

PTS=$(getopt -o cehw: --long config:,expire-limit:,help,webservice: -n "$0" --
"$@")
if [ $? != 0 ]; then
  echo "Terminating ..." >&2
  exit 1
fi

CONFIG=/etc/letsencrypt/oc.ini
WEBSERVICE=apache2
EXPIRE_LIMIT=14
EXPIRE=
DOMAIN=oc.omnitec.net
CERT_FILE=/etc/letsencrypt/live/oc.omnitec.net/cert.pem
CERT_LIVE_PATH=/etc/letsencrypt/live
VALID_DAY=90

OPENSSL=$(which openssl)

if [ -z "$OPENSSL" ]; then
  echo "OpenSSL is required, please install" >&2
  exit 1
fi

print_help () {
  echo "Usage: $0 [Options]"
  echo "Options:"
  echo " -c, --config <config_file>  Configuration file"
  echo "                             default: /etc/letsencrypt/live/<somesite>.ini"
  echo " -e, --expire-limit <day>    Expire limit in day to perform the renewal"
  echo "                             default: 7"
  echo " -w, --webservice <name>     Web service name"
  echo " -h, --help                  Print this help"
  exit 0
}
print_settings () {
  printf "Start: %s\n" "$(date)"
  printf "Settings ...\n" "$(date)"
  printf " - Config File : %s\n" "$CONFIG"
  printf " - Domain: %s\n" "$DOMAIN"
  printf " - Certificate File: %s\n" "$CERT_FILE"
  printf " - Certificate Valid For: %d %s\n" $VALID_DAY $(test $VALID_DAY -gt 1 &&
echo days || echo day)
  printf " - Web Service : %s\n" "$WEBSERVICE"
  printf " - Expire Limit: %d %s\n" $EXPIRE_LIMIT $(test $EXPIRE_LIMIT -gt 1 &&
echo days || echo day)
}

parse_config () {
  DOMAIN=$(grep "^\s*domains" $CONFIG | sed 's/,/ /g' | sed 's/^\s*domains\s*=\s*\
(.*\)$/\1/')

  if [ -z "$DOMAIN" ]; then
```

```
    echo "No domains specified in $CONFIG" >&2
    exit 1
  fi

  get_certfile
  get_expire
}

get_certfile () {
  for domain in $DOMAIN; do
    if [ -f $CERT_LIVE_PATH/$domain/fullchain.pem ]; then
      CERT_FILE=$CERT_LIVE_PATH/$domain/fullchain.pem
      break
    fi
  done

  if [ -z "$CERT_FILE" ]; then
    echo "No valid certificate files for domain $DOMAIN" >&2
    exit 1
  fi
}

get_expire () {  EXPIRE=$(date -d"$($OPENSSL x509 -in $CERT_FILE -noout -enddate |
cut -d= -f2)" +%s)
  VALID_DAY=$((($EXPIRE - $(date +%s)) / 86400))
}

start () {
  if [ $VALID_DAY -lt $EXPIRE_LIMIT ]; then
    renew
  else
    echo "The certificate for $DOMAIN is up to date"
  fi
}

renew () {
  # First, stop Apache
  systemctl stop ${WEBSERVICE}.service

  # Second, make SURE it's stopped!
  pkill -9 apache2

  # Run the renewal
  /root/letsencrypt/letsencrypt-auto -c $CONFIG -a standalone --agree-tos --renew-
by-default certonly

  # Start Apache
  systemctl start ${WEBSERVICE}.service

  get_expire
  echo "The certificate for $DOMAIN is valid for next $VALID_DAY days"
}

eval set -- "$OPTS"

while true; do
  case "$1" in
    -c | --config)
      CONFIG="$2"; shift 2
      if [ ! -f "$CONFIG" ]; then
        printf "Config file "%s" does not exist ...\n" $CONFIG >&2
        exit 1
      fi

      parse_config
      ;;
    -e | --expire-limit)
      EXPIRE_LIMIT=$2; shift 2
      ;;
    -w | --webservice)
      WEBSERVICE="$2"; shift 2
      ;;
    -h | --help)
      print_help
      ;;
    *)
      break
      ;;
  esac
done

test -z "$CONFIG" && print_help || print_settings

start
exit 0
```