

```
#!/usr/bin/env bash

EMAIL_TO="empfänger@adresse.tld"
EMAIL_FROM="absender@adresse.tld"

#Ablaufdatum in Tagen
EXPIRE=25

LOG_FILE="/var/log/plesk_ssl_check.log"

CONF_PATHS=$(find /etc/apache2/plesk.conf.d/vhosts/
/etc/httpd/conf/plesk.conf.d/vhosts/ 2>/dev/null -name "*.conf")

[ -z "$CONF_PATHS" ] && echo "$(date) - Keine Vhost-Konfigurationen
gefunden.">> "$LOG_FILE" && exit 1

TMPFILE=$(mktemp)
ALERT=0

# Zertifikatdaten sammeln
for conf in $CONF_PATHS; do
    DOMAIN=$(basename "$conf" .conf)
    CERT_PATH=$(grep -i 'SSLCertificateFile' "$conf" | awk '{print $2}' | head
-n1)

    [ -z "$CERT_PATH" ] && continue

    if [ ! -f "$CERT_PATH" ]; then
        echo "$DOMAIN|$CERT_PATH|Datei nicht gefunden!|***|0">> "$TMPFILE"
        ALERT=1
        continue
    fi

    END_DATE_RAW=$(openssl x509 -in "$CERT_PATH" -noout -enddate 2>/dev/null)
    END_DATE=$(echo "$END_DATE_RAW" | cut -d= -f2)

    if [ -n "$END_DATE" ]; then
        EXP_TS=$(date -d "$END_DATE" +%s 2>/dev/null)
        NOW_TS=$(date +%s)
        DIFF_DAYS=$(( (EXP_TS - NOW_TS) / 86400 ))
    else
        END_DATE="Unbekannt"
        EXP_TS=0
        DIFF_DAYS=999
    fi

    MARKER=""
    if [ "$EXP_TS" -eq 0 ] || [ "$DIFF_DAYS" -le $EXPIRE ]; then
        MARKER="***"
        ALERT=1
    fi
fi
```

```
CERT_FILE=$(basename "$CERT_PATH")
echo "$DOMAIN|$CERT_FILE|$END_DATE|$MARKER|$EXP_TS">> "$TMPFILE"
done

TMPFILE_SORTED=$(mktemp)

{
echo "SSL-Zertifikatsreport für $(date)"
echo
printf "%-25s %-25s %-25s %-15s %-5s\n" "Domain" "Zertifikat-Datei"
"Ablaufdatum" "Verbleibend" "Hinweis"
echo "-----"
-----

sort -t'|' -k5,5n "$TMPFILE" | while IFS='|' read -r DOMAIN CERT_PATH
END_DATE MARKER EXP_TS; do

if [ "$EXP_TS" -eq 0 ]; then
DATE_PART="$END_DATE"
AGE_PART=""
else
NOW=$(date +%s)
DIFF_DAYS=$(( (EXP_TS - NOW) / 86400 ))
DATE_PART="$END_DATE"
AGE_PART="in $DIFF_DAYS Tagen"
fi

printf "%-25s %-25s %-25s %-15s %-5s\n" "$DOMAIN" "$CERT_PATH"
"$DATE_PART" "$AGE_PART" "$MARKER"

done

printf "\n\n\n"

}> "$TMPFILE_SORTED"

# Report in Variable lesen
REPORT=$(cat "$TMPFILE_SORTED")

# In Log schreiben
echo -e "$REPORT">> "$LOG_FILE"

# Mail senden, wenn nötig
if [ "$ALERT" -eq 1 ]; then
# HTML-Mail aufbauen
MAIL_HTML="<html><body>"
MAIL_HTML+="<h2>ACHTUNG: Plesk SSL-Zertifikate laufen bald ab!</h2>"
MAIL_HTML+="<table border='1' cellpadding='5' cellspacing='0'
style='border-collapse: collapse;'>"
MAIL_HTML+="<tr><th>Domain</th><th>Zertifikat-
Datei</th><th>Ablaufdatum</th><th>Restzeit</th></tr>"
```

```
# Zeilen aus temporärer Datei auslesen
while IFS='|' read -r DOMAIN CERT_PATH END_DATE MARKER EXP_TS; do
# Wenn Marker *** (bald ablaufend), dann rot färben
if [ "$MARKER" = "***" ]; then
    ROW_COLOR=" style='background-color:#ffcccc;'" # Hellrot
else
    ROW_COLOR=""
fi

if [ "$EXP_TS" -eq 0 ]; then
    AGE_PART=""
else
    NOW_TS=$(date +%s)
    DIFF_DAYS=$(( (EXP_TS - NOW_TS) / 86400 ))
    AGE_PART="$DIFF_DAYS Tage"

fi

MAIL_HTML+="<tr$ROW_COLOR><td>$DOMAIN</td><td>$CERT_PATH</td><td>$END_DATE</td><td>$AGE_PART</td></tr>"
done <"$TMPFILE"

MAIL_HTML+="</table></body></html>"

/usr/sbin/sendmail -f $EMAIL_FROM $EMAIL_TO <<EOF
Subject: ACHTUNG: Plesk SSL-Zertifikate laufen bald ab!
From: $EMAIL_FROM
To: $EMAIL_TO
MIME-Version: 1.0
Content-Type: text/html; charset="UTF-8"
Content-Transfer-Encoding: 8bit

$MAIL_HTML
EOF
fi

# Aufräumen
rm -f "$TMPFILE" "$TMPFILE_SORTED"
```

Logrotate einrichten:

```
nano /etc/logrotate.d/plesk_ssl_check
```

```
/var/log/plesk_ssl_check.log {
    monthly
    rotate 4
    compress
    missingok
    notifempty
    create 640 root adm
```

```
su root root
postrotate
endscript
}
```

Logrotate testen:

```
logrotate --force /etc/logrotate.d/plesk_ssl_check
```

From:

<https://wiki.mahlen.eu/> - **Smart-Home Wiki**

Permanent link:

[https://wiki.mahlen.eu/doku.php?id=linux:plesk\\_server](https://wiki.mahlen.eu/doku.php?id=linux:plesk_server)

Last update: **24.07.2025**

