# Huiswerk Linux: Sendmail AUTH via SASL

We hebben in de vorige opdracht de *saslauth* daemon geïnstalleerd om te kunnen communiceren met een Windows *Domain Controller*. Sendmail, Postfix, Cyrus IMAP zijn voorbeelden van programma's die de *saslauth* daemon gebruiken. Als je Authentificatie voor Sendmail activeert gebruikt Sendmail, net als Apache, een eigen wachtwoordbestand.

Dit willen we niet. We willen dat Sendmail de *Domain Controller* raadpleegt voor het controleren van het wachtwoord. De opdracht voor deze week is: configureer Sendmail zodanig dat de *saslauth* daemon gebruikt wordt voor wachtwoordcontrole op de Domain Controller.

## Over de Sendmail AUTH mechanismen

Sendmail beschikt over verschillende methodes om e-mail te versturen. Dit worden ookwel *capabilities* genoemd. Door de jaren heen zijn er meerdere aan het SMTP protocol toegevoegd. Eén van die methodes noemen we de *AUTH* (authenticatie) capability. Deze kent meerdere zogenaamde *mechanismen*:

| **Mechanisme** | **omschrijving** |
|---|---|
| ANONYMOUS | Mail wordt verstuurd zonder wachtwoordverificatie (standaard) |
| PLAIN | Het wachtwoord wordt in platte (leesbare) tekst naar de server gestuurd |
| LOGIN | Het wachtwoord wordt versleuteld met het MD5 algoritme |
| DIGEST-MD5 | 128 bytes MD5 wachtwoordversleuteling |
| CRAM-MD5 | Challenge Response Authentication met MD5 versleuteling |
| SCRAM-SHA-1 | Salted Challenge Response Authentication met SHA-1 versleuteling |
| GSSAPI | Generic Security Services Application Programming Interface |
| GSS-SPNEGO | Simple and Protected GSSAPI Negotiation Mechanism |
| GS2-IAKERB | |
| GS2-KRB5 | Kerberos network authentication protocol (via een *ticket*) |
| OTP | |

## Het *sudo* mechanisme

Het configureren van Linux doen we namens de *super-user*. We moeten daarom tijdelijk inloggen als Administrator.

### Cygwin gebruikers

Sudo voor Cygwin gebruikers: rechts-klik op het icoon van de Cygwin terminal, en kies voor **Als administrator uitvoeren**. Zorg ervoor dat we kunnen beschikken over de Sendmail mail-server. Eerst vernieuwen we de pakketten-database:

```
apt-get update
```

Naast *sendmail*, installeren we ook het *inetutils* pakket, zodat we straks het *telnet* programma kunnen gebruiken:

```
apt-get upgrade inetutils sendmail
```

Installeer vervolgens de sendmail daemon als Windows service:

```
win-svc install sendmail
```

Zorg ervoor dat de benodigde daemons zijn gestart:

```
service saslauth start
```

### Andere Linux gebruikers

Sudo voor gebruikers van andere Linux-versies (*Ubuntu*, *Lubuntu*, *Android*, *Debian, UberStudent*, etc): start een terminal met de toetsencombinatie **<Ctrl><Alt>-T**. We gebruiken het commando *sudo* om in te loggen met het *su* (become Super User) commando. Daardoor blijven we ingelogd:

```
sudo su
```

Zorg ervoor dat we kunnen beschikken over de Sendmail mail-server. Ook installeren we het *telnet* pakket, zodat we straks het *telnet* programma kunnen gebruiken:

```
apt-get install sendmail telnet
```

# De *saslauth daemon* activeren

Om ervoor te zorgen dat Sendmail onze Domain Controller raadpleegt om het wachtwoord te verifiëren, moeten we aangeven dat Sendmail de in de vorige opdracht geïnstalleerde *saslauthd* gebruikt. Dit doen we in het bestand */etc/sasl2/Sendmail.conf*. Open dit bestand met de *vi* tekst-editor:

```
vi /etc/sasl2/Sendmail.conf
```

Het kan zijn dat je een leeg scherm ziet. Toets **i** (insert) om naar de *INSERT* modus te gaan en geef de volgende regel in of wijzig de instelling:

```
pwcheck_method: saslauthd
```

Toets **<Esc>** om uit de *INSERT* modus te komen en geef dan de commando's *w* (write) en *q* (quit):

```
:wq
```

Het bestand wordt opgeslagen en we zijn terug op de command-line.

# Sendmail configureren

We gaan zometeen de login testen via ons huiswerk-email programma. Dit programma ondersteunt alleen de *PLAIN* en *LOGIN* mechanismen, dus die moeten we eerst activeren. Ga allereerst naar de Sendmail configuratiemap.

### Cygwin gebruikers

De *m4* scripts van Sendmail kun je vinden in de *cf* (config) directory van sendmail. Wissel naar deze directory met het *cd* (change directory) commando:

```
cd /usr/share/sendmail/cf
```

### Andere Linux gebruikers

Gebruikers van andere Linux-versies (*Debian*, *Ubuntu*, *Lubuntu*, *Kubuntu*, *Gentoo*, *Mint*, etc) kunnen de *m4* scripts vinden in de */etc/mail* directory van sendmail. Wissel naar deze directory met het *cd* (change directory) commando:

```
cd /etc/mail
```

### Wijzigen M4 configuratie-script

Open nu het Sendmail *m4* configuratie-script, genaamd *sendmail.mc*. Dit doe je met het *vi* commando:

```
vi sendmail.mc
```

Het *m4* script voor Linux ziet ongeveer alsvolgt uit. Toets (hoofdletter) **G** om naar het einde van het bestand te gaan. Toets dan het **o** (open) commando. We gaan dan naar de *INSERT* modus en de cursor komt op een nieuwe regel te staan. Voeg nu de onderstaande vetgedrukte regel in.

```
define(`SMART_HOST', `smtp.boland.nl')
define(`RELAY_MAILER_ARGS', `TCP $h 443')
define(`confAUTH_MECHANISMS', `PLAIN LOGIN GSSAPI DIGEST-MD5 CRAM-MD5')
MAILER(local)
MAILER(smtp)
```

Toets **<Esc>** om uit de *INSERT* modus te komen en geef de commando's *w* (write) en *q* (quit):

```
:wq
```

Het bestand wordt opgeslagen en we zijn terug op de command-line.

# Configuratiebestand genereren

We hebben het configuratie-script aangepast. Nu kunnen we het eigenlijke configuratie-bestand aanmaken en installeren. Dit doen we met het *make* programma.

### Cygwin gebruikers

Cygwin gebruikers kunnen dit doen met het commando *install-cf* van het *make* programma:

```
make install-cf
```

### Andere Linux gebruikers

Gebruikers van andere Linux-versies (*Debian*, *Ubuntu*, *Lubuntu*, *Kubuntu*, *Gentoo*, *Mint*, etc) gebruiken ook het *make* programma, maar op de volgende manier:

```
make
```

Het configuratiebestand wordt nu aangemaakt en in de map */etc/mail* geplaatst. De output is op ieder Linux systeem verschillend, maar ziet er ongeveer alsvolgt uit:

```
rm -f sendmail.cf
m4 ../m4/cf.m4 sendmail.mc > sendmail.cf || ( rm -f sendmail.cf && exit 1 )
echo "### sendmail.mc ###" >>sendmail.cf
sed -e 's/^/# /' sendmail.mc >>sendmail.cf
chmod 444 sendmail.cf
install -c -m 0444 sendmail.cf /etc/mail/sendmail.cf
install -c -m 0444 submit.cf /etc/mail/submit.cf
```

### Herstarten Sendmail

Het configuratie-bestand (*sendmail.cf)* is aangemaakt en geïnstalleerd in de directory */etc/mail*. Sendmail moet de nieuwe instellingen nog inlezen. Daarvoor moeten we Sendmail herstarten. Dit doen we met het *pkill* (process kill) programma:

```
service sendmail restart
```

Normaal zouden we eerst het *PID* (program id) van de Sendmail daemon moeten opzoeken en dat gebruiken om Sendmail te herstarten. Het programma *pkill* doet dit automatisch en we kunnen meteen het *HUP* (hangup) signaal sturen.

# *Capabilities* controleren

Om te zien of alles goed is geconfigureerd kunnen we een telnet sessie met Sendmail starten en kijken of we nu kunnen beschikken over het *PLAIN* mechanisme. Dit doen we met het *telnet* programma:

```
telnet localhost smtp
```

Middels het argument *localhost* geven we aan dat we verbinding willen maken met onze eigen computer (host). Via het tweede argument geven we aan dat we de *smtp* poort (25) willen gebruiken. De output ziet er alsvolgt uit:

```
Trying 10.0.20.22...
Connected to optiplex.
Escape character is '^]'.
220 optiplex.sassenheim.dmz ESMTP Sendmail 8.14.9/8.14.9; Tue, 2 Jun 2015 19:38:11 +0200
```

We hebben nu een *live* sessie met onze mail-server. De mail-server wacht op onze commando's. We kunnen nu een conversatie opstarten met het *EHLO* (extended hello) commando:

```
ehlo localhost
```

De output moet er ongeveer alsvolgt uitzien:

```
250-optiplex.sassenheim.dmz Hello raspberrypi.sassenheim.dmz [10.0.0.160], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH PLAIN LOGIN GSSAPI DIGEST-MD5 CRAM-MD5
```

```
250-DELIVERBY
250 HELP
```

Sluit de sessie af met het *quit* commando:

```
quit
```

# Het hoera-moment

Zorg dat je Domain Controller on-line en bereikbaar is. We gebruiken het *email* programma om de login te testen. We sturen de tekst "Hallo" via een e-mail aan mij (*daniel@boland.nl*):

```
echo "Hallo" | email -r localhost -p 25 -m PLAIN -u testuser -s "Test" dboland@idcollege.nl
```

**Let goed op**: de gebruiker *testuser* is geldig op mijn eigen *Domain Controller*. Gebruik login-gegevens welke op jouw DC geldig zijn. De *Admninistrator* gebruiker mag/werkt niet. Maak een aparte test-gebruiker aan. Deze test kun je bovendien alleen uitvoeren vanuit school (intra.rocleiden.nl). Indien je thuis werkt, stuur mij een email, dan zet ik je in mijn lijst met *trusted hosts*.

Via de optie *r* (relay) geven we aan dat we onze eigen mail-server, welke luistert op de *localhost* (127.0.0.1), willen gebruiken. Via optie *p* (port) geven we aan dat de email via SMTP poort 25 verzonden moet worden. Via de optie *m* (mechanism) geven we aan dat we het *PLAIN* mechanisme willen gebruiken. Via optie *u* (user) geven we aan dat we de test-gebruiker van onze DC willen gebruiken.

Om te zien of het gelukt is kunnen we in het maillog kijken:

```
tail /var/log/maillog
```

**Let op**: op sommige Linux systemen heet het log-bestand **/var/log/mail.log**. De output moet er ongeveer alsvolgt uitzien:

```
Jun  3 13:17:59 probook sm-mta: PID 1716: AUTH=server, relay=localhost.localdomain [127.0.0.1],
     authid=testuser, mech=PLAIN, bits=0
Jun  3 13:17:59 probook sm-mta: PID 1716: u53BHxg0001716: from=<d.boland@rocleiden.nl>, size=379,
     class=0, nrcpts=1, msgid=<201606031117.u53BHxg0001716@Probook.sassenheim.dmz>, proto=ESMTP,
     daemon=MTA-SMTP, relay=localhost.localdomain [127.0.0.1]
Jun  3 13:18:09 probook sm-mta: PID 4336: STARTTLS=client, relay=smtp.boland.nl.,
     version=TLSv1/SSLv3, verify=FAIL, cipher=DHE-RSA-AES256-SHA, bits=256/256
Jun  3 13:18:10 probook sm-mta: PID 4336: u53BHxg0001716: to=<dboland@idcollege.nl>,
     delay=00:00:11, xdelay=00:00:11, mailer=relay, pri=120379, relay=smtp.boland.nl.
     [85.92.128.191], dsn=2.0.0, stat=Sent (u53C0i6G000299 Message accepted for delivery)
```

## Huiswerk opsturen

Zoals je weet, kun je de opdracht aftekenen door een e-mail met daarin de output van het *history* commando te versturen. Dit doen we met een pijpleiding tussen de commando's *history* en *email*. Zorg dat het onderwerp van de e-mail tussen aanhalingstekens staat:

```
tail /var/log/maillog | email -s "AUTH via SASL" docent@localhost
```

**Let op**: op sommige Linux systemen heet het log-bestand **/var/log/mail.log**. Als je geen foutmelding ziet is de e-mail succesvol verstuurd.

## Administrator uitloggen

We hebben Sendmail zodanig geconfigureerd dat de gebruikersnamen en wachtoorden van een *Domain Controller* worden gebruikt, maar we zijn nog steeds ingelogd als Administrator (*root*).

Geef tot slot **<Ctrl>-D** om de Administrator uit te loggen.

# Problemen oplossen

Het kan zijn dat je de volgende foutmelding krijgt. Dit betekent dat je Domain Controller geen internet heeft:

```
email: FATAL: Smtp error: Timeout(10) while trying to read from SMTP server
```

Sendmail probeert via het DNS van de Domain Controller mijn e-mail adres te controleren. De Domain Controller heeft geen internet en kan dus het domein *boland.nl* niet vinden. Dit kun je oplossen door mijn e-mail adres te vervangen door **root@localhost**. De test-email wordt dan locaal op je eigen laptop bezorgd.

Hetzelfde geldt voor de afzender. Op dit moment staat deze ingesteld op je ROC Leiden e-mail adres. Ook dit adres wordt gecontroleerd via het DNS op de Domain Controller. Dit kun je oplossen middels de *f* (from) optie. Het volledige commando wordt dan:

```
echo "Hallo" | email -r localhost -p 25 -m PLAIN -u testuser -s "Test" -f daniel@localhost root@localhost
```

**Let op**: gebruik in plaats van *daniel* je eigen gebruikersnaam. Als je deze niet weet, zoek hem dan op via het commando *whoami*.