Hamster: EMails über SSL

Dokumenteigenschaften:

Erstellt am:	08.06.2010	Geändert am:	18.08.2018	
	Autor:	Müller	Lizenztyp:	(cc) BY-NC-SA

Beim Abholen von EMails über das POP3-Protokoll und das Versenden der EMails über SMTP gehen die Passwörter in Klartext (also unverschlüsselt) zum entsprechenden Server. Sicherheit bietet SSL.

Hamster (ein kostenfreier POP3,SMTP, IMAP und NNTP-Server) ist Ideal für den heimischen Gebrauch bestimmt. Wer sich jedoch etwas Bastelarbeit in diese Sache investiert wird mit Sicherheit bei der EMail-
Übertragung belohnt. Bereits für Hamster umgewandelte Zertifikate gibt es hier.

Zuerst benötigen wir einiges an Software:

Hamster-Classic	Version: 2.1.0.11	3 MB	
OpenSSL & DLL-Dateien	Version: [aktuelle]	?? kB	
MakeCert.zip		2 kB	makecert.zip

Ich gehe nun davon aus, daß Hamster eingerichtet ist und einen richtige FQDN besitzt.

Bitte verwenden Sie immer die akuellste OpenSSL-Version!
Immer wieder werden neue Sicherheitlücken gefunden - daher ist es wichtig immer auf den aktuellen Stand zu bleiben. Aktuelle Windows-Versionen bei slproweb.com.

Möchte man das OpenSSL-Setup nicht ausführen und einfach nur die OpenSSL-Dateien haben, so sollte man sich den „**Inno Setup Unpacker**“ Link in Form einer RAR-Datei downloaden.
Mit folgenden Befehl entpackt man alle Dateien:

```
innounp.exe -x Win32OpenSSL-1_0_1f.exe
```

1. Hamster als SSL-Server

In einen leeren Verzeichnis werden die Dateien aus den von OpenSSL: „openssl.exe“, „libeay32.dll“, und „ssleay32.dll“ und aus den Archiv „MakeCert.zip“ alle Dateien kopiert.
So muß das Verzeichnis aussehen:

```
[..]                    <DIR>     06.09.2004 17:50 ----
[doc]                   <DIR>     06.09.2004 18:14 ----
cert                cnf       1.234 29.12.2002 21:35 -a--
libeay32            dll   1.179.472 18.03.2004 23:21 -a--
libssl32            dll     254.679 18.03.2004 23:22 -a--
MakeCert            bat         634 17.01.2003 22:38 -a--
openssl             exe   1.414.996 18.03.2004 23:20 -a--
unix2dos            exe       3.072 15.01.2003 22:40 -a--
```

Die Datei „**MakeCert.bat**“ wird nun ausgeführt. Es wird nun ein Zertifikat (selbstsignierend) für Hamster erstellt.

```
Generating a self-signed RSA X.509 certificate …
Loading 'screen' into random state - done
Generating a 1024 bit RSA private key
……++++++
……++++++
writing new private key to stdout
Enter PEM pass phrase:
```

Jetzt muß ein Kennwort eingegeben werden. Also ein nettes Kennwort ausdenken und nicht vergessen. 
Später werden wir es in Hamster noch benötigen.

```
Verifying - Enter PEM pass phrase:
```

Nochmals wird das Kennwort abgefragt. Jetzt werden enige Informationen abgefragt.

```
——
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Die Abfragen:

```
[]:
```

Eingabe: <ENTER>

```
Common Name (eg, FQDN of your server) []:
```

Eingabe: Die FQDN von Hamster. z.B. mail.example.com

```
Email Address []:
```

Eingabe: Der Verantwortliche für diesen Server. z.B. postmaster@example.com

```
Organizational Unit Name (eg, section) []:
```

Eingabe: Organisation z.B. Privat

```
Locality Name (eg, city) []:
```

Eingabe: Der Wohnort …

```
State or Province Name (full name) []:
```

Eingabe: z.B. Name des Bundeslandes

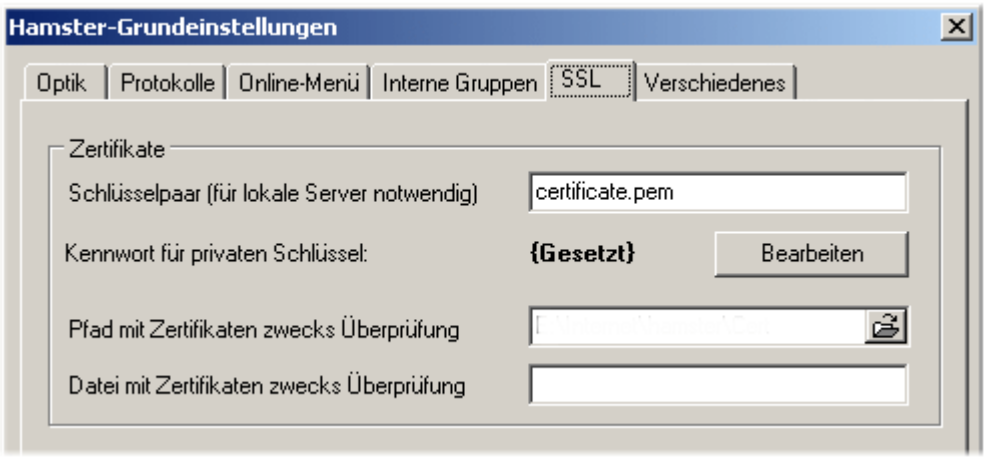
```
Country Name (2 letter code) [DE]:
```

Eingabe: Landeskennung: Vorgabe ist DE für Deutschland

Jetzt gibt es eine neue Datei in diesen Verzeichnis: „**certificate.pem**“ Diese enthält den öffentlichen und
privaten Schlüssel des erzeugten Zertifikates. Diese Datei wird nun in das Hamsterverzeichnis kopiert. Die Dateien „**libeay32.dll**“, „**libssl32.dll**“
(Zur Verwendung mit Hamster Classic muß die Datei ssleay32.dll in libssl32.dll umbenannt werden.) und „**openssl.exe**“ kommen ebenfalls in das Hamsterverzeichnis.

In der Hamsteroberfläche geht es in das Menü „**Einstellungen / Grundeinstellungen**“.
Die Option „Temp. erweiterte Einstellungen“

muß nun aktiviert sein um die SSL-Einstellungen vorzunehmen.
Im Feld „Schlüsselpaar“ wird nun die Datei „certificate.pem“ angegeben. Und das Kennwort des Zertifikates
wird über die Schaltfläche „Bearbeiten“ angeben. Mit OK werden die Einstellungen übernommen, Hamster
beendet und neu gestartet.



Nun muß man in den Servereinstellungen angeben, daß SSL verwendet werden kann.
Im Menü „**Einstellungen / Lokale Server ...**“ kann man nun für jeden Server die Option: „Erlaube TLS“ auswählen.

Jetzt ist es möglich daß z.B. E-Mailprogramme Ihre E-Mails mit einer SSL-verschlüsselten Verbindung am
einstellten POP3-Port bei Hamster abholen können. Nicht jedes Programm ist jedoch in der Lage SSL-Verbindungen zu Hamster erfolgreich aufzubauen. Leider fällt hierbei Microsoft Outlook 2000, Mozilla Thunderbird 0.72 durch. Im Test funktionierte es nur mit den E-Mailprogramm „The Bat!“
in der Version 2.12.00 richtig.

2. Hamster holt E-Mails

Unser kleiner Nager bekommt nun die Aufgabe über eine SSL-verschlüsselte Verbindung nun auch E-Mail
von einen POP3-Server abzuholen. Einige Vorbereitungsarbeit ist notwendig, damit alles reibungslos funktioniert.
Grundlage für diese Anleitung ist der Infotext von Robert Lieske Link und
die Ausführungen zum Thema SSL in der Hamster-Hilfedatei.

Am Beispiel des Dienstanbieters Web.de beschreibe ich die Vorgehensweise.
In der Hilfe von Web.de ist zu lesen, welche Einstellungen für SSL zu verwenden sind:

POP3 (SSL)	pop3.web.de	Port: 995
SMTP (SSL)	smtp.web.de	Port: 25

Vorbereitung

Nun benötigen wir das Zertifikat vom POP3-Server. Dieses beschaffen wir uns aus einer mitgeloggten Verbinung zu diesen Server.
In unseren Verzeichnis in der wir schon das Zertifikat erstellt habe werden wir nun weiter machen.

Wir gehen online und geben folgendes in der Commandozeile ein:

```
openssl.exe s_client -connect pop3.web.de:995 -showcerts > OpenSSL.log
```

Die Openssl.exe baut nun eine Verbindung zu diesen POP3-Server auf. Dieser sendet nun sein Zertifikat
und wartet auf die Antwort von Openssl das gleiche zu tun. Die Verbindung kommt zum Abbruch ….verständlich.
Die Logdatei enthält nun das Zertifikat des POP3-Servers. Wir schauen mal: (OpenSSL.Log)

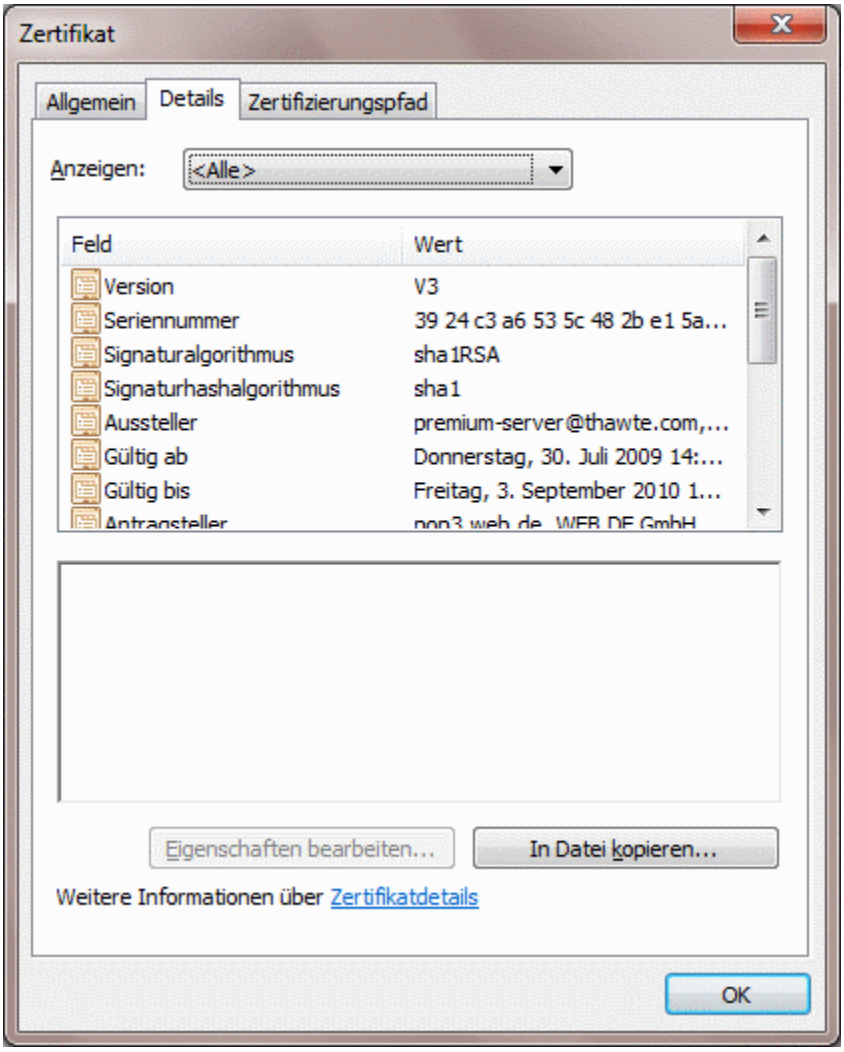
```
 CONNECTED(00000168)
—
Certificate chain
0 s:/C=DE/ST=Baden-Wuerttemberg/L=Karlsruhe/O=WEB.DE GmbH/CN=pop3.web.de
i:/C=ZA/ST=Western Cape/L=Cape Town/O=Thawte Consulting cc/
OU=Certification Services Division/CN=Thawte Premium Server CA/
emailAddress=premium-server@thawte.com
—
Server certificate
```

```
 ——BEGIN CERTIFICATE——
MIIDZTCCAs6gAwIBAgIQOSTDplNcSCvhWmQIXXQGBjANBgkqhkiG9w0BAQUFADCB
zjELMAkGA1UEBhMCWkExFTATBgNVBAgTDFdlc3Rlcm4gQ2FwZTESMBAGA1UEBxMJ
Q2FwZSBUb3duMR0wGwYDVQQKExRUaGF3dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UE
CxMfQ2VydGlmaWNhdGlvbiBTZXJ2aWNlcyBEaXZpc2lvbjEhMB8GA1UEAxMYVGhh
d3RlIFByZW1pdW0gU2VydmVyIENBMSgwJgYJKoZIhvcNAQkBFhlwcmVtaXVtLXNl
cnZlckB0aGF3dGUuY29tMB4XDTA5MDczMDEyMjQwM1oXDTEwMDkwMzA5MzEzNVow
ajELMAkGA1UEBhMCREUxGzAZBgNVBAgTEkJhZGVuLVd1ZXJ0dGVtYmVyZzESMBAG
A1UEBxMJS2FybHNydWhlMRQwEgYDVQQKEwtXRUIuREUgR21iSDEUMBIGA1UEAxML
cG9wMy53ZWIuZGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAORsV+BxgXzp
UJLsb6mZwIIifM933EloK1zsb6KZSCNgd05ifp1JfGHlMXxtMRaR5iIV7ejaPvue
kkYzOU/VmXCsP5YeSFp4a2GZKZ/bXpYAqucz1QJr0sVqxAKD0Boew+c2hB9c/7Ne
3lhKy1bwc+H4e34kU+L6OBw/NeoxiFW7AgMBAAGjgaYwgaMwHQYDVR0lBBYwFAYI
KwYBBQUHAwEGCCsGAQUFBwMCMEAGA1UdHwQ5MDcwNaAzoDGGL2h0dHA6Ly9jcmwu
dGhhd3RlLmNvbS9UaGF3dGVQcmVtaXVtU2VydmVyQ0EuY3JsMDIGCCsGAQUFBwEB
BCYwJDAiBggrBgEFBQcwAYYWaHR0cDovL29jc3AudGhhd3RlLmNvbTAMBgNVHRMB
Af8EAjAAMA0GCSqGSIb3DQEBBQUAA4GBAMoPCbHShZt6esCnPEg0Ougl6d9zFgDM
PixJUd+2RwwLNvqAgG3WkcDEm0wCuzcHZ1BEyG8VkYVuDYvf1zB86zSOc4eR0+pE
vaalb3CaYbZd2QgYqpSLQ3WNKA2T8lw2gfqpeEiu1m1on+OoH++b8tjBMysQsj0i
6qbBWipzVvVg
——END CERTIFICATE——
 subject=/C=DE/ST=Baden-Wuerttemberg/L=Karlsruhe/O=WEB.DE GmbH/CN=pop3.web.de
issuer=/C=ZA/ST=Western Cape/L=Cape Town/O=Thawte Consulting cc/
OU=Certification Services Division/CN=Thawte Premium Server
CA/emailAddress=premium-server@thawte.com
—
No client certificate CA names sent
—
SSL handshake has read 1035 bytes and written 340 bytes
—
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol : TLSv1
Cipher : AES256-SHA
Session-ID: 563D5158F42668AED52FEE44CD8862D21F7B0455493E3EDD4A3F03CF99AF329C
Session-ID-ctx:
Master-Key: 42FFF6DC41F1ACD55A608B7ADC013D35B7FA7FD83CCD54F4BEBC3D
A9691BA8714E74868719D656C67775CF70A6955781
Key-Arg : None
Start Time: 1276012123
Timeout : 300 (sec)
Verify return code: 21 (unable to verify the first certificate)
—
+OK POP server ready H miweb101
```

Das Zertifikat befindet sich nun hier:

```
 ——BEGIN CERTIFICATE——
MIIDZTCCAs6gAwIBAgIQOSTDplNcSCvhWmQIXXQGBjANBgkqhkiG9w0BAQUFADCB
. . . . .
——END CERTIFICATE——
```

Dies wird nun markiert und in ein Editor eingefügt. Abgespeichert wird es mit den Namen „pop3.web.de.pem“
Später wissen wir nun an Hand des Dateinamens genau von welchen Server das Zertifikat stammt.
Wir sehen uns nun das Zertifikat an und benennen die Datei um. Hierbei wird einfach die Dateiendung von .PEM in .CER
gewechselt. Jetzt haben wir also eine Datei namens „**pop3.web.de.cer**“.
Ein Doppelklick auf die Datei und wir können uns das Zerifikat ansehen.

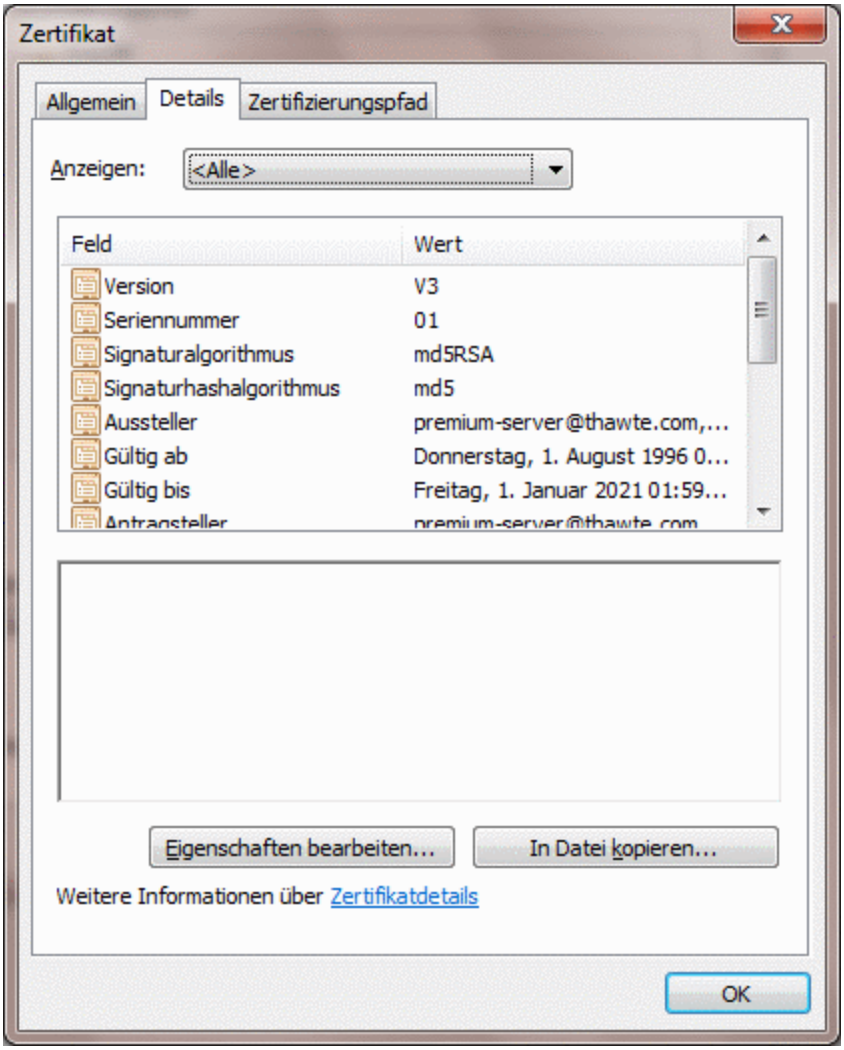


Zu sehen ist, das dieses Zertifikat nicht von Web.de ausgestellt wurden ist sondern von „Thawate Premium Server CA“. Die Gültigkeit ist auch zu sehen. Wir wechseln in die Registerkarte „Details“ und lesen die Informationen im Feld „Aussteller“. Wir sehen:

„E = premium-server@thawte.com“ In der Registerkarte „Zerifizerungspfad“ sieht man den Namen des Zerifikats „**Thawate**“.

Also benötigen wir noch das Zertifikat des Ausstellers dieses Zertifikats - das Rootzertifikat. Nun schnell mal online auf http://www.thawte.com und suchen das Rootzertifikat. Auf der Internetseite von Thawte gibt man im Suchefeld: „root certificate“ ein. Jetzt kommt man doch auf die Seite zum Download des Zertifikats ... im ZIP-Dateiformat vorliegt.

In diesen Archiv finden wir die Datei „ThawteServerCA.cer“ und es handelt sich um unser gesuchtes Zertifikat. Auch hier kann man man mit einen Doppelklick sich das Zertifikat ansehen.



Hier ist das gesuchte Stammzertifikat.
Mit diesen Zerifikat wurde das Zertifikat für den Server „pop3.web.de“ digital signiert.

Schade nur das dieses Zertifikat im falschen Format vorliegt. Wir müssen wieder Openssl.exe benutzen:

2.1 Zertifikatsformat ändern

```
openssl.exe x509 -inform DER -in ThawteServerCA.cer -out thawte.serverca.pem -outform PEM
```

Nun haben wir das richtige Format des Zertifikats als Datei „thawte.serverca.pem“ vorliegen. Hamster kann jedoch nichts damit anfangen.

Nun machen wir denn mal das „ThawateServerCA“-Zertifikat für Hamster schmackhaft, so daß er auch was damit umgehen kann.

2.2. Hashwert bestimmen

Nein, Hamster ist nicht drogensüchtig … 🙂
Falls uns das Stammzertifikat gleich im PEM-Format vorliegt, können wir den vorhergehenden Schritt weglassen.

```
openssl.exe x509 -inform DER -in ThawteServerCA.cer -out thawte.serverca.pem -outform PEM
```

Die Ausgabe des Hashwertes ist in der Datei „thawte.server.hash“ eingetragen.
Beim Öffnen der Datei lesen wir den Wert aus:

```
ddc328ff
```

Diesen Wert brauchen wir gleich.

2.3. Zertifikat in Textform erstellen

Jetzt muß noch die Textform der Zertifikats her: Als Outputdatei geben wir hier den **<Hashwert>.txt** an.

```
openssl.exe x509 -text -noout -in thawte.serverca.pem -out ddc328ff.txt
```

Nun benennen wir das Zertifikat **thawte.serverca.pem** in: **<Hashwert>.0** um.

```
ren thawte.serverca.pem ddc328ff.0
```

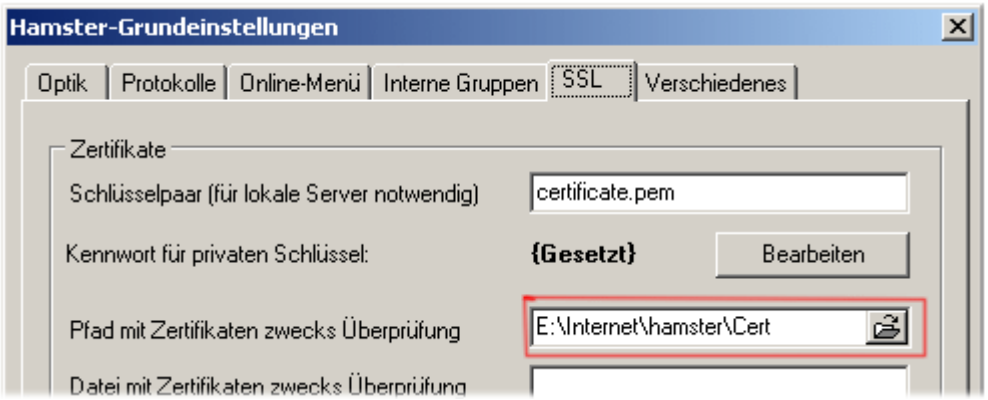
So … jetzt erstellen wir im Hamsterverzeichnis ein Unterverzeichnis z.B. „`cert`“.
Beide nun vorhanden Dateien „<Hashwert>.0“ und „<Hashwert>.txt“ werden nun in dieses Verzeichnis kopiert.

2.4 Zertifikat von "pop3.web.de" bereitstellen

Jetzt muß nur noch das Zertifikat „pop3.web.de.cer“ die gleiche Prozedur „Hashwert bestimmen“ und „Zerifikat in Textform erstellen“ auch noch durchlaufen und in das neu ersteller Verzeichnis „`cert`“ kopiert werden.

2.5. Hamster einstellen

Nun muß Hamster auch wissen, daß es Zertifikate zu Verifizierung in einen Verzeichnis vorhanden sind.\
Im Menü „Einstellungen / Grundeinstellungen“, Registerkarte „SSL“ wird nun im Feld:
Pfad mit Zerifikatenzwecks Überprüfung das neu erzeugte Verzeichnis angeben.



Mit **OK** wird die Änderung bestätigt.

2.6. Script anpassen

HamFetchMail(<server>, <port>, <user>, <pass>, <destuser>, <filter>, <LeaveOnServer>,

<SSLMode>, <SSLVerify>, <SSLCaFile>)

Dieser Befehl dient zum Abholen von E-Mails von einen POP3-Server in einen Script.

Parameter		
<server>	POP3-Server	z.B. pop3.web.de
<port>	POP3-Serverport	z.B. 995 (für SSL)
<user>	Benutzernamen	
<pass>	das dazugehörige Passwort	
<destuser>	lokaler Benutzer welcher die E-Mail erhalten soll	
<filter>	Name des Filters in MailFilt.hst	
<LeaveOnServer>	0 = auf Server löschen 1 = auf Server belassen	
<SSLMode>	0 - SSL/TLS abgeschaltet 1 - SSL auf separatem Port (sPOP3 bzw, sSMTP) 2 - TLS, wenn möglich 3 - TLS wird erzwungen	
<SSLVerify>	Überprüfung der fremden Server-Zertifikate: 0 - Zertifikatsüberprüfung abgeschaltet 1 - Zertifikatsüberprüfung, falls Zertifikat vorgelegt 2 - Zertifikatsüberprüfung immer 3 - Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie	
<SSLCaFile>	kann eine Datei zur Zertifikatsüberprüfung angegeben werden	

Beispiel:

```
HamFetchMail("pop3.web.de","995","$6","","InBoundMail","InBound-Filter",0,1,3,"")
```

Holt E-Mails über eine verschlüsselte SSL-Verbindung von pop3.web.de an Port: 995 und übergibt diese den Benutzer „InBoundMail“
und filtert über „InBound-Filter“ alle eingenden E-Mails. Dabei werden alle E-Mails vom Server gelöscht.
Das Zertifikat wird über die lokale Kopie im Unterverzeichnis *<Hamsterverzeichnis>*/cert geprüft.

3. Hamster sendet E-Mails

Jetzt wird es Zeit, daß Hamster EMails via SSL an einen SMTP-Host sendet. In useren Beispiel ist es „smtp.web.de“ auf Port 25.
Diese Informationen stellt der Dienst „freemail.web.de“ zur Verfügung.

3.1 Das Zertifikat von smtp.web.de

Nun brauchen wird das Zertifikat des SMTP-Servers und rufen die OpenSSL.exe wie folgt auf:

```
openssl.exe s_client -connect smtp.web.de:25 -starttls smtp -showcerts > OpenSSL.log
```

In der LOG-Datei „OpenSSL.log“ ist nun das Zertifikat des Servers smtp.web.de.
Auszug aus der LOG-Datei:

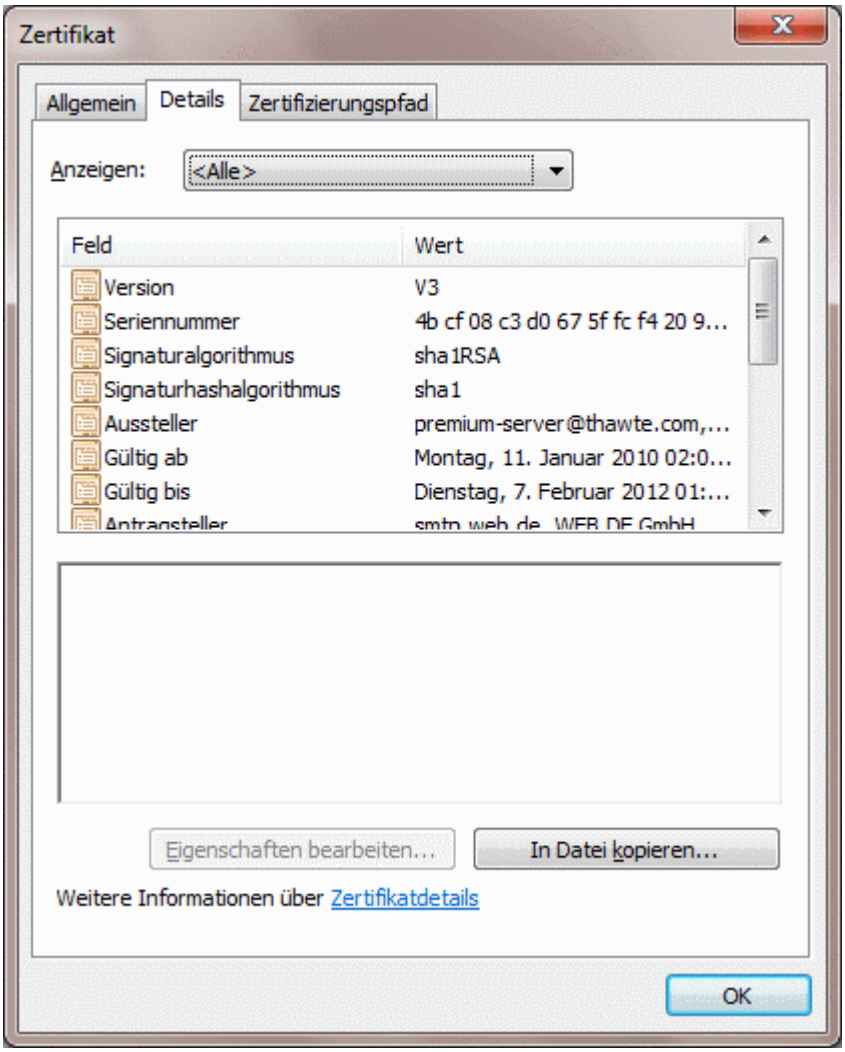
```
 CONNECTED(00000168)
—
Certificate chain
0 s:/C=DE/ST=Baden-Wuerttemberg/L=Karlsruhe/O=WEB.DE GmbH/CN=smtp.web.de
i:/C=ZA/ST=Western Cape/L=Cape Town/O=Thawte Consulting cc/OU=Certification
Services Division/CN=Thawte Premium Server CA/emailAddress=premium-
server@thawte.com
—
Server certificate
 ——BEGIN CERTIFICATE——
MIIDZTCCAs6gAwIBAgIQS88Iw9BnX/z0IJtuEodEyzANBgkqhkiG9w0BAQUFADCB
zjELMAkGA1UEBhMCWkExFTATBgNVBAgTDFdlc3Rlcm4gQ2FwZTESMBAGA1UEBxMJ
Q2FwZSBUb3duMR0wGwYDVQQKExRUaGF3dGUgQ29uc3VsdGluZyBjYzEoMCYGA1UE
CxMfQ2VydGlmaWNhdGlvbiBTZXJ2aWNlcyBEaXZpc2lvbjEhMB8GA1UEAxMYVGhh
d3RlIFByZW1pdW0gU2VydmVyIENBMSgwJgYJKoZIhvcNAQkBFhlwcmVtaXVtLXNl
cnZlckB0aGF3dGUuY29tMB4XDTEwMDExMTAwMDAwMFoXDTEyMDIwNjIzNTk1OVow
ajELMAkGA1UEBhMCREUxGzAZBgNVBAgTEkJhZGVuLVd1ZXJ0dGVtYmVyZzESMBAG
A1UEBxQJS2FybHNydWhlMRQwEgYDVQQKFAtXRUIuREUgR21iSDEUMBIGA1UEAxQL
c210cC53ZWIuZGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAN3gTpL4FiIp
7G3KTZwHkAlXpHhvyueBIpsAV5qSSNkKvSAOu2UzYLJVc5NjSUWgYz/55Qm2mZc0
eFYV2d2eIPWmLPycYJu18si8W4CehEqlKsT/BnJIFGEUTr/HstYWRlILVm8L5vUJ
3a3uaIL5MZ8T1oXjqQ/w8FzJCv2ft0UHAgMBAAGjgaYwgaMwDAYDVR0TAQH/BAIw
ADBABgNVHR8EOTA3MDWgM6Axhi9odHRwOi8vY3JsLnRoYXd0ZS5jb20vVGhhd3Rl
U2VydmVyUHJlbWl1bUNBLmNybDAdBgNVHSUEFjAUBggrBgEFBQcDAQYIKwYBBQUH
AwIwMgYIKwYBBQUHAQEEJjAkMCIGCCsGAQUFBzABhhZodHRwOi8vb2NzcC50aGF3
dGUuY29tMA0GCSqGSIb3DQEBBQUAA4GBAK3L/LOg61zvKop3Vk6iNZP3KUVsCvOE
OFGMj5QxxJQhY8EZgPlDmTYAVns6jFi2+T7+R/w1j/nKTJFSLXkuKc1ZT2p6BP8z
G8TY6k74rZK0vKpbrqP+6RYE31Eo48kPfdFMvDzoeIjunx87MoGUNOlMbXC/wUBF
+DaI69u75nJr
——END CERTIFICATE——
 subject=/C=DE/ST=Baden-Wuerttemberg/L=Karlsruhe/O=WEB.DE GmbH/CN=smtp.web.de
issuer=/C=ZA/ST=Western Cape/L=Cape Town/O=Thawte Consulting
cc/OU=Certification
Services Division/CN=Thawte Premium Server CA/emailAddress=premium-
server@thawte.com
—
No client certificate CA names sent
—
SSL handshake has read 1282 bytes and written 375 bytes
—
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Compression: NONE
Expansion: NONE
SSL-Session:
Protocol : TLSv1
Cipher : AES256-SHA
Session-ID: 0BE41738458F6CAF752A88AC0E17D7DE26A4DEB33D9DAC13D09750D7D1AA64D4
Session-ID-ctx:
Master-Key: 771105DEE7A187183469443D1A6573D11A12B30FAB5CEDD78B5A3D0A9D419CBD25
2921B273A35E46B2BAA3720F1E9FEA
Key-Arg : None
Start Time: 1276014105
Timeout : 300 (sec)
Verify return code: 21 (unable to verify the first certificate)
—
421 smtp03.web.de: SMTP command timeout - closing connection
```

Das Zertifikat befindet sich nun hier:

```
 ——BEGIN CERTIFICATE——
MIIDZTCCAs6gAwIBAgIQS88Iw9BnX/z0IJtuEodEyzANBgkqhkiG9w0BAQUFADCB
. . . . .
——END CERTIFICATE——
```

Dies wird nun markiert und in ein Editor eingefügt. Abgespeichert wird es mit den Namen „**smtp.web.de.pem**„

Später wissen wir nun an Hand des Dateinamens genau von welchen Server das Zertifikat stammt.
Wir sehen uns nun das Zertifikat an und benennen die Datei um. Hierbei wird einfach die Dateiendung von .PEM in .CER gewechselt.
Jetzt haben wir also eine Datei namens „**smtp.web.de.cer**“. Ein Doppelklick auf die Datei und wir können uns das Zerifikat ansehen.



... und hier ist das Zertifikat von „smtp.web.de“.
Zum Glück hat auch hier der gleiche Aussteller des Zertifikats wie „pop3.web.de“ auch dieses Zertifikat signiert.
Das benötigte Root-Zertifikat haben wir ja schon.

Jetzt muß das Zertifikat für Hamster gebrauchbar gemacht werden.

3.2. Hashwert bestimmen

```
openssl.exe x509 -hash -noout -in smtp.web.de.cer > smtp.web.de.hash
```

Die Ausgabe des Hashwertes ist in der Datei „smtp.web.de.hash“ eingetragen.
Beim Öffnen der Datei lesen wir den Wert aus:

```
a75426a4
```

Diesen Wert brauchen wir gleich.

3.3. Zertifikat in Textform erstellen

Jetzt muß noch die Textform der Zertifikats her:
Als Outputdatei geben wir hier den **<Hashwert>.txt** an.

```
openssl.exe x509 -text -noout -in smtp.web.de.cer -out a75426a4.txt
```

Nun benennen wir das Zertifikat **smtp.web.de.cer** in: **<Hashwert>.0** um.

```
ren smtp.web.de.cer a75426a4.0
```

So ... jetzt erstellen wir im Hamsterverzeichnis ein Unterverzeichnis z.B. „cert“.
Beide nun vorhanden Dateien „**<Hashwert>.0**“ und „**<Hashwert>.txt**“ werden nun in dieses Verzeichnis kopiert.

3.4. Script anpassen

Damit Hamster nun auch EMails via SSL an den SMTP-Server senden kann, muß man in einen Script den
Befehl „HamSendMailAuth“ anpassen.

HamSendMailAuth(<server>, <port>, <user>, <pass>, <from-select>, <to-select>, <SSLMode>, <SSLVerify>, <SSLCaFile>)

Dieser Befehl dient zum Versenden von E-Mails an einen SMTP-Server in einen Script.

Parameter		
<server>	SMTP-Server	z.B. smtp.web.de
<port>	SMTP-Serverport	z.B. 25
<user>	Benutzernamen	
<pass>	das dazugehörige Passwort	
<from-select>	Entspricht der Angabe im Header „FROM“ der zu versendeten EMail	z.B: „user1@example.com“
<to-select>	Entspricht der Angabe im Header „TO“ der zu versendenten EMail	
<SSLMode>	0 - SSL/TLS abgeschaltet 1 - SSL auf separatem Port (sPOP3 bzw, sSMTP) 2 - TLS, wenn möglich 3 - TLS wird erzwungen	
<SSLVerify>	Überprüfung der fremden Server-Zertifikate: 0 - Zertifikatsüberprüfung abgeschaltet 1 - Zertifikatsüberprüfung, falls Zertifikat vorgelegt 2 - Zertifikatsüberprüfung immer 3 - Zertifikatsüberprüfung immer und Vergleich des Serverzertifikats mit lokaler Kopie	
<SSLCaFile>	kann eine Datei zur Zertifikatsüberprüfung angegeben werden	

Beispiel:

```
HamSendMailAuth( "smtp.web.de", "smtp", "$6", "",
"user1@example.com",".*",3,3,"" )
```

Versendet eine EMail via SSL über den Server smtp.web.de Port:25 mit den gespeicherten Benutzernamen und Kennwort aus der Liste $6

nur vom Benutzer `user1@example.com`.

Nun dürfte eine sichere Kommunikation via SSL für Sicherheit sorgen. Benutzernamen und Kennwörter
können nun nicht mehr von anderen Personen mitgelesen werden.
Jetzt fehlt nun noch die „sichere EMail“ - also eine verschlüsselte EMail die nur der Empfänger lesen kann.

Download

Diese Dokumentation steht auch als PDF-Datei zur Verfügung:

a1 [A1]	Doku "Hamster & SSL" von 2014	(PDF-Datei)	
a2 [A2]	Doku "Hamster: EMails über SSL" von 2006	(PDF-Datei)	

Links

Weitere Links zu diesen Thema.

01 [01]	Hamster-Classic	(Seite von Thomas G. Liesner)	
02 [02]	Win32 OpenSSL - Slprpweb.com	OpenSSL für Windows	