

ANHANG A

Das SMTP-Protokoll

Dieser Anhang gibt eine kurze Einführung in das SMTP-Protokoll, um zum Verständnis des Buchs beizutragen.

Grundlagen

SMTP steht für Simple Mail Transfer Protocol. Es regelt die vom Absender gesteuerte Übertragung von elektronischer Post (E-Mail) zwischen zwei Rechnern. Das »Abholen« von elektronischer Post vom Server auf den Client, also die vom Empfänger gesteuerte Übertragung, wird üblicherweise über die Protokolle IMAP oder POP3 durchgeführt und ist somit hier nicht Thema.

Das SMTP-Protokoll ist im RFC 2821 definiert. Dieses Dokument ersetzt das ursprüngliche Dokument RFC 821, auf das gelegentlich noch verwiesen wird. Diese Dokumente definieren nur das Übertragungsprotokoll. Das interne Format einer gültigen E-Mail-Nachricht ist in RFC 822 und anderen Dokumenten definiert und vom SMTP-Protokoll vollkommen unabhängig.

SMTP ist ein Client/Server-Protokoll. Der Client sendet Befehle an den Server, dieser antwortet mit einer Bestätigung oder einem Fehler. Der Client ist der Absender, der Server ist der Empfänger der E-Mail. Beide Parteien können aber auch Mittelsmänner in einer Übertragungsreihe sein und müssen nicht der ursprüngliche Absender oder der endgültige Empfänger sein. Man spricht bei einer solchen Weiterleitung von Relaying.

SMTP ist textbasiertes Protokoll. Alle Befehle und Antworten sind lesbare Zeichenketten. Zu Testzwecken ist es also möglich und üblich, SMTP-Befehle von Hand einzutippen. SMTP erfordert am Ende jeder Zeile ein Carriage-Return gefolgt von einem Zeilenwechsel, also Zeilenenden im Windows-Stil. Viele SMTP-Server haben dabei jedoch eine gewissen Flexibilität.

Der SMTP-Dialog

Der folgende Ausschnitt zeigt einen vollständigen SMTP-Dialog. Mit dem Befehl

```
telnet hostname 25
```

kann man derartige Testsitzungen selbst durchführen. Die hervorgehobenen Zeilen werden vom Client gesendet, die anderen Zeilen sind die Antworten des Servers.

```
220 mail.example.net ESMTP Postfix (Debian/GNU)
HELO workstation.example.net
250 mail.example.net
MAIL FROM: <joe@example.net>
250 Ok
RCPT TO: <bob@elsewhere.org>
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: foo <foo@bar.com>
To: bar <bar@foo.com>
Subject: Test

This is a test.
.
250 Ok: queued as 20FCE3C0026
QUIT
221 Bye
```

Die Zeilen des Servers beginnen stets mit einer Zahl, dem Statuscode. Daran kann das Client-Programm erkennen, ob die Eingabe erfolgreich bearbeitet wurde. Die Statuscodes werden unten erklärt. Der jeweils nachfolgende Text ist für die automatische Bearbeitung uninteressant. Er ist nur für Rückmeldungen an den Benutzer oder für Log-Aufzeichnungen relevant.

Die Verwendung von Groß- und Kleinschreibung in SMTP-Befehlen ist egal, mit Ausnahme der Benutzernamen in E-Mail-Adressen. Gewöhnlich werden SMTP-Befehle großgeschrieben, wie auch hier.

In den folgenden Abschnitten wird der Verlauf des SMTP-Dialogs erläutert.

Begrüßung

Zuerst sendet der Server eine Begrüßungszeile. Diese enthält üblicherweise den Hostnamen des Servers sowie Informationen über das verwendete SMTP-Server-Produkt.

Als ersten Befehl muss der Client den Befehl HELO mit dem eigenen Hostnamen als Argument senden. Konzeptionell dient dieser Befehl als Begrüßung des Clients an den Server, ist aber aus heutiger Sicht funktionell sinnlos. Aus diesem Grund wird der Befehl von selbst gebauten Skripten oder schlecht implementierter Spam-Software häufig vergessen. SMTP-Server ignorieren den Fehler zwar in der Regel, kön-

nen aber so konfiguriert werden, dass sie den HELO-Befehl verlangen. Laut SMTP-Standard wird der angegebene Hostname nicht überprüft, aber auch das kann teilweise eingeschaltet werden und der Spam-Abwehr dienen. Kapitel 3, *Spam- und Virenabwehr mit Postfix, Exim und Sendmail* enthält Informationen dazu.

Clients, die das erweiterte SMTP-Protokoll (ESMTP) unterstützen – heutzutage die meisten –, senden stattdessen den Befehl EHLO. Für die Betrachtungen hier ist dieser Befehl aber identisch.

Als Antwort auf HELO oder EHLO gibt der Server seinen eigenen Hostnamen zurück.

Angabe des Absenders

Nach der Begrüßung sendet der Client den Befehl MAIL FROM: gefolgt von der Absenderadresse. Diese Adresse muss laut RFC 821 wie gezeigt in spitzen Klammern stehen und enthält nur die Adresse und keine Namen. Angaben wie

```
MAIL FROM: joe@example.net
```

oder

```
MAIL FROM: Joe User <joe@example.net>
```

sind also ungültig. (Sie entsprechen dem Standard RFC 822 für den *Inhalt* von E-Mail-Nachrichten.) Viele SMTP-Server akzeptieren aber zumindest auch die erste Form.

Der angegebene Absender kann im Prinzip frei erfunden werden, wenn man seine Identität zu verbergen versucht. Einige SMTP-Server können so konfiguriert werden, dass sie zumindest die Plausibilität der Absenderangabe überprüfen.

Ein Sonderfall ist die Angabe:

```
MAIL FROM: <>
```

Dies ist ein so genannter Null Return Path, der verwendet wird, wenn das E-Mail-System eine Fehlermeldung sendet. Falls die Fehlermeldung selbst einen Fehler erzeugt, weiß das E-Mail-System, dass es auf diese E-Mail nicht antworten soll und kann.

Auf die Angabe des Absenders reagiert der Server mit einer kurzen Bestätigung.

Angabe der Empfänger

Nach der Angabe des Absenders sendet der Client den Befehl RCPT TO: gefolgt von der Adresse des Empfängers im selben Format wie oben beschrieben. Die Empfängeradresse muss natürlich richtig sein, wenn die E-Mail ankommen soll.

Auf die Angabe des Empfängers reagiert der Server wieder mit einer kurzen Bestätigung.

Eine E-Mail kann mehrere Empfänger haben. Wenn das der Fall ist, wird der Befehl einfach mit anderen Adressen wiederholt.

SMTP-Server sollten in aller Regel nicht beliebige Empfängeradressen akzeptieren. Sonst könnte jeder Rechner im Internet an jeden anderen Rechner im Internet E-Mails über diesen SMTP-Server versenden. Das nennt sich offenes Relay und ist eine Einladung an Spammer, ihre E-Mails anonym über diesen Server zu versenden. Die Adressen von offenen Relays werden auf DNS-Blackhole-Lists gesammelt und von vielen Anwendern blockiert.

SMTP-Server sollten daher nur E-Mails annehmen, die an die eigene Domain gerichtet sind oder aus dem eigenen Netzwerk gesendet wurden. Die SMTP-Server-Produkte haben normalerweise eine derartige Voreinstellung.

Übertragung der Daten

Nach der Angabe der Empfänger sendet der Client den Befehl DATA, um anzuzeigen, dass er jetzt den Inhalt der E-Mail senden möchte. Darauf reagiert der Server mit einer Meldung, die anzeigt, wie die Daten formatiert werden sollen. Client-Programme lesen diese Meldung natürlich nicht, da die Daten immer auf dieselbe Weise formatiert werden.

Es ist zu beachten, dass der Inhalt der E-Mail vollkommen unabhängig von den oben getätigten Angaben zu Absender und Empfänger sind. Von diesem Umstand wird beim Versand von Spam und Viren über E-Mail reichlich Gebrauch gemacht. Da der Inhalt der E-Mail nicht in den Verantwortungsbereich des SMTP-Servers fällt, sind auch keine Schutzmechanismen gegen derartige Fälschungen vorgesehen. Lediglich Exim bietet zurzeit die Option, die Adressen im Inhalt der E-Mail zu überprüfen.

Abgeschlossen wird der Inhalt der E-Mail mit einem Punkt allein auf einer Zeile.

Danach muss der SMTP-Server die E-Mail endgültig annehmen oder ablehnen. Wenn er sie annimmt, hat er die Verantwortung für sie übernommen und muss sie entweder an den endgültigen Bestimmungsort leiten oder per E-Mail eine Fehlermeldung an den Absender schicken. Der Client hat sich in diesem Moment der Verantwortung für die E-Mail entledigt und kann sie beispielsweise aus dem lokalen Speicher löschen.

Verbindungsende

Nach Abschluss der E-Mail-Übertragung sendet der SMTP-Client normalerweise den Befehl QUIT, worauf der Server die Verbindung schließt. Ein Client könnte nach dem Ende einer E-Mail-Übertragung auch die nächste Übertragung beginnen, indem er einfach wieder mit MAIL FROM anfängt.

Umschlag und Inhalt

Wie bereits gesehen, haben die Adressangaben im SMTP-Dialog und im Inhalt der E-Mail nichts miteinander zu tun. Man kann sich den SMTP-Dialog als Umschlag eines Briefs vorstellen und die Daten im DATA-Teil als Inhalt des Briefs. Die im Befehl MAIL FROM angegebene Adresse wird deshalb auch Envelope-Absender («Umschlag-Absender») genannt, die Adressen im Befehl RCPT TO entsprechend Envelope-Empfänger.

Nur die Adressen auf dem Umschlag eines Briefs werden von der Post für die Zustellung herangezogen. An wen der Brief gerichtet ist oder auf welchem Briefpapier er gedruckt ist, ist dabei uninteressant. Ebenso wird nur die Absenderadresse auf dem Umschlag verwendet, wenn der Brief nicht zugestellt werden kann. Wer den Brief wirklich geschrieben hat, ist irrelevant.

Analog funktioniert es beim Versand von E-Mails. Nur der Envelope-Empfänger bestimmt, wohin die E-Mail gesendet wird. Und nur der Envelope-Absender bestimmt, wohin Fehlermeldungen gesendet werden. Die Adressen in den E-Mail-Headern wie From und Reply-To werden von der SMTP-Software nicht beachtet und sind höchstens für E-Mail-Leseprogramme relevant.

Weitere Befehle

Das SMTP-Protokoll unterstützt einige weitere Befehle, die nicht im Rahmen einer E-Mail-Übertragung verwendet werden:

VRFY

Dieser Befehl überprüft, ob eine E-Mail-Adresse existiert, zum Beispiel:

```
VRFY joe
252 joe
VRFY joe@example.net
252 joe@example.net
VRFY bob
550 <bob>: Recipient address rejected: User unknown in local recipient table
VRFY joe@foo.com
554 <joe@foo.com>: Relay access denied
```

In der heutigen »feindseligen« Internetumgebung ist ein solcher Befehl eher ein Sicherheitsrisiko als ein Benutzerdienst. Normalerweise bietet der SMTP-Server daher die Möglichkeit, ihn auszuschalten.

EXPN

Dieser Befehl gibt die Mitglieder einer Mailingliste aus. Er ist ebenfalls ein Sicherheitsrisiko und in den meisten SMTP-Servern gar nicht implementiert.

RSET

Der Server wird angewiesen, einen »Reset« auszuführen. Danach kann mit einer neuen E-Mail-Übertragung (MAIL FROM) begonnen werden.

HELP

Dieser Befehl sollte einen Hilfetext ausgeben, ist aber größtenteils nicht implementiert.

NOOP

Dieser Befehl macht gar nichts, und der Server gibt nur eine erfolgreiche Antwort zurück.

Statuscodes

Die vom SMTP-Server gesendeten Statuscodes folgen einem bestimmten Schema. Zunächst gilt: Steht nach der Zahl ein Leerzeichen, ist dies die letzte vom Server gesendete Zeile. Wenn der Server mehrere Zeilen senden muss, steht nach der Zahl ein Minuszeichen, außer in der letzten Zeile. Folgendes könnte zum Beispiel eine mehrzeilige Antwort auf den Befehl EHLO sein:

```
EHLO workstation.example.net
250-mail.example.net
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250 8BITMIME
```

Die erste Zahl des Codes besagt, um was für eine Antwort es sich handelt:

1yz

Der Befehl wurde vorläufig angenommen, muss aber bestätigt werden. Dieser Typ kommt in SMTP aktuell nicht vor.

2yz

Der Befehl wurde erfolgreich angenommen. Danach kann ein neuer Befehl gesendet werden.

3yz

Der Befehl wurde vorläufig angenommen, aber weitere Informationen werden benötigt, um ihn zu vervollständigen. Der SMTP-Client sollte die benötigten Informationen als Nächstes senden. Dieser Code wird als Reaktion auf den DATA-Befehl gesendet.

4yz

Der Befehl wurde abgelehnt, aber der Client kann später erneut versuchen, den Befehl zu senden. Dies wird üblicherweise bei vorübergehenden Problemen auf dem Serversystem verwendet.

5yz

Der Befehl wurde abgelehnt, und der Fehler ist permanent.

Hier folgt die vollständige Liste aller definierten SMTP-Statuscodes mit dem dazugehörigen Text aus RFC 2821. Der von einer bestimmten SMTP-Implementierung verwendete Wortlaut kann jedoch davon abweichen.

211

System status, or system help reply

214

Help message

220

<domain> Service ready

221

<domain> Service closing transmission channel

250

Requested mail action okay, completed

251

User not local; will forward to <forward-path>

252

Cannot VRFY user, but will accept message and attempt delivery

354

Start mail input; end with <CRLF>.<CRLF>

421

<domain> Service not available, closing transmission channel

450

Requested mail action not taken: mailbox unavailable (e.g., mailbox busy)

451

Requested action aborted: local error in processing

452

Requested action not taken: insufficient system storage

500

Syntax error, command unrecognized

501

Syntax error in parameters or arguments

502

Command not implemented

503

Bad sequence of commands



504

Command parameter not implemented

550

Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)

551

User not local; please try <forward-path>

552

Requested mail action aborted: exceeded storage allocation

553

Requested action not taken: mailbox name not allowed

554

Transaction failed

