The sys admin's daily grind: Swaks

For the Protocol

Searching for errors on an SMTP server via Telnet and test mails can seem like a never-ending obstacle course. The utility called Swaks helps bring the finish line within reach. *By Charly Kühnast*

SMTP is a plain text protocol, which proves to be quite useful when searching for errors on the mail server. I can simply connect to port 25 via Telnet and then act like a mail client. By means of the server's answers, I can quickly see where the problem is – theoretically, at least. In practice, as you may have already guessed, things don't go that smoothly.

SMTP has become fatter and fatter over the years, not least because security functions were desperately needed. Although basic commands like `HELO <FQDN>` or `RCPT` are easy for me to type quickly, authentication with CRAM-SHA-1 no longer has anything to do with plain text as far as I am concerned, and any fun I might have been having is long gone.

Swaks [1] has made diagnosing mail servers easy again. With the right parameters, the tool executes the SMTP dialogs for the diagnosis all by itself. I just send a simple test mail with:

```
swaks --from=charly@kuehnast.com ➥
      --to=charly@example.com ➥
      --server=mail.example.com
```

If the test is successful, charly@example.com will receive a test mail. I can follow each step of the SMTP dialog in the console and get any error messages served on a silver platter.

A more comprehensive example is shown in Figure 1. Among other things, Swaks has discovered that the mail server supports TLS. To find that out, I entered `-tls` (attention: with only one hyphen, not two). If the server were not capable of TLS, I would have received the message ** Host did not advertise STARTTLS*, and Swaks would have closed the SMTP dialog.

Beyond that, the example demonstrates SMTP authentication via the plain mechanism. In a real situation, you would, of course, use a secure method, because the plain mechanism doesn't encrypt the username or the password – the seemingly cryptic jumble of characters comes from the base64 encoding of the data. I tell Swaks that I want to use an authentication mechanism with the `--auth` parameter and give it my login data with

```
--auth-user=testuser ➥
--auth-password=testpass
```

Swaks and the mail server will then negotiate with each other to find the highest security level they both support.

Figure 1: Swaks takes on the role of an SMTP client and checks to see whether the mail server is behaving correctly.

Remedy for Congestion

When I send a dozen or more test messages while searching for errors, it can be quite bothersome when they or their bounces clog up my inbox. Swaks, therefore, allows me to terminate the SMTP dialog at any point I choose. In the example, I have entered `-q RCPT`, so Swaks closes the dialog after it has transferred the recipients' addresses to the mail sever and has received the answer:

```
RCPT TO:<charly@kuehnast.com>
250 2.1.5 Ok
QUIT
```

At this point, the most common errors (e.g., `HELO` string, TLS, and SMTP authentication) have already been ruled out. The sender and recipient addresses also have been confirmed with *OK*. Swaks can do a lot more than mentioned in this short introduction, so take a closer look at the man page. For example, since its last major release in Spring 2012, the tool can also handle IPv6. ■■■

INFO

[1] Swaks: *http://www.jetmore.org/john/code/swaks/*

AUTHOR

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.