## The sys admin's daily grind: Siege

# SIEGE WARFARE

The siege of Troy is said to have taken 10 years, ending only after Odysseus introduced a wooden horse into the mix. Charly is planning a siege, too, and the target is his own web server. Of course, he doesn't have 10 years to complete the task, and Odysseus isn't on his team.

**BY CHARLY KÜHNAST**

When contemplating my Apache server, I'm reminded of the mythical sea creature Scylla, which wiped out no fewer than six of Odysseus's men on his way home from Troy. My server's Worker MPM can devour multiple requests in one fell swoop. But where's the limit? How many threads do I need to configure to achieve maximum performance? And how many threads are just overkill?

## Stress Test

With the use of the attack tool Siege [1], I will try to answer these questions with a stress test. Siege supports two modes of attack. The first simulates human web-surfing behavior, which explains the gaps of three seconds between individual access instances.

In the second mode, benchmark mode, these gaps are dropped and Siege incessantly throws requests at the server. To make the server sweat, Siege fields a configurable army of users – 10 by default, but you can use the following pa-



```
File  Edit  View  Terminal  Tabs  Help
Transactions:              2566135 hits
Availability:              100.00 %
Elapsed time:              1800.48 secs
Data transferred:          16593.17 MB
Response time:             0.07 secs
Transaction rate:          1425.25 trans/sec
Throughput:                9.22 MB/sec
Concurrency:               99.40
Successful transactions:   2566014
Failed transactions:       0
Longest transaction:       9.01
Shortest transaction:      0.00

                           14,0-1        Top
```

**Figure 1: After the battle with the web server, Siege shows the results.**

rameter to increase the size of the virtual vanguard until either the web server cries mercy or the attackers run out of system resources:

```
--concurrent=<number>
```

## Which Way to Troy?

How does Siege know which server to attack? Again, there are two options. The parameter

```
--url="http://<my.site.com>/<index.
html>"
```

lets you enter a URL that Siege will reload repeatedly. The option

```
--file="/home/charly/
siege-urls.txt"
```

is more interesting in that I can add an arbitrarily long list of links to the file and Siege will attack them one after another. The *--reps = < number >* parameter lets me specify how many times Siege should repeat the stress test.

Also, I can set *--internet* to tell Siege not to request the pages in the order specified in the URL file, but to use a random order instead, which is far more realistic.

## Battle Over

To prevent the test going on indefinitely, it makes sense to set the *--time =* parameter to limit the test run time.

The attack stops at the end of this time, even if the loops configured via the *--reps =* parameter have not been completed.

After finishing its task, Siege shows a summary of the results (see Figure 1). ■

**INFO**

[1]   Siege: *http://www.joedog.org/ JoeDog/Siege*

**THE 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, fresh water aquariums, and learning Japanese, respectively.