

Siege Frequently Asked Questions

By [jdfulmer](#) | Published January 30, 2012 | [5 Comments](#)

- [Why do I need it?](#)
- [Where is the latest version?](#)
- [What platforms does it run on?](#)
- [Hey! I tried running it, but it doesn't work...](#)
- [What's the Bull Dog's name?](#)
- [Can I stress cgi-bin programs?](#)
- [Does siege support HTTP/1.1 protocol?](#)
- [I chose HTTP/1.0 protocol but Siege is reporting 1.1](#)
- [If Siege supports HTTP/1.1 protocol why would I choose 1.0?](#)
- [Does siege support the POST directive?](#)
- [Why doesn't my POST work?](#)
- [Why is POST data smaller than the size of my post file?](#)
- [Does siege support HTTPS protocol?](#)
- [Is the siege HTTPS support secure?](#)
- [Can I run siege for a pre-determined period of time?](#)
- [How many concurrent users should I use?](#)
- [What does "Error: system resources exhausted" mean?](#)
- [How Does Siege Calculate Concurrency?](#)
- [Does siege support cookies?](#)
- [Can I run siege with multiple resource files?](#)
- [Can I run siege with multiple IP addresses from the same machine?](#)
- [What is the easiest way to build a urls.txt file?](#)
- [How can I be notified when a new release is available?](#)
- [Where is the license information?](#)
- [Are siege binaries available?](#)

Why Do I Need It? ▲

Siege was designed to stress test and benchmark web servers. You need it to examine the performance and efficiency of your code under duress. Siege provides mechanisms to control the number of simulated hits on a web server over a controlled period of time. Use siege to save yourself future embarrassment.

Where is the latest version? ▲

The latest version of siege is available via anonymous ftp from <http://www.joedog.org/pub/siege>. Links are available to the mirrors on the top of every frame. There are two links to the Austin, TX site. The first is FTP and the second is HTTP for those behind oppressive, non-user friendly firewalls.

What platforms does it run on? ▲

Siege has been compiled and run on AIX 4.x, GNU/Linux 2.2.x, HP-UX 11.x, and Solaris 2.x. Siege relies on UNIX-centric functionality. Due to this fact, it will not run natively on Microsoft Windows. No problem. You can run it under [cygwin](#). Here's a "[cheat sheet](#)" to help get you started on Cygwin. It's a quick reference to help get you started.

Hey! I tried running it, but it doesn't work... ▲

'man siege' is a good place to start. You probably have a configuration problem. Run "siege -C" to view your current working configuration. If the output of that command indicates that you do not have a resource file, .siegerc, then run "siege.config" to generate one. You can configure siege so that it will run properly by either setting command line options or by editing the resource file.

What's the Bull Dog's name? ▲

The original spokes model for <http://www.joedog.org> was Limey Tango Jones. Sadly, he passed away in 2008. He was nearly thirteen years old. He left some mighty big shoes to fill ... The newest spokes model is Pommie Tango Jones. He never met a tennis ball he didn't like. If YOU have an English bulldog, please send us a link to his pictures. (We know you have some...)

Can I stress cgi-bin programs? ▲

Yes. If they accept GET, you can just build your URLs with the form values like this:

<http://www.hjs.com/cgi-bin/doh.cgi?name=homer&last=simpson>

If they accept POST, then you have to construct your URL with the POST directive like this:

www.bart.com/cgi-bin/aye.pl POST name=bart&last=simpson

If you are using the POST directive at the command line, you will have to put quotes ("") around it. The quotes are not necessary if the URL is in the urls.txt file.

Does siege support HTTP/1.1 protocol? ▲

Mostly. Support for protocol 1.1 has trickled in gradually. Starting with v2.03, an option was added to .siegerc to enable 1.1 protocol. Overtime, features of the protocol have gradually slipped into the program. The default protocol is now 1.1. To switch to 1.0 protocol make sure this directive is in your .siegerc file: protocol = HTTP/1.0.

Features of 1.1 protocol not yet supported by siege:

- Pipelined connections
- 100 Continue
- Support for persistent connections is poor. Siege prefers Connection: close

I chose HTTP/1.0 protocol but Siege is reporting 1.1 ▲

Several HTTP/1.1 compliant servers return headers that read HTTP/1.1 even though the client requested HTTP/1.0. Don't worry, the server is executing the selected protocol despite the fact that it is reporting otherwise. (Bug? Feature? Who knows?)

If Siege supports HTTP/1.1 protocol why would I choose 1.0? ▲

There are numerous reasons, actually. Perhaps the biggest reason for choosing 1.0 is to avoid skewing benchmarks. Some HTTP servers IGNORE the client's close directive on a redirect. If this occurs, the siege instance will "hang" with an open socket until the server finally closes it. This could take up to 30 seconds, a number that obviously skews your benchmark. If you notice siege instances hanging for extended periods of time, you probably want to switch to 1.0 protocol. For the most part, you probably want to use 1.1

Does siege support the POST directive ▲

Yes. As of siege-2.06 the POST directive is supported. You will need to construct your URLs in the following manner:

www.lisa.com/cgi-bin/haha.cgi POST first=lisa&last=simpson
<http://www.whooohoo.com/Auth> POST user=home&password=secret

Siege 2.65 added support for posting file contents. To post a file, use the redirect character:

<http://www.haha.com/reader.php> POST < /path/to/file

NOTE: this is not a multi-part file POST. Siege does not currently support that feature. Versions prior to 2.67 sent "application/x-www-form-urlencoded" as the default Content-type. Newer versions map the file extension to the appropriate content-type. In the example above, siege will send "Content-type: text/xml". The complete map is in src/load.c.

If you run POSTS from the command line, make sure you wrap the URL with quotes.

```
siege -g "http://www.haha.com/reader.php" POST user=howard&pass=stern"
```

DO NOT quote URLs in the urls.txt file!!!!

Why doesn't my POST work? ▲

The most common mistake we see occurs when people post to the file instead of the FORM action. In other words, if haha.jsp contains a FORM whose action is the servlet UserAuth, then you must construct your URL to post to UserAuth, NOT haha.jsp.

Why does siege POST data that is smaller than the size of my post file? ▲

Siege chomps excessive white space, carriage returns and newlines from the POST data contained within a post file that has been fed to the application with a redirect character: "<http://www.joedog.org/haha.php> POST < /my/file.txt" Therefore it is possible that siege will post less data to the server than it has read from the file.

Does siege support HTTPS protocol? ▲

Yes it does. But it requires additional work on your part. You have to download openssl from <http://www.openssl.org> and compile it and install it on your system. You will have to reconfigure and rebuild siege to use the openssl libraries. Consult the installation instructions for more details. In a nutshell, if you have ssl installed, then build siege like this: `./configure --prefix=/some/dir --with-ssl=/path/to/ssl make make install`

Is the siege HTTPS support secure? ▲

It is not as secure as it could be. Not all systems have a /dev/random and I've chosen to seed the entropy pool in the highly portable manner. Technically the method used is not as secure as it could have been. But siege was developed as a pre-production testing tool. It is not intended to handle sensitive data transactions. Use siege to test your https code in development, not in production.

Can I run siege for a pre-determined period of time? ▲

Yes. The frequency of this question has waned which I hope means that this option is no longer confusing. The format for time based testing is this: -tNUMm where NUM is an integer value and "m" is the modifier H, M, or S for hours minutes and seconds. Therefore if you want to run siege for two hours you could invoke it in any of the following ways:

```
-t2H -t120M -t7200S
```

NOTE: The modifier is NOT case sensitive, therefore -t2H is the same as -t2h.

How many concurrent users should I use? ▲

Integrity testing: I can count on one hand the number of times I found a problem with 800 simulated users that I wouldn't have found with 25. If your programmers aren't reclaiming resources such as database connections, you will identify those problems with repetitive hits to the troublesome application. Do yourself a favor, during integrity testing, keep the users under 100. You're more likely to create a mess than identify a problem. Capacity testing: Regression testing is a good method to measure your web server's load capacity. [Bombard](#) is a siege wrapper that stair steps increasing amounts of load and charts the results. Work the number of concurrent users up *slowly*.

Too frequently, we receive gripes from people who've made a mess with high levels of concurrent users. If you schedule more users than your webserver is configured to handle, then of course you'll make a mess. An HTTP server handles a fixed number of incoming requests. A pre-forking server has a limited number of child processes (MaxClients) and a threaded server has a fixed number of threads (MaxThreads). A server can handle up to its maximum number of handlers, no more. Once its handlers are exhausted, socket connections are queued by the OS. They wait for the next available handler. If the httpd server is overwhelmed by more users than it can handle, sockets will timeout before they ever get a handler. Siege reports a timeout, a browser reports "Connection refused." If the server is completely inundated, the OS may not have enough resources to even make a socket. If you scheduled so many siege clients that you've reached this point, then you've made a complete mess. Congratulations. Please don't email us with complaints.

If you want to measure your code under load that reflects your site's traffic, then you should schedule concurrent users based on current traffic. An access log analyzer like webalizer (free) or WebTrends (expensive) can help you determine that load. If your site peaks at 100 hits every five seconds, then use these parameters to invoke siege: -c100 -d5. For 50 hits a second you would run it like this: -c50 -d1. We've had people ask about configurations like -c1000 -d1. A single pre-forking apache server cannot handle such load without modification. It has a hardcoded limit of 256.

What does "Error: system resources exhausted" mean? ▲

This error occurs when you've selected a number of simulated users which is greater than your system can handle, i.e., you've run out of memory. The other likely scenario is that you've selected a number of simulated users which is greater than the number of processes your UNIX account is allowed to run. This is likely when a non-root user attempts to run more than 256 siege users on most commercial UNIX systems. You can run siege as root to get around this problem or you can have your administrator increase the number of processes you are allowed to run. Keep in mind that -c256 -d1 translates to roughly 47,000 unique visitors per day.

How Does Siege Calculate Concurrency? ▲

Concurrency is the total transactions divided by total elapsed time. So if we completed 100 transactions in 10 seconds, our concurrency was 10.00. This [blog entry](#) provides more details which should help you understand this metric.

Does siege support cookies? ▲

Netscape cookie support was added to siege-2.00; it does not support RFC 2965 cookies. This feature was "Microsofted" in that it shipped with a known issue: siege ignores path information and returns cookies based on server and domain. All cookies will be discarded after the run, they are not stored locally on disk. Cookies will be expired during the run as per the Netscape expire directive. The primary concern with regard to cookie support was to maintain state with the server. Siege supports that feature nicely.

Can I run siege with multiple resource files? ▲

Yes. The siege resource file is located in your home directory. It is called .siegerc. If you don't have a resource file, you can run "siege.config" to generate one. This file contains a runtime configuration. You may run siege with an alternative resource file in one of several ways, with a command line argument or an environment variable.

At run time siege searches for a resource file in the following manner. It checks to see if one was named with a "-R filename" argument. If that argument was not invoked, siege will check for a SIEGERC environment variable. Finally it will rely on a \$HOME/.siegerc file. You may set an environment variable like so: export SIEGERC=/home/jeff/etc/siege.conf

To make sure siege is using that file, run "siege -C" to view the configuration. You can set up several sieges with shell wrappers:

```
1. !/bin/sh
```

```
export SIEGERC=/home/jeff/etc/siege_one.conf /home/jeff/bin/siege
```

```
export SIEGERC=/home/jeff/etc/siege_two.conf /home/jeff/bin/siege
```

```
export SIEGERC=/home/jeff/etc/siege_thr.conf /home/jeff/bin/siege
```

Can I run siege with multiple IP addresses from the same machine? ▲

Yes. The best solution we've found comes to us from Robert Hartman although it is GNU/Linux specific and it requires IP tables support. Basically there are two steps.

1.) Add IP aliases. Example:

```
1. !/bin/sh
```

```
for i in `seq 1 254` do
```

```
    ifconfig eth0:$i 192.168.1.$i;
```

```
done
```

2.) Reverse NAT with iptables. So that the Linux kernel acts as a client from more than one IP address use iptables to do reverse natting. Example:

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 192.168.1.1-192.168.1.254
```

NOTES: This method can be used for Class B address spaces as well with proper masking on the eth0 interface. Robert tested this to work with 2500+ IPs on a single ethernet card. You can contact Robert via email: "robert at roberthartman dot net"

What is the easiest way to build a urls.txt file? ▲

There are two supported utilities that can help you build a urls.txt file. One is [sproxy](#) which is a functional HTTP proxy server that collects URLs including POST and GET data in a "siege-friendly" format and a second is [logparse](#) which parses URLs from an apache-style access_log and builds a siege-style urls.txt file. NOTE: logparse will build one URL for each entry in the access_log. It is recommended that you run logparse on a subset of access_log. Another unsupported utility is available. Scout surveys a webserver and prepares the urls.txt file for a siege.

How can I be notified when a new release is available? ▲

The simplest way to do this is to subscribe to updates on freecode. Click [here](#) and choose "subscribe" in the project menu.

Where is the license information? ▲

Siege is published under the GPL, the GNU Public License. A copy of that license is included in every siege distribution in the file called "COPYING." Click [here](#) for more information about the GPL.

Are siege binaries available? ▲

Yes. On Ubuntu systems connected to the Internet, you should be able to run "apt-get install siege" to install the software. Debian also makes siege available in its networked repositories. AIX binaries are available from Public Domain Software Library for AIX. Less recent binaries are also distributed for Solaris 2.5.1, 2.6, 7, 8/SPARC; HP-UX 10.20, 11.00; Tru64 UNIX 4.0D, 5.1; IRIX 6.5 and AIX 4.3.2 by The Written Word.