

Forum:[Hack Forums](#)

Thread: [Fastest way to spam http requests](#)

siwa - June 9, 2023, 5:34 am

I've looked into Go (fasthttp lib) and Rust (reqwests crate) for sending http requests and surprisingly Go was 3x faster (possibly due to Goroutines being faster than Rust threads). I pretty much need to spam the same exact request concurrently as much as possible and wait for a specific status code. What is the fastest way to do this, im guessing sockets in C would be the way to go but i am not sure, any ideas?

Ducky - June 11, 2023, 7:44 am

Have you tried libcurl with C++?

Hackers - June 12, 2023, 6:18 am

I have a feeling Go would be faster, but Python isn't slow! The C langs will probably be fastest because most other langs are just wrappers or rewrites of a C anyways, but with more bloat.

Sounds like you are trying to either hit a rate limit or resource maximum. If you are pentesting your own software/site, that's fine - You should look into multi threading though, or having multiple copies of same script running at same time. A lot of times your actual core server is capable of doing more network requests than any 1 script will do. If you run the same script multiple times, that's often a better way to increase the requests per second than swapping languages, but it depends.

Doug Dimmadome - March 14, 2024, 10:28 am

Do you need to make an HTTPS connection or simply send an HTTP payload? If just HTTP, the fastest way would be to simply write out the HTTP payload in plain text, read it from a file and cache it in a variable, and then spin up a bunch of threads and use async IO to send it over a socket to the server. There's no need to use a full-fledged library if you don't want to do anything with the response and the goal is just throughput.

Hackers - March 14, 2024, 11:14 am

(Jun 11, 2023 01:48 AM)phantomman55 Wrote: [\[CITE\]](#) Are you trying to carry out a dos attack? it sounds more like they're trying to trigger a timing attack by sending the same request at same exact second to get something twice, or write to a file twice (if not locked) and cause a reset, or similar. Could obviously be a DOS but for some reason a time-based attack on a specific request route feels like it could be the reason here.