

Forum: [Hack Forums](#)

Thread: [What is the easiest way to produce 10k+ junk instructions from c++ code for stalling?](#)

Tr0jandog - October 2, 2019, 8:46 am

For doing stalling code, one way is to just use a for loop

but for example a for loop with 10k iteration doesn't generate 10k instructions and it is just based on inside of it

so what is the easiest way to actually make my stalling code produce 10k instructions other than copying some simple lines of codes thousands of times? any better way to do this?

BASICALLY i want to produce 10,000+ junk instructions, what is the easiest way to achieve this?

and yes i know there are better ways for stalling than this but i'm focused on this method right now

HFDone - October 2, 2019, 11:50 am

http://www.seclab.tuwien.ac.at/papers/acm_ccs11_hasten.pdf

binbus - October 6, 2019, 3:15 pm

There is a public AI project that can generate code based off of open github repos

themayonnaise - October 6, 2019, 3:47 pm

I've never really attempted any stalling, so forgive me if this is a silly suggestion. If you want to reproduce 10,000 of the same instruction in the disassembly, you could force it with preprocessor recursion:

PHP Code

```
<?php
#define R10(x) x;x;x;x;x;x;x;x;x
#define R100(x) R10(R10(x))
#define R10K(x) R100(R100(x))
```

```
R10K(test_function());
```

would for example, give

PHP Code

```
<?php
0030229E call    test_function(030145Bh)
003022A3 call    test_function(030145Bh)
003022A8 call    test_function(030145Bh)
003022AD call    test_function(030145Bh)
003022B2 call    test_function(030145Bh)
003022B7 call    test_function(030145Bh)
```

```
003022BC call    test_function (030145Bh)
003022C1 call    test_function (030145Bh)
003022C6 call    test_function (030145Bh)
003022CB call    test_function (030145Bh)
003022D0 call    test_function (030145Bh)
003022D5 call    test_function (030145Bh)
003022DA call    test_function (030145Bh)
003022DF call    test_function (030145Bh)
003022E4 call    test_function (030145Bh)
003022E9 call    test_function (030145Bh)
003022EE call    test_function (030145Bh)
003022F3 call    test_function (030145Bh)
```

...

But I'm not sure if this is of any use to you. Perhaps some of the other suggestions are the way to go! :)