Forum:Hack Forums

Thread: forking off a new process/executing code from memory

Faith Connors - *December 15, 2018, 4:39 pm*

okay so

i know in C you can fork a program off to create two processes, something like this:
Code int main() {
```
    pid_t pid = fork();
    if(pid==0){
        ChildProcess();
    } else {
        ParentProcess();
    }
}

void ChildProcess() {
    kill(getppid(), SIGKILL)
}

void ParentProcess() {
    sleep(5)
}
```
thus resulting in the child process becoming the new sole process. this is a super basic example with no actual application.

but what if i want to do this in java?

is there an easy fork() function that can let me do this?

essentially what i want to accomplish is write a byte array stream to memory (the easy part) then create a new forked process like this (NOT A THREAD, an entirely new process with unique process ID and everything) then execute that byte array like shellcode (something similar to whats documented here) from within that forked child process and killing the parent process (or maybe skip the parent killing part and just make it look like a cool application or something.)

i know i can just run something using runtime.exec() or whatever, done that already. problem is, that relies on a file. i want to write that byte array stream directly from memory as i download the stream from some random site. hence, i want to be able to do everything from memory, and shellcode or whatever.

so are there any easy-to-use methods or classes that i can use to handle this? ive asked around a bit already and some people have told me to take a look at invokers/reflection methods, or something named 'classconstructor' or 'classinvoker' or 'classexecuter' or something like that, it started with 'class-'.

pls help tyty <333

SIM-429621 - *December 16, 2018, 4:47 am*

Stick to C/C++, Java is not meant for this.

Vanagander - *December 22, 2018, 5:33 pm*

Skimmed through the documentation so might not be what you're after but check out ProcessBuilder