**Safeexec docomentation**

**What is safeexec?**

Safeexec (safe execution environment) is a general purpouse lightweight sandbox[[1]](#footnote-2) for safely executing user programs.

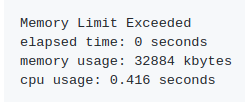
**How does this program works?**

Safeexec forks a child process setting limitations on that process via *setrlimit* so, safeexec should have a setuid bit on and a uid root. Then, the parent process polls memory usage looking into */proc* file system on linux or via *kvm* on freebsd. After that, the wall clock is enforced with *alarm()* & *signal()*. In the end, the reports usage stats to *stdout* or file specified by –*usage* argument.

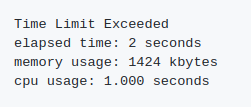
**Resource usage report format**

There are three (3) resource usage which are in four (4) line plain text, each.

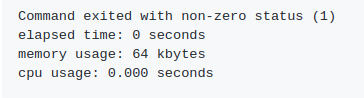
For memory usage:



For time limit:



For write attempt:



**Memory limits**

You can set strict memory limitations for languages like C/C++ whereas, for interpreted languages, like python; ruby; java; etc, instead of setting strict memory limitations, you can give then enough memory so your language is able to load.

**Unpriviliged *uid* range**

Since safeexec runs users programs as a random **uid** where the default **gid/uid** is between 5000 and 65535 you need to check if there is any **uid** in this range being used. If there is any **uid** in this range, then you can change it with the arguments –*minuid* and *maxuid*. The whole user range is treated as “unpriviliged” (“other”) for all files.

### **Network access**

While on Freebsd network access is blocked by setting on socket buffer size, on linux this does not happen. If you pretend for that to happen than, you must use other tools, like iptables, etc, for network access blockage.

**Number of open files**

The maximum number of open files is not limited because virtual machines (vm) languages, like java, or interpreted languages, like ruby or python, open a lots of files.

**Chroot**

From ubuntu’s [BasicChroot](http://help.ubuntu.com/community/BasicChroot) page:

<<A chroot is basically a special directory on your computer which prevents applications, if run from inside that directory, from accessing files outside the directory. In many ways, a chroot is like installing another operating system inside your existing operating system. Technically-speaking, chroot temporarily changes the root directory (which is normally /) to the chroot directory (for example, /var/chroot). As the root directory is the top of the filesystem hierarchy, applications are unable to access directories higher up than the root directory, and so are isolated from the rest of the system. This prevents applications inside the chroot from interfering with files elsewhere on your computer.>>

You can use safeexec without chroot if your OS doesn't allow arbitrary users to write filesystems.

1. Sandbox is like a antivirus although, instead of trying to block the virus and such out of the system, the sandbox environment set limits on system resources like memory, CPU time effectively preventing intended or unintended breach of system. However, running untrusted code on your server should be taken at your own risk. [↑](#footnote-ref-2)