Running jmap getting Unable to open socket file

Asked 5 years, 11 months ago Active 1 year, 10 months ago Viewed 56k times

84

46

I had to run `jmap` in order to take heap dump of my process. but `jvm` returned:

```
Unable to open socket file: target process not responding or HotSpot VM not loaded
The -F option can be used when the target process is not responding
```

So I used the `-F`:

```
./jmap -F -dump:format=b,file=heap.bin 10330
Attaching to process ID 10331, please wait...
Debugger attached successfully.
Server compiler detected.
JVM version is 24.51-b03
Dumping heap to heap.bin ...
```

1. Using `-F` is allright for taking heap dump?
2. I am waiting 20 minutes and not finished yet. Any ideas why?

java linux jvm jvm-hotspot

edited Jul 20 '15 at 7:30
jezrael
542k 47 701 721

asked Oct 1 '14 at 11:20
rayman
17.4k 38 128 226

4 Answers

Active Oldest Votes

176

+50

`jmap` vs. `jmap -F`, as well as `jstack` vs. `jstack -F` use completely different mechanisms to communcate with the target JVM.

jmap / jstack

When run without `-F` these tools use Dynamic Attach Mechanism. This works as follows.

1. Before connecting to Java process 1234, `jmap` creates a file `.attach_pid1234` at the working directory of the target process or at `/tmp`.
2. Then `jmap` sends `SIGQUIT` to the target process. When JVM catches the signal and finds `.attach_pid1234`, it starts `AttachListener` thread.
3. `AttachListener` thread creates UNIX domain socket `/tmp/.java_pid1234` to listen to commands from external tools.
4. For security reasons when a connection (from `jmap`) is accepted, JVM verifies that credentials of the socket peer are equal to `euid` and `egid` of JVM process. That's why `jmap` will not work if run by different user (even by root).
5. `jmap` connects to the socket, and sends `dumpheap` command.

only at safepoints. If a safepoint cannot be reached (e.g. the process is hung, not responding, or a long GC is in progress), `jmap` will timeout and fail.

Let's summarize the benefits and the drawbacks of Dynamic Attach.

Pros.

- Heap dump and other operations are run collaboratively by JVM at the maximum speed.
- You can use any version of `jmap` or `jstack` to connect to any other version of JVM.

Cons.

- The tool should be run by the same user (`euid` / `egid`) as the target JVM.
- Can be used only on live and healthy JVM.
- Will not work if the target JVM is started with `-XX:+DisableAttachMechanism` .

jmap -F / jstack -F

When run with `-F` the tools switch to special mode that features HotSpot Serviceability Agent. In this mode the target process is frozen; the tools read its memory via OS debugging facilities, namely, `ptrace` on Linux.

1. `jmap -F` invokes `PTRACE_ATTACH` on the target JVM. The target process is unconditionally suspended in response to `SIGSTOP` signal.
2. The tool reads JVM memory using `PTRACE_PEEKDATA` . `ptrace` can read only one word at a time, so too many calls required to read the large heap of the target process. This is very and very slow.
3. The tool reconstructs JVM internal structures based on the knowledge of the particular JVM version. Since different versions of JVM have different memory layout, `-F` mode works only if `jmap` comes from the same JDK as the target Java process.
4. The tool creates heap dump itself and then resumes the target process.

Pros.

- No cooperation from target JVM is required. Can be used even on a hung process.
- `ptrace` works whenever OS-level privileges are enough. E.g. `root` can dump processes of all other users.

Cons.

- Very slow for large heaps.
- The tool and the target process should be from the same version of JDK.
- The safepoint is not guaranteed when the tool attaches in forced mode. Though `jmap` tries to handle all special cases, sometimes it may happen that target JVM is not in a consistent state.

Note

There is a faster way to take heap dumps in forced mode. First, create a coredump with `gcore` , then run `jmap` over

edited May 23 '17 at 12:18
Community ♦
1 1

answered Mar 12 '16 at 20:43
apangin
70.5k 9 148 181

83

I just found that jmap (and presumably jvisualvm when using it to generate a heap dump) enforces that the user running jmap must be the same user running the process attempting to be dumped.

in my case the jvm i want a heap dump for is being run by linux user "jboss". so where `sudo jmap -dump:file.bin <pid>` was reporting "Unable to open socket:", i was able to grab my heap dump using:

```
sudo -u jboss jmap -dump:file.bin <pid>
```

answered Dec 11 '14 at 3:39
ben_wing
941 6 4

I think it should be \-dump:file.bin <pid> as you need to escape the - when passing the parameter through from sudo into jmap. – adam Dec 28 '15 at 17:57

This is it! You need to sudo for jmap and jcmd too. – xtian May 23 '16 at 17:21

wow.. This actually worked. This should be the accepted answer – Lalit Rao Feb 9 '17 at 12:48

2

Just like ben_wing said, you can run with:

```
sudo -u jboss-as jmap -dump:file.bin <pid>
```

(in my case the user is `jboss-as`, but yours could be `jboss` or some other.)

But it was not enough, because **it asked me for a password** (`[sudo] password for ec2-user:`), although I could run `sudo` without prompting me for a password with other commands.

I found the solution here, and I just needed to add another `sudo` first:

```
sudo sudo -u jboss-as jmap -dump:file.bin <pid>
```

It works with other commands like `jcmd` and `jinfo` too.

answered May 31 '17 at 20:40
Lucas Basquerotto
3,526 2 25 42

Double `sudo` saves my day! – Sher10ck Aug 9 '19 at 17:20

2

If your application is runing as a systemd service.You should open service file that under `/usr/lib/systemd/system/` and named by your service name. Then check whether **privateTmp** attribute is true.

```
systemctl restart [servicename]
```

If you want runing jmap/jcmd before restart, you can make use of the execStop script in the service file. Just put command in it and to execute `systemctl stop [service name]`

answered Nov 4 '18 at 12:30
Simple
31 5

Before I updated the /usr/lib/systemd/system/elasticsearch.service, setting privateTmp to false, I got this error: Unable to open socket file: target process not responding or HotSpot VM not loaded - even though I was running jmap as the elasticsearch user – imdibiji May 10 '19 at 19:09