Installation

`criu` is an utility to checkpoint/restore a process tree. This page describes how to get CRIU binary on your box.

Contents

Installing from packages

Many distributions provide ready-to-use packages. If no, or the CRIU version you want is not yet there, you will need to get CRIU sources and compile it.

Obtaining CRIU sources

You can download the source code as a release tarball (https://download.openvz.org/criu/) or sync the git repository (https://github.com/checkpoint-restore/criu). If you plan to modify CRIU sources (e.g. to contribute the code back) the latter way is highly recommended. The latest and greatest sources are:

Tarball:	criu-3.15.tar.bz2
Version:	3.15 "Titanium Falcon"
Released:	03 Nov 2020
GIT tag:	v3.15

Installing build dependencies

Compiler and C Library

CRIU is mostly written in C and the build system is based on Makefiles. Thus just install standard `gcc` and `make` packages (on Debian use `build-essential (https://packages.debian.org/build-essential)`).

For building with 32bit tasks C/R support you will need `libc6-dev-i386, gcc-multilib` instead of `gcc`.

Cross-compilation for ARM is also possible.

Protocol Buffers

CRIU uses the Google Protocol Buffers (https://developers.google.com/protocol-buffers/) to read and write images. The `protoc` tool is used at build time and CRIU is linked with the `libprotobuf-c.so`. Also CRIT uses python bindings and the `descriptor.proto` file which typically provided by a distribution's protobuf development package.

RPM packages

```
protobuf protobuf-c protobuf-c-devel protobuf-compiler protobuf-devel
protobuf-python
```

Deb packages

```
libprotobuf-dev libprotobuf-c-dev protobuf-c-compiler protobuf-compiler
python-protobuf
```

Optionally, you may build protobuf from sources.

Other stuff

- `pkg-config` to check on build library dependencies.
- `python-ipaddress` is used by CRIT to pretty-print IP addresses and is also required by zdtm.py
- `libbsd-devel` (RPM) / `libbsd-dev` (DEB) If available, CRIU will be compiled with `setproctitle()` support and set verbose process titles on service workers.
- `iproute2` version 3.5.0 or higher is needed for dumping network namespaces. The latest one can be cloned from iproute2 (http://git.kernel.org/?p=linux/kernel/git/shemminger/iproute2.git;a=summary). It should be compiled and a path to ip set as the `CR_IP_TOOL` variable
- `nftables` (RPM) / `libnftables-dev` (DEB) If available, CRIU will be compiled with nftables C/R support
- `libcap-devel` (RPM) / `libcap-dev` (DEB)
- `libnet-devel libnl3-devel` (RPM) / `libnet1-dev` (DEB) / `libnl-3-dev libnet-dev` (Ubuntu)
- `libaio-devel` (RPM) / `libaio-dev` (DEB) is needed to run tests
- `python2-future` or `python3-future` is now needed for zdtm.py tests launcher

For APT use the `--no-install-recommends` parameter is to avoid asciidoc pulling in a lot of dependencies. Also read about ZDTM test suite if you will run CRIU tests, those sources need other deps.

Building the tool

Simply run `make` in the CRIU source directory. This is the standard way, but there are some options available.

1. There's a docker-build target in Makefile which builds CRIU in Ubuntu Docker container. Just run `make docker-build` and that's it.

2. CRIU has functionality that is either optional or behaves differently depending on the kernel CRIU is running on. By default build process includes maximum of it, but this behavior can be changed.
3. You may specify build dependencies by hands

Installing

CRIU works perfectly even when run from the sources directory (with the `./criu/criu` command), but if you want to have in standard paths run `make install`. You may need to install `asciidoc` and `xmlto` packages to make install-man work.

Checking That It Works

Linux kernel v3.11 or newer is required, with some specific config options turned on. Various advanced CRIU features might require even newer kernel. So the first thing to do is to check the kernel by running `criu check`. At the end it should say "Looks OK", if it doesn't the messages on the screen explain what functionality is missing. If your distribution does not provide needed kernel, you might want to compile one yourself.

You can then try running the ZDTM Test Suite which sits in the `tests/zdtm/` directory.

Further reading

- Usage
- Advanced usage
- Category:HOWTO

Retrieved from "https://criu.org/index.php?title=Installation&oldid=5118"

This page was last edited on 24 December 2020, at 05:52.