Manticore, DeepState (and DeforaOS)

Pierre Pronchery <khorben@defora.org>

A few words about myself

- Debian user since 1999 (2.0 Hamm)
- IT-Security Consultant, freelance since 2007
- Official NetBSD developer since 2012
  - Kernel development
  - Security integration, interested in Reproducibility
  - Maintaining some packages (pkgsrc)
- Now on the board of directors there

Manticore

- Symbolic execution tool
- Uses a CPU emulator to run (and analyze) programs or algorithms on a simulated system
  - Input Generation: Manticore automatically generates inputs that trigger unique code paths
  - Crash Discovery: Manticore discovers inputs that crash programs via memory safety violations
  - Execution Tracing: Manticore records an instruction-level trace of execution for each generated input
  - Programmatic Interface: Manticore exposes programmatic access to its analysis engine via a Python API

Manticore

- Supports static Linux binaries (Intel 32- and 64-bits, ARM 32-bits) and Ethereum bytecode
- Official releases on GitHub: https://github.com/trailofbits/manticore
- Already packaged in pkgsrc by myself (NetBSD, Linux, macOS…)
- Volunteer for Debian?

DeepState

- Unit-testing for symbolic execution
- Supports Manticore and angr as backends
- Two packages: binaries, Python bindings
- On GitHub with no releases yet: https://github.com/trailofbits/deepstate
- Already packaged in pkgsrc-wip (not upstreamed yet)
- Volunteer for Debian?

DeforaOS

- Open Source, BSD- and GPL3-licensed software
- https://git.defora.org and mirrored on GitHub at https://github.com/DeforaOS/
- Desktop environment and some more
- I am the main developer
- Volunteer for Debian?

Debian developer?

- I’m here so I might as well get my key signed
- Other steps? (I heard there are plenty)