Meeting Minutes

2021-10-04:

* Milan:
  + Showed a Dead Store
* Nghia:
* Sudip:
  + CodeChecker issue reported

2021-09-20:

* Syzkaller Issues for investigation:
* <https://elisa-builder-00.iol.unh.edu/syzkaller/crash?id=5b0c829736bb0f315d9ba993726b66ff19b17a93>
* <https://elisa-builder-00.iol.unh.edu/syzkaller/crash?id=54bf5f782a406d45b90c547637a1094a454e762a>
* KCSAN:
  + Finds many concurrency bugs, but it is not looking for reproducers
* CodeChecker:
  + Two new dead stores in lib/mpi/ec (probably just config issue and needlessly complex code)
  + Needs further investigation
* Nghia:
  + Is busy this week.
* Jochen: no open points.
* Milan: no open points.
* Sudip: no open points.

2021-08-23:

* Newcomer progress:
  + Nghia will submit a clean-up patch.
* Syzkaller reports:
  + For now: keep as-is if not disturbing.
  + In future: store them and archive them?
    - Store reproducers in linux-arts <https://www.lkml.org/lkml/2019/10/15/756>
    - Or to: <https://github.com/dvyukov/syzkaller-repros>
* Syzkaller refinements:
  + New C reproducer for bug:
    - <https://elisa-builder-00.iol.unh.edu/syzkaller/report?id=54bf5f782a406d45b90c547637a1094a454e762a>
  + KCSAN could not be enabled yet, many bugs and no good reproducers.
  + Sudip will set up a separate instance only with KCSAN and then we can have a look.
* Dead Store reports:
  + futex: <https://lore.kernel.org/lkml/CAKXUXMzqmN1dYpbYSCXWN9VwHn8+MXj3P=G09qD6=atwrcJ8WA@mail.gmail.com/>
  + crypto: <https://lore.kernel.org/lkml/20210822103107.28974-1-lukas.bulwahn@gmail.com/>
  + ext4: <https://lore.kernel.org/lkml/20210820120853.23134-1-lukas.bulwahn@gmail.com/>
  + netfilter (after the meeting):

https://lore.kernel.org/lkml/CAKXUXMzdGdyQg9CXJ2AZStrBk3J10r5r=gyiAuU4WimnoQNyvA@mail.gmail.com/T/#u

* + Open:
    - Two new in fs.
    - Lukas to report.
* Mailing lists:
  + Let us introduce three mailing lists. We can add more when we touch on more topics.
* Server is at <https://elisa-builder-00.iol.unh.edu/>
* Welcome, Richard Wagener. Richard Wagener is listening...

2021-06-08:

* Meet on Monday, 16:00 CEST
* Syzkaller reproducers?
* Kernel configuration
* Update of CodeChecker?

2021-05-04:

* Ideas for Setup on the Build Machine:
  + Some cross-builds:
    - x86, arm, mips
  + Run some tests (LTP, kselftest, etc.):
    - With coverage measured (gcov), providing test reports with lcov (provides html reports).
    - Eli could provide instructions or even set it up.
    - Sudip&Lukas to get an account for Eli.
  + CodeChecker UI
    - Provide more hints in the UI (documentation task)
  + Setup of an own Elixir instance, for further extension to add
  + Setup of syzkaller for fuzzing
* Kernel configuration is relevant?

2021-01-19: Milan, Sudip, Lukas

* Discussion on documentation warnings (Lukas, Milan)
* Discussion on documentation clean-ups (Lukas, Milan)
* CI (Sudip):
  + We are still waiting for hardware, but we are getting closer.

2021-01-05:

* ECLAIR:
  + Had a second look at results.
* Helping contributors:
  + Milan:
    - On next-20200105 and current master:

kernel/stacktrace.c:137: warning: Function parameter or member 'tsk' not described in 'stack\_trace\_save\_tsk'

kernel/stacktrace.c:137: warning: Excess function parameter 'task' description in 'stack\_trace\_save\_tsk'

Find it with `make W=1 kernel/stacktrace.c`.

* Jochen: No news.
* Sudip: we are still waiting for hardware for our CI server.
* Lukas: Some investigations with fuzzing (will share results on Friday.)
* Presenting activities of group at next workshop:
  + Mission statement, goal
  + Investigations:
    - Open-source tools, Sudip’s CI
    - Roberto’s tool (ECLAIR)
  + Patch Submission Process:
    - Learning patch submissions
    - Feedback from patch submissions

2020-12-15: Milan, Roberto Bagnara, Sudip, Lukas

* Infrastructure Setup: Shuah reached out to OSUOSL (Oregon State University Open-Source Lab) to get machine access.
* Helping contributors:
  + Milan: is working on a documentation patch.
  + Jochen: found the reason for coccinelle installation issue. Sudip’s container for coccinelle works, though. Ubuntu 20.04 seems not to work.
* Kernel configuration:
  + AGL kernel configuration: requires too much computing power.
    - AGL configuration: <https://github.com/sudipm-mukherjee/linux-test/blob/linux/defconfig>
* Selinux:
  + Jochen showed his insights of the analysis.
* ECLAIR:
  + Looked at results together.

2020-12-01:

* Infrastructure setup: waiting for the feedback from Shuah
* Intel Kernel Test Robot: Oliver will present; Eli sent abstract.
* Klokwork:
  + Did not look into tracebacks.
* Helping contributors:
  + Jochen: Installation of coccinelle failed with latest OCaml version (4.07)
  + Error message:

“

/usr/lib/ocaml/pcre/pcre.cmi

is not a compiled interface for this version of OCaml.

It seems to be for a newer version of OCaml.

make: \*\*\* [Makefile:423: globals/regexp\_pcre.cmx] Error 2

“

* + - See docker image from Sudip: https://hub.docker.com/r/sudipm/cocci
  + Parth Dode:
    - A student interested in introduction to kernel work.
    - Steps: downloading the kernel, compiling the kernel, running clang-analyzer
* CI:
  + No update; CI functionality is very limited due to missing compute resources
* Kernel configuration:
  + Use configuration from the “AGL kernel”
    - Let us include AGL kernel configuration.
    - Provide statistics on findings for AGL and some open questions for the AGL group.
* First prototype of Codechecker exporter:
  + Exporter works; importer needs to be implemented now.

2020-11-24: Milan Lakhani, Eli Gurvitz, Sudip, Jochen Oscar, Lukas

* Infrastructure setup: waiting for the feedback from Shuah
* Intel Kernel Test Robot: Eli will invite Oliver to present Linux Kernel Performance Team’s robot (kernel test robot) at the next Workshop
* Klokwork: Eli sent results.
  + Two results we looked at were false positives.
  + Eli will check if he can provide the tracebacks from Klokwork.
* Helping contributors:
  + Milan:
    - Learning on staging & checkpatch patches, but ready to move on.
  + Jochen: currently not writing patches
* CI:
  + Maybe travis.ci will cut the “free” service for us.
  + “Ignore list” for coccinelle rules

2020-11-17: Milan Lakhani, Eli Gurvitz, Sudip, Jochen Oscar, Lukas

* Infrastructure setup: waiting for the feedback from Shuah
* Intel Kernel Test Robot: Eli will invite Oliver to present Linux Kernel Performance Team’s robot (kernel test robot)
* Klokwork: Eli will run Klokwork on linux-next
* Helping contributors:
  + Milan: send out his first patch.
    - Potential lists to CC:
      * linux-safety@elisa.tech
      * [linux-kernel-mentees@lists.linuxfoundation.org](mailto:linux-kernel-mentees@lists.linuxfoundation.org)
  + Jochen: Compiling the kernel with bitbake
    - Probably with ‘bitbake linux’, Sudip will follow up with Jochen.
  + Eli: playing around with compiling the kernel, working towards being able to send patches.
  + Oscar: no help needed.
  + Sanjay: no response.
* Potential work:
  + Static analysis findings
  + Broken MAINTAINERS entries
    - ./scripts/get\_maintainer.pl --self-test
* CI:
  + Sudip updated to latest master
  + cross-compilation support in CI for arm, mips

Participants:

2020-11-10:

Participants: Milan Lakhani, Eli Gurvitz, Sudip, Jochen, Lukas

* Infrastructure setup: waiting for the feedback from Shuah:
  + <https://aws.amazon.com/blogs/opensource/aws-promotional-credits-open-source-projects/>
* Intel Kernel Test Robot:
  + Oliver Sang is technical contact.
  + Which tools are we interested in?
    - Intel Performance Team to share their results with sparse and coccinelle; maybe include clang-analyzer for their builds;
  + Invite Oliver Sang to present all static analysis runs done by Intel kernel Test Robot (Linux Kernel Performance Team)
    - Continue discussion on what kind of sharing would help us to fix more and quicker.
* Klocwork:
  + Eli ran Klocwork on tinyconfig with 4.19.x
    - A few hundred findings
    - We hope that we can share
    - Titles and description of MISRA rules are copyrighted. Eli will share the MISRA identifiers (numbers).
    - Eli will provide a scan with a recent linux-next on tinyconfig.
* Helping contributors with their setup:
  + Milan: tools set up; just a few steps away from sending his first patch.
  + Jochen: still busy…
* Sudip’s CI:
  + Working smoothly.

2020-11-03:

Participants: Milan Lakhani, Eli Gurvitz, Jochen Kall, Sudip, Lukas

Agenda:

* ECLAIR Report:
  + Evaluation of report:
    - No critical findings.
    - Ran on next-20201029 and with tinyconfig
      * All mandatory MISRA rule, except one rule descoped.
  + Next steps of extending ECLAIR run:
    - Re-run with Klokwork for soundness/consistency.
    - Going for more rules to check on tinyconfig?
    - Going for a larger config?
      * Defconfig or defconfig without USB, DRM, and sound?
  + Testing patch proposals with ECLAIR?
    - Suggest to create service to test patches with ECLAIR.
* Infrastructure setup:
  + AWS is wasteful; too expensive
  + Recommendation to get root server or to get “cheap/free” compute power from Amazon?
    - Lukas to make our recommendation clear to Shuah, TSC and the Governance Board.
  + Reach out to Intel test robot Team? (<https://01.org/lkp/documentation/0-day-test-service>; formerly: Fengguang Wu, now: [oliver.sang@intel.com](mailto:oliver.sang@intel.com)?)
    - Eli Gurvitz will ask Intel internally.
* Ideas for required CodeChecker features and improvements:
  + Idea from Sudip:
    - User Authentication and User Permissions for specific operations (and more advanced workflows)
* Helping contributors with their setup:
  + No help needed.
* Any further points?
  + Compare the difference between two binaries
    - Lukas current approach: objdump -d .o file and diff the two objdump files.
    - Eli: make with parameters to dump .S files; compare .S files.
    - Sudip: diffoscope (diffoscope.org) allows to compare for differences on binary file.

2020-10-27:

Participants:

Agenda:

- If Roberto joins, new results of Roberto's run on the latest version:

Mainline:

* https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

Linux-next:

* https://git.kernel.org/pub/scm/linux/kernel/git/next/linux-next.git/
* Further processing:
  + git fetch
  + git checkout -f master
* Working with linux-next: https://www.kernel.org/doc/man-pages/linux-next.html

Consideration on continuous working mode:

* Access to running tool:
  + License is limited to use on the kernel source code.
  + Access must be limited to system administration.
* Publishing results:
  + Concern:
    - Capabilities of tool shall not be visible.
  + Restricting access to project participants
    - Sounds like it can be resolved.
    - Need to understand that: We want to share withKernel CI database (<https://github.com/kernelci/kcidb>)
* Infrastructure:
  + Compute power for running the tool?

When we create a patch based on the tool, we will acknowledge

- News on CI from Sudip:

- CI is good for now: just some stability issues.

- We need a proper server.

- Ideas for required CodeChecker features and improvements: No try out yet.

- Discussion: Infrastructure setup

- TSC proposed to simply reach out to Amazon.

- contact Amazon: Sudip through Debian community, we asked Greg KH.

- Greg KH had some compute power from Amazon in the past; he recommends getting a different setup than Amazon for kernel builds (if we need to pay for it).

- Helping contributors with their setup:

- Milan is making progress (wants his own CI).

- Jochen is setting up tools, then get the setup for sending patches.

2020-10-13:

* Jochen Kall
* Oscar Slotosch
* Artem Mygaiev
* Gabriele Paolini
* Sudip
* Lukas Bulwahn

Agenda:

- Roberto B.: New insights on uninitialized values and other MISRA rules

- Roberto did not join; postponed.

- Sudip: Status of current CI

- current config: defconfig without drm, usb, sound

- Discussion: relevant builds and configs to consider?

- configs are still unknown. We keep what we currently have.

- Discussion: Infrastructure setup

- keep it as is or get more compute power from some CI provider?

- based on a private free account at the moment.

- more reasonable storage than the current aws s3.

- for CodeChecker server, we would need a web server.

- Lukas: status of Dead Store assessment on tinyconfig

- Helping contributors with their setup:

- collect status on setup of group members

- some kernel compilation set up?

- running clang-analyzer set up?

- running coccinelle set up?

- running CodeChecker server set up?

- sending kernel patches set up?

2020-10-06:

Participants:

* Sudip
* Paul Albertella
* Lukas Bulwahn
* Jochen Kall
* Doris Wild
* Oscar Slotosch
* Gab Paolini

Agenda on Ongoing activities:

**- Revisit and extend mission statement:**

- Phase 1: Consideration of contribution to safety should not be a criteria now.

- Definition of the groups' work results:

- "Make tools happy" kernel patches

- Comments/assessments on findings from tools

- Collaborative setup and workflow for tool reports, assessment and patches

- Additions to kernel documentation explaining tools and agreed policies wrt. coding conventions, handling certain bug classes

- Actual "bug fix" patches (\*at this stage, we need to be careful, though.)

- Verification suite for our purposes of patch submission

- **Sketch Envisioned "Big Picture" Workflow:**

- Analysis and Tool Results:

- Run tools regularly and collect results.

- Shared Build infrastructure for these activity:

- Maybe just github with travis.ci?

- Maybe dedicated server infrastructure through ELISA LF Project?

- Store results of tool runs in kcidb.

- Triage finding:

- Obtain results of tool runs from kcidb

- Import these results into CodeChecker (or some other suitable tool)

- Collect comments in CodeChecker

- Export comments to kcidb (unclear at the moment.) to share with others.

- Import comments from others into CodeChecker

- Derived actions:

- Create clean-up patches

- Refine tool setup

- Create Statistics of Findings to serve as information

- Proposal: Understand and describe required extensions to CodeChecker

**- Helping contributors with their setup:**

- collect status on setup of group members

- some kernel compilation set up?

- running clang-analyzer set up?

- running coccinelle set up?

- sending kernel patches set up?

- clang-analyzer runs:

- quickly collect info on current clang-analyzer runs

- Sudip: running next with defconfig (not regularly)

- Lukas: running next-20200922 with tinyconfig

**- Continue on investigation on uninitialized values (Roberto B.)**

- Roberto still needs to share results

- can we get the results for x86-64 tinyconfig as well?

- just to have a reduced set of findings to start with.

- how to import into CodeChecker?

**- Proposal: Investigation of OASIS Static Analysis Results Interchange Format (SARIF)**

- <https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=sarif>

- Oscar will have a look.

Any other Business?

2020-09-22:

Participants:

* Lukas Bulwahn
* Jochen Kall
* Laurent Cremmer
* Roberto Bagnara
* Sudip

2020-09-15:

Participants:

* Shuah Khan
* Mohammed Billoo
* Jochen Kall
* Gabriele Paolini
* Laurent Cremmer
* Roberto Bagnara
* Roberto Paccapeli
* Sudip

Mission Statement:

* Focus on application of tools and handling the tool results, improving the kernel based on the tools’ feedback.
* No disagreement, no further discussion.

Cocci magic number:

* Mohammed Billoo showed the results: still many false positives, and still some false negatives

TODO/FIXME/HACK script:

* Still some work

CodeChecker:

* Lukas did not have time to share more information yet.
* Nobody installed the CodeChecker yet.

Shuah updates:

* Work on coccinelle.

Setup for kernel patches:

* Jochen is still working on his setup.
* Laurent is still fighting with the corporate approval for the setup.
* Coccinelle setup:
  + still difficult to set up.
* Docker setup for tool setup.
  + Some unmaintained docker setup:
    - https://github.com/bulwahn/linux-kernel-analysis

Shuah suggestion:

* Read and update documentation.

2020-09-08

Participants:

* Mohammed Billoo
* Patrick Probst
* Maurizio Iacaruso
* Roberto Paccapeli
* Wenhui Zhang
* Lukas Bulwahn

Agenda:

* Goal of the current discussion:
  + Focus on application of tools and handling the tool results, improving the kernel based on the tools’ feedback.
* Mohammed’s coccinelle contribution:
  + Based on Shuah’s CWE initial investigation
  + <https://lists.elisa.tech/g/linux-safety/message/4>
    - working on CWE-547
    - Mohammed tried it out on a few code examples, has not run it yet on the full kernel source code. (No time for that so far.)
    - The commercial tools we looked at (coverity) does not address this CWE.
    - **Mohammed: shares the output of this script on the mailing list.**
* CodeChecker investigation:
  + CodeChecker allows you to import results from tools.
    - Rationale: coccinelle, smatch, sparse, clang static analyser, clang-tidy do not provide any local UI interface and any WEB user interface.
    - CodeChecker provides a Web UI and hence distributed collaborative work.
    - Imports from multiple tools is supported, hence it could be a single UI for multiple tools.
  + What is the end goal?
    - Having a collaborative setup to track tool findings continuously on the latest kernel version for some relevant kernel configurations.
    - Towards that goals:
      * Which kernel configurations are really relevant?
      * Which kernel configuration should we start with?
        + Suggestion: x86-64 tinyconfig
      * What needs to be added to CodeChecker collaboratively with many people?
      * Support for kernel static analysis tools, coccinelle, sparse, smatch etc.
    - active project on github
    - Goal: agree on features and changes here first and then open pull request towards main CodeChecker repository
  + More information: https://www.slideshare.net/OliveraMilenkovic/codechecker-overview-nov-2019
  + Next steps:
    - Installation, setup, run some tools, import results, work with UI to learn how to use it.
    - **Lukas: provide first instructions on previous exploration**
* More tools:
  + Wenhui suggested looking at <https://cisofy.com/lynis/> and <https://www.sonarqube.org/>.
  + <https://cisofy.com/lynis/>
    - Open-source, GPLv2.
    - <https://github.com/CISOfy/Lynis>
    - Run-time penetration testing for kernel on the UNIX interfaces (glibc & kernel).
    - Requires an installed system; it does not work on a source code basis.
  + <https://www.sonarqube.org/>:
    - <https://github.com/SonarSource/sonarqube>
    - Open-source, [GNU Lesser General Public License, Version 3.0](https://www.gnu.org/licenses/lgpl.txt)
  + Next step:
    - Have a second look, try to install them, try to run them on the kernel.
    - If interesting, reach out to the tool developers and get a better understanding of the tools.