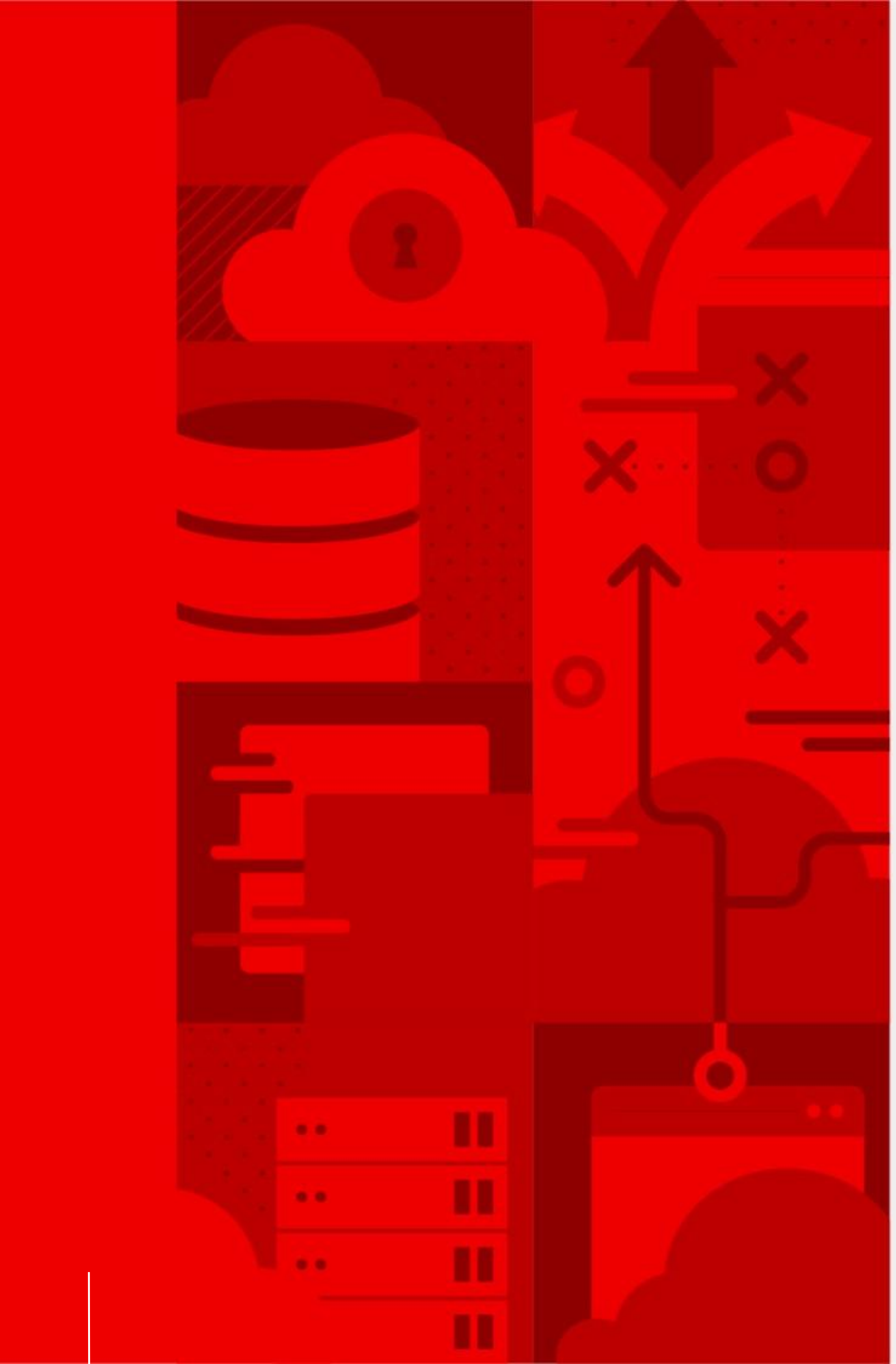


KS-NAV - What’s next

Alessandro Carminati - Principal Software Engineer

What is ks-nav?

A tool set designed to ease the Linux kernel architecture by reverse engineering kernel binary

- **Key Features:**
  - **Call Tree Diagrams**:
    - Generates diagrams showing the call trees of kernel functions.
  - **Subsystem Interactions**:
    - Illustrates how subsystems interact each other in a given function call.
  - **Interface Mapping**:
    - Maps interfaces between subsystems during specific calls.
  - **Global Data Sharing**:
    - Visualizes global data shared between functions.

Why it uses binary

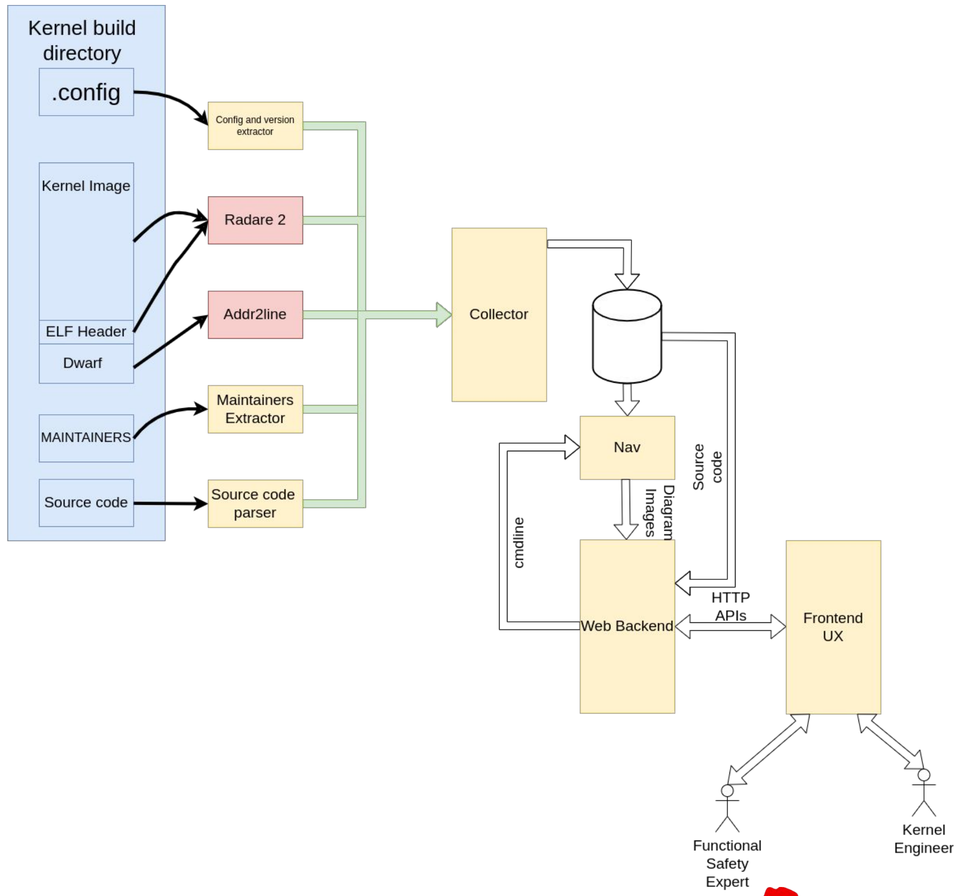
	Pros	Cons
Source code	+ Can be read by humans + Is immediately available	- Needs preprocessing: ○ C Macro alters the code ○ The files that are actually used are select by logic buried in the Makefile ○ Compiler optimizations changes the execution code. - Source code is not written in an homogeneous language
Binary	+ No preprocess is needed + It is homogeneous + There’s no extra code to consider	- Needs tools to be accessed - Need to correlate info from the binary analysis with info from debug information.

Benefits

- Simplifies the understanding of complex kernel interactions.
- Enhances debugging by clarifying call dependencies.

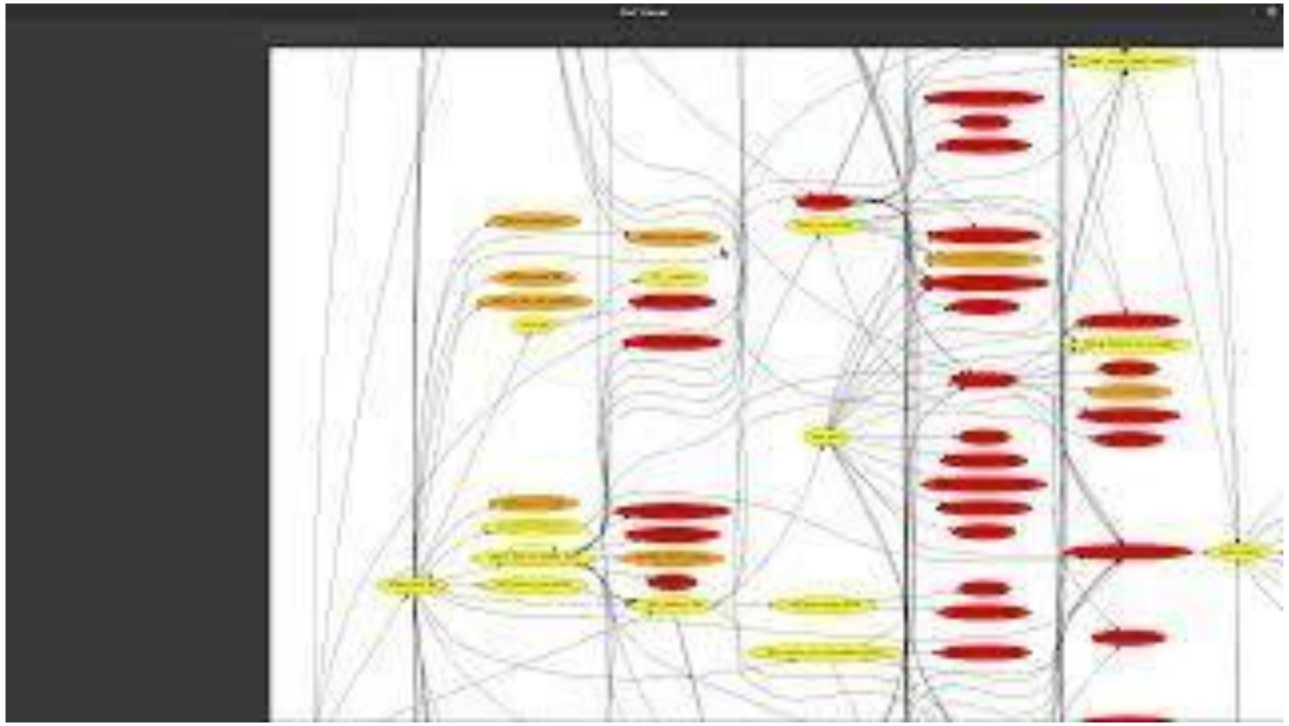
How does it work? - The Architecture

- Toolset
  - **kern_bin_db** - Produce data sets
  - **nav** - Draws diagrams
  - **navweb** - provide a web interface
- **kern_bin_db** uses **Postgres** DB to store data, **Binutils** and **Radare2** to extract info from the kernel image binary
- **nav** uses **Graphviz** library to produce diagrams
- **navweb** wraps the two tool to provide a basic web interface… in future the backend service for the full fledge interface



Basic features

- Call-tree view
- Subsystem tree view
- Subsystem interface view



Future developments

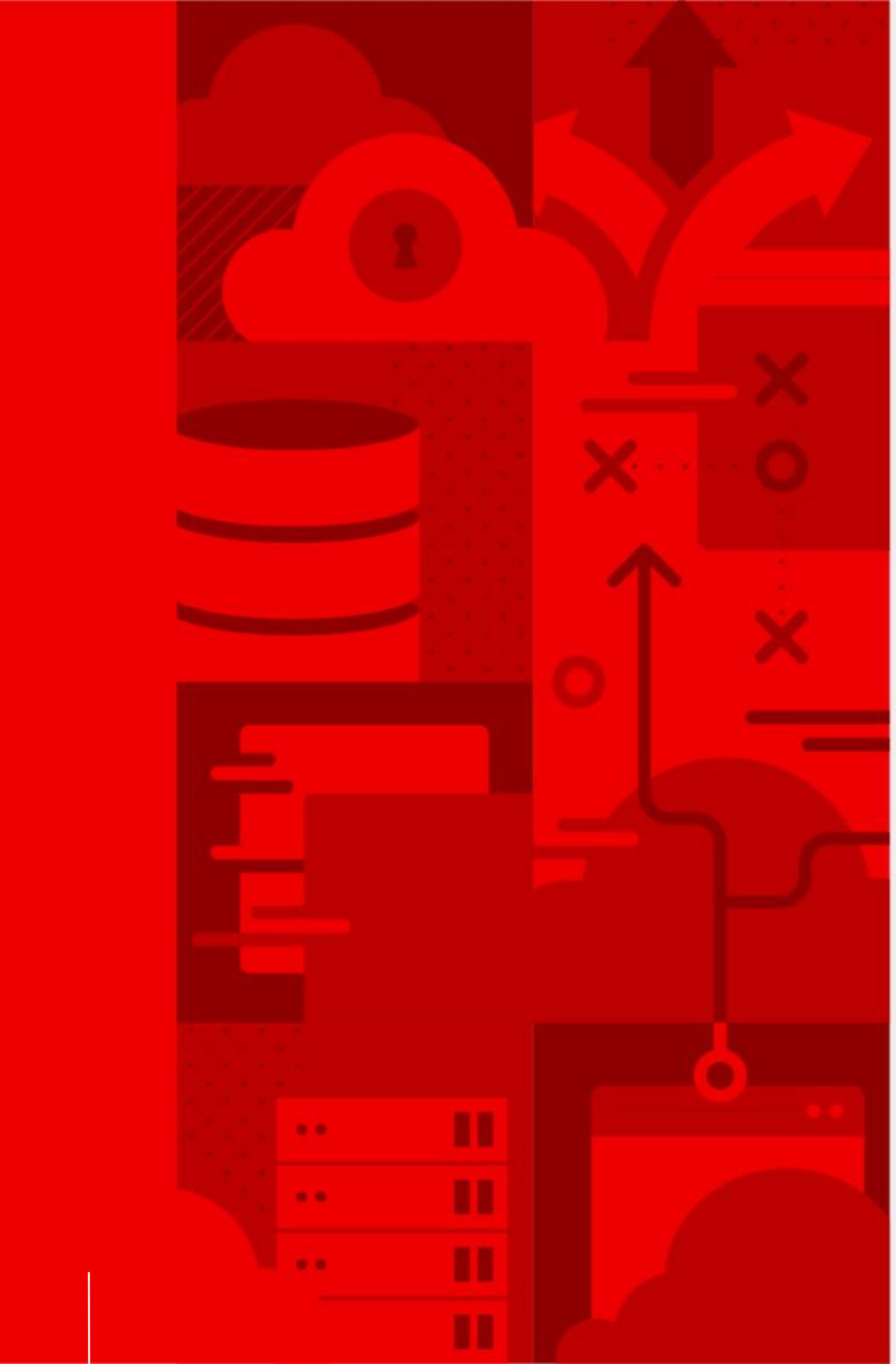
- Capacity to operate with git Repository.
- Capacity to automate builds.
- Extend the global variable sharing diagrams to subsystems.
- Include modules in the database
- Add more attributes to functions
- Resolve indirect calls.
- Build a new web interface that allows to browse kernel source code while seeing diagrams.
- Export Kernel data to graph database

Red Hat

Future developments

Contributing:
If you're interested in contributing, the project's public repository is hosted in the Elisa namespace. You can access it directly using the QR code on this slide. We welcome contributions from anyone

https://t.ly/ivT9g



Thank you

QA

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat