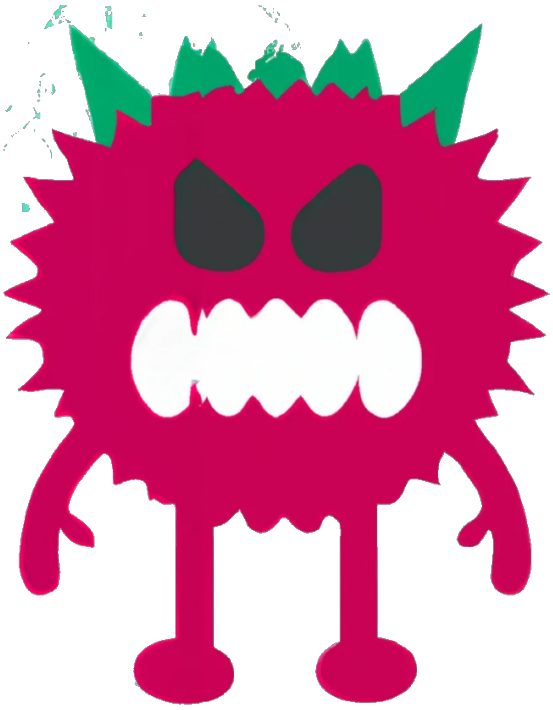


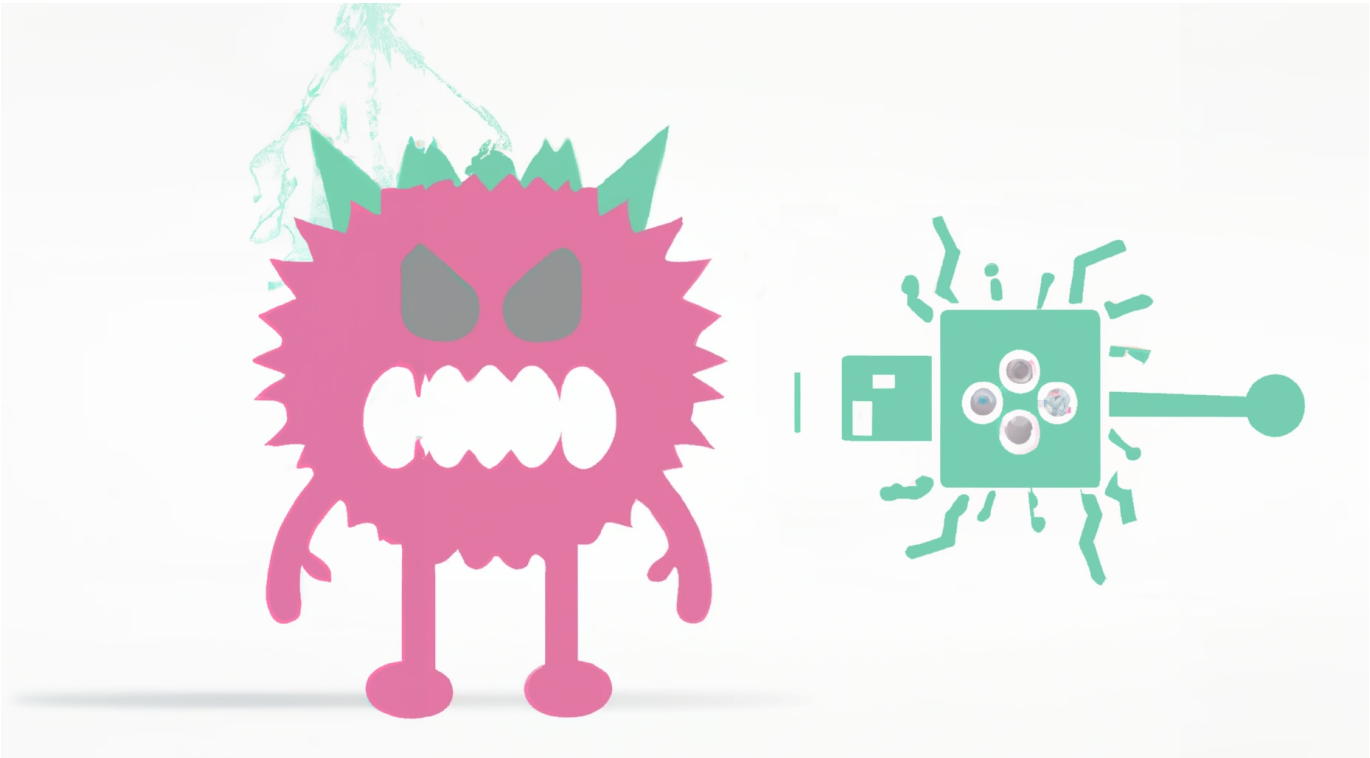
Raspirus docs

A simple hash-based virus scanner

Benjamin Demetz

Table of contents

1. HOME



1.1 Introduction

Raspirus: Empowering Your Malware Protection

Welcome to the official documentation for Raspirus, your lightweight signature-based malware scanner. Originally designed to scan attached USB drives using a Raspberry Pi, Raspirus has evolved into a versatile tool capable of scanning local files and folders as well. Some of its standout features include:

- **Cost-Free Protection:** Raspirus operates solely on donations, ensuring top-notch protection without any financial burden.
- **Tailored Signature Detection:** Our custom signature-based approach guarantees accurate identification of malware.
- **Comprehensive File Scans:** Raspirus can efficiently scan compressed files, ensuring no threat goes undetected.
- **Privacy Prioritized:** Offering a privacy-friendly option, Raspirus keeps your personal information secure.
- **Cross-Platform Convenience:** Enjoy the benefits of Raspirus protection on a variety of operating systems.
- **Swift and Dependable:** Count on Raspirus for fast and reliable malware detection.
- **Sleek Modern Interface:** With user-friendliness at its core, Raspirus boasts a beautiful and intuitive UI.

1.2 Getting Started

1.2.1 For Regular Users

Getting started with Raspirus is a breeze. Follow these simple steps:

1. Visit our website or head to the GitHub release page.
2. Download the executable that matches your operating system.

> **Note**
>
> If you plan to use Raspirus on the Raspberry Pi as a standalone application, we have a dedicated guide for it.

1.2.2 For Developers

Are you a developer looking to set up Raspirus? We've got you covered. Check out our comprehensive guides for various operating systems in the Developers section.

1.3 Questions?

Got questions about Raspirus? We're here to help!

- Visit our FAQ section for answers to common queries.
- Join our thriving community on the Discord server to engage with fellow users.
- If you've encountered a bug, browse the GitHub issues to see if it's already reported.

Thank you for choosing Raspirus for your malware protection needs. Together, we're making the digital world safer for everyone.

August 29, 2023

April 3, 2023

2. CONTRIBUTING

2.1 Enhance the Codebase

As a developer looking to amplify Raspirus's functionality, your contributions are invaluable. Please adhere to the following guidelines:

- **Check Existing Issues:** Before diving in, peruse existing issues to avoid creating duplicates. If none exist, feel free to open a new one.
- **Documentation is Key:** Always ensure your code is well-documented. Remember to adapt tests to maintain code integrity.
- Read the Code of Conduct

2.2 Enrich the Documentation

Raspirus's documentation is crafted using Markdown and powered by MkDocs and Python. To contribute to the documentation:

- **Follow the Guide:** To begin, set up the documentation on your local machine by following the guide.
- **Material Theme:** Our documentation utilizes the Material Theme, which extends Markdown functionality. Discover more about it here.

2.3 Drive Translations

Our documentation is multilingual thanks to Crowdin, an intuitive external service. GUI translations are handled via JSON files. Refer to the guide for comprehensive insights.

2.4 Infuse Artwork and Media

The repository for artwork (logos, banners, etc.) and media (powerpoints, articles, graphs, etc.) is hosted separately at this repository. Feel free to contribute and bolster our visual presence. Note that the current logo and banner were AI-generated due to the lack of artistic expertise.

2.5 Provide Feedback

Beyond the above contributions, there are other impactful ways to get involved:

- **Join Discord:** Engage with our community on the Discord server to share valuable feedback and ideas.
- **Consider Donating:** If you find value in Raspirus, consider donating to support ongoing development.
- **Spread the Word:** A simple yet effective way to contribute is by downloading Raspirus and introducing it to friends.

Thank you for your dedication to Raspirus's growth and improvement. Your involvement is a cornerstone of our success.

August 31, 2023

August 22, 2023

3. DEVELOPERS

3.1 Navigating the Architecture

```
graph LR
     A[Start] --> B{Scan location specified?};
     B --> |Yes| C[Start scan];
     C --> |Start Loop| D[File found];
     D --> E[Create Hash];
     E --> F[Compare Hash];
     F --> G{Hash found in DB?};
     G --> |Yes| H[Flag as Malware];
     G --> |No| I[Flag as Safe];
     H & I --> J[Continue iteration];
     J --> K{Last file?};
     K --> |Yes| L[Stop scanner];
     L --> M[Display Results];
     K --> |No| N[Start again];
     N --> D;
     B --> |No| O[Stop]
```

Raspirus is structured into two integral components: frontend and backend. These components, built using distinct languages and frameworks, are interconnected via a third-party framework called Tauri. This framework not only facilitates communication between the frontend and backend but also enables us to incorporate Rust functions into the frontend. Furthermore, Tauri empowers the distribution of Raspirus across various operating systems.

3.2 Starting Your Development Journey

Windows **Linux** **macOS**

1. Clone the repository
2. Install Tauri and Prerequisites
3. Install npm
4. Install Next.js with `npm install next@latest react@latest react-dom@latest`
5. Install npm dependencies with: `npm i`
6. Start development with `cargo tauri dev`
7. or build Raspirus with `cargo tauri build`

1. Clone the Repository
2. Execute `make install`
3. Run the application with `raspirus`

1. Clone the repository
2. Install Tauri and Prerequisites
3. Install npm
4. Install Next.js with `npm install next@latest react@latest react-dom@latest`
5. Install npm dependencies with: `npm i`
6. Start development with `cargo tauri dev`
7. or build Raspirus with `cargo tauri build`

Should you encounter any hiccups during your initial run or build, ensure that you've followed each step diligently. Confirm the accurate creation of both logs and config files.

3.3 Exploring the Backend

```
classDiagram
     Main <|-- DBOps
     Main <|-- Configs
     Main <|-- FileLogs
     Main <|-- Scanner
     Main: +Config config_file

     class DBOps {
          +Connection db_conn
          +String db_file
          +TauriWindow t_win
          +new()
          +update_db()
          +hash_exists()
     }

     class Configs {
          +Data data
          +new()
          +save()
          +load()
     }

     class FileLogs {
          +File file
          +log()
          +create_file()
     }

     class Scanner {
          +String scan_loc
          +DbOps db_ops
          +Vec malware_list
          +search_files()
          +create_hash()
          +get_folder_size()
     }
```

The backend, an essential cog in the Raspirus machinery, is meticulously crafted in Rust for superior performance. The primary file houses functions accessible from the frontend, which must yield JSON-compatible outcomes. For a detailed breakdown, reference the graph above outlining the backend's modular arrangement.

3.4 Unpacking the Frontend

Our frontend, developed with JavaScript via the Next.js framework, emphasizes user-friendliness and functionality. Comprising components and pages, it mirrors the simplicity and robustness of Next.js. Refer to the illustrated graph above for an approximate visual representation of the frontend's architecture.

3.5 Evaluating Test Coverage

- Backend tests, authored in Rust, can be executed via the `cargo test` command. Access these tests in the tests directory. Check test coverage on Codecov.
- Frontend tests, created with Selenium, are currently in development.

Thank you for your interest in contributing to Raspirus's development. Your expertise fuels our progress.

August 31, 2023

August 22, 2023

4. FAQ

App crashes when updating

On Windows, it has been observed that the app crashes when attempting to update the database. We are aware of this issue and actively working to resolve it. The problem arises because the update function requires administrative privileges, which Windows does not automatically provide. To temporarily resolve this issue, you can execute the app with administrative privileges. Right-click on the app and select "Run as administrator" to launch it with the necessary privileges.

Where does the Raspirus logo come from?

The logo of the Raspirus app features a red monster named Stuart, who is designed to represent a virus-eating creature. The logo was generated using DALL-E, along with creative image editing and merging. Stuart is a friendly monster, except when he's hungry for viruses. You can find additional media and documents in the dedicated repository. Feel free to use these images to create your own artwork and share them in the discussion boards.

Can I use Raspirus offline?

Yes, except for the update, everything on Raspirus works offline. The database is built locally, you only need to be connected to the internet during the database update, as we fetch the signatures from our GitHub repository.

What are the minimum requirements for the app to work?

Raspirus is built to work and almost any system. It even works well on a RPi 3B+. Nonetheless, we recommend having:

- 1 GB of RAM
- minimum of 10 GB of space (the database is sadly quite big)
- Dual core CPU (The better, the faster Raspirus will be)
- A display for the GUI, else you can only use it in CLI mode
- Any graphics card

My VScode setup is giving me issues

The Rust Analyzer plugin in Visual Studio Code searches for a `Cargo.toml` file in the current directory or its parent directory. To address this issue, you can add an option to the plugin settings and specify the location of your `Cargo.toml` file.

As mentioned in this comment, you can add the following lines to the end of your plugin settings JSON. Afterward, restart the Rust Analyzer for the modifications to take effect.

```
{
    "rust-analyzer.linkedProjects": [
        "/home/stuart/raspirus/raspirus/Cargo.toml"
    ]
}
```

Can't select directories/files

Unfortunately, as of this issue with Tauri, we currently can't allow users to select both files and folders. To switch between selecting a single file or folder, you need to change it in the Raspirus settings. There you will find a switch for it.

What is obfuscated mode?

Raspirus was originally intended for enterprise usage and therefore needed to be privacy-friendly. To ensure that, it added the "Obfuscation Mode", which will hide everything, detect malware faster and only display: "Malware found" or "No malware found". It is on by default, so if you want to know a bit more about your scan, you should probably deactivate it. You can do so in the settings.

error: found a virtual manifest instead of a package manifest

If you get this error when performing `cargo install` or using the Makefile, please note that it's a know issue. The solution is simple, as explained on this Stackoverflow answer, simply change the command to include the workspace, like this:
`cargo install --path src-tauri/`

How do I add my own signatures to the main repository?

Raspirus fetches signatures from the signatures repository and creates the database locally. You can add signatures to the repository by creating an issue or PR request on that repository. Or if you want to experiment with it locally, you can use the signatures-builder that we use to update the signatures in the signatures repository

March 30, 2024

April 5, 2023

4.1 Comments

5. GUIDES

5.1 Setting Up Your Local Documentation

Easily set up the documentation on your machine with these straightforward steps:

1. Clone the Repository: Start by cloning the repository to your local machine using this command:

```
git clone https://github.com/Raspirus/docs.git
```

1. Install Python Dependencies: Navigate to the project directory and install the necessary Python dependencies from the `requirements.txt` file using this command:

```
pip install -r requirements.txt
```

1. Launch the Project Locally: Begin your local project by executing this command:

```
mkdocs serve
```

This will initiate a development server, allowing you to access the documentation via http://localhost:8000 in your preferred web browser.

5.2 Exporting Documentation to PDF

If you prefer an offline PDF version of this documentation, follow these step-by-step instructions:

1. Follow the setup process outlined above.
2. Build the Documentation: Employ mkdocs and use the build command: `mkdocs build`.
3. Access the PDF: If the process goes smoothly, locate the resulting PDF file named `document.pdf` within the `site/pdf` directory.

Please note that the exported PDF may display minor issues with images and iFrames. Nevertheless, the text remains legible and suitable for offline sharing.

5.3 Introducing Translations

For translations of this documentation, visit Crowdin, a collaborative platform. Translations for UI strings are managed through JSON files. To contribute, craft your translations within the locales folder. Retain the uniqueness of keys when translating from the `en.json` file.

5.4 Scanner Insights

5.4.1 Scanning Compressed Files

> **Note**
>
> This option is not yet available for the Raspberry Pi. see why in the FAQ section

While Raspirus defaults to scanning folders, you can alter this behavior via the settings page. By toggling a switch, you can scan individual files, including compressed ones. It's important to note that only one file can be scanned at a time.

5.4.2 Navigating Logs and Configurations

In the event of unexpected occurrences or sudden app crashes, examining logs and configurations can provide insights. Locate these files in the following folders, based on your operating system. The ProjectDirs crate stores them at the following location:

Config file:

Platform	Value	Example
Linux	`$XDG_DATA_HOME` / `_project_path_` or `$HOME` /.local/share/ `_project_path_`	/home/alice/.local/share/barapp
macOS	`$HOME` /Library/Application Support/ `_project_path_`	/Users/Alice/Library/Application Support/ com.Foo-Corp.Bar-App
Windows	`{FOLDERID_RoamingAppData}` \ `_project_path_` \data	C:\Users\Alice\AppData\Roaming\Foo Corp\Bar App\data

Log files:

Platform	Value	Example
Linux	`$XDG_DATA_HOME` / `_project_path_` or `$HOME` /.local/share/ `_project_path_`	/home/alice/.local/share/barapp
macOS	`$HOME` /Library/Application Support/ `_project_path_`	/Users/Alice/Library/Application Support/ com.Foo-Corp.Bar-App
Windows	`{FOLDERID_LocalAppData}` \ `_project_path_` \data	C:\Users\Alice\AppData\Local\Foo Corp\Bar App\data

Ensure to include these files when reporting bugs, as they greatly assist in troubleshooting.

5.5 Enriching the Signature Database

Raspirus has its own signatures repository which are collected from various sources and updated approximately once or twice every month. You can check out all the signatures there and report false positives or missing signatures directly there by opening an issue or Pull request. If you want to build your own signatures database, you can use the signature-builder.

5.6 Raspberry Pi Deployment

Originally tailored for standalone Raspberry Pi deployment with touchscreen functionality (akin to kiosk mode), this project's primary purpose was scanning attached USB drives. While the project's scope has expanded, this feature remains intact. Follow the guide on Tauri and if you encounter any issue with it, make sure to let us know. We also provide pre-built executables on our website!

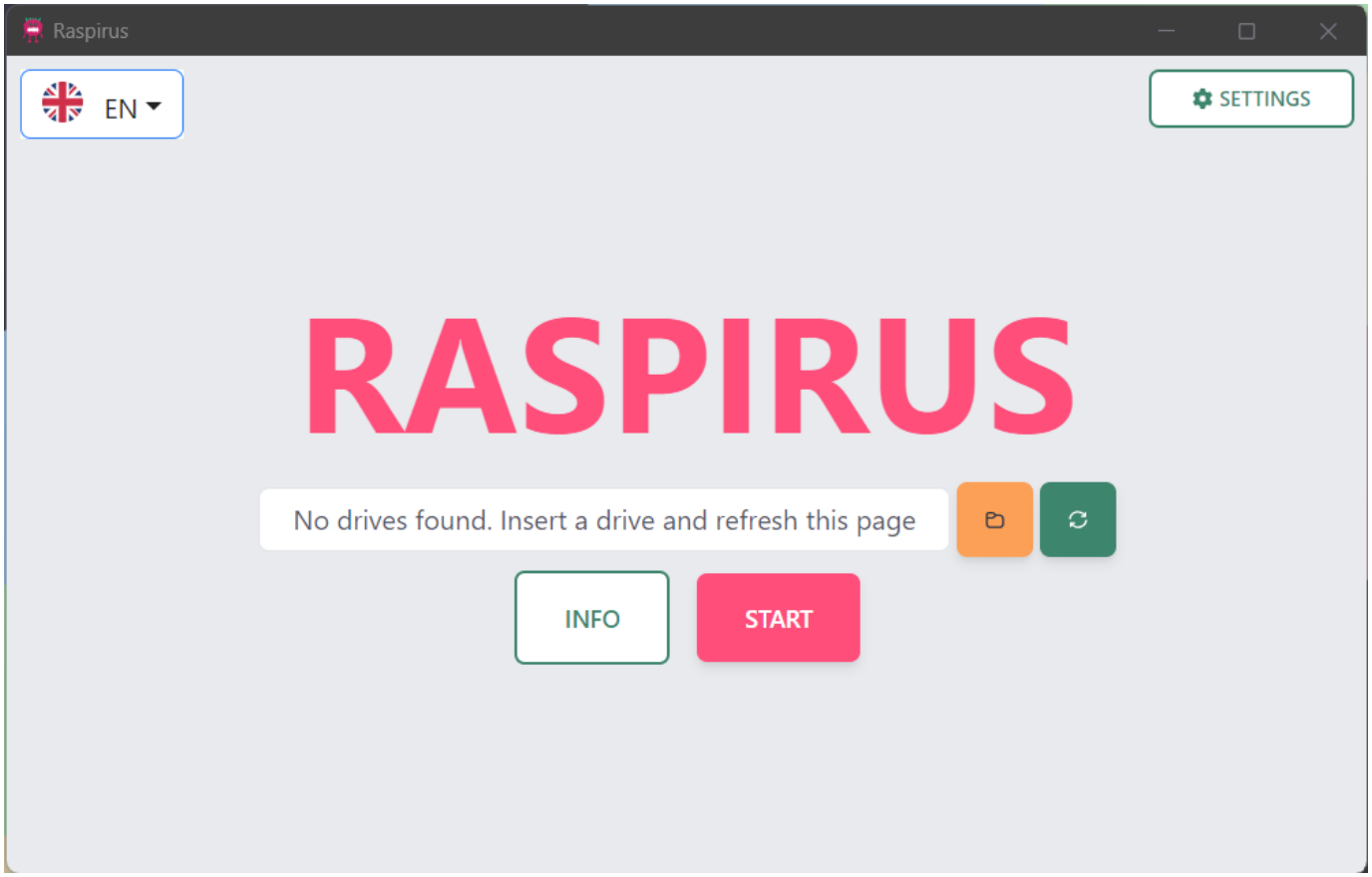
Thank you for choosing Raspirus as your malware protection solution. These comprehensive guides will ensure your experience is seamless and efficient.

March 30, 2024

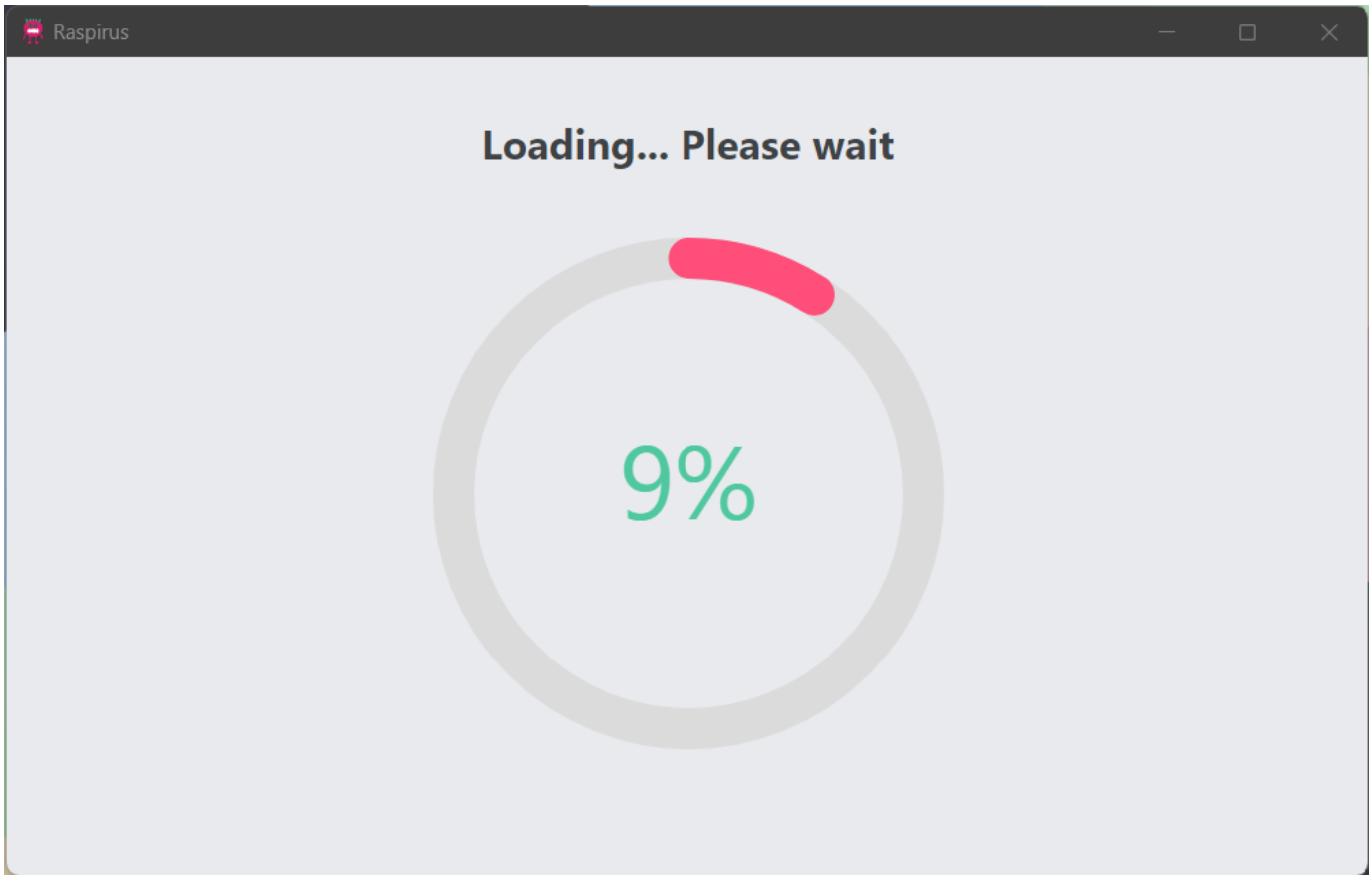
April 3, 2023

6. SCREENSHOTS

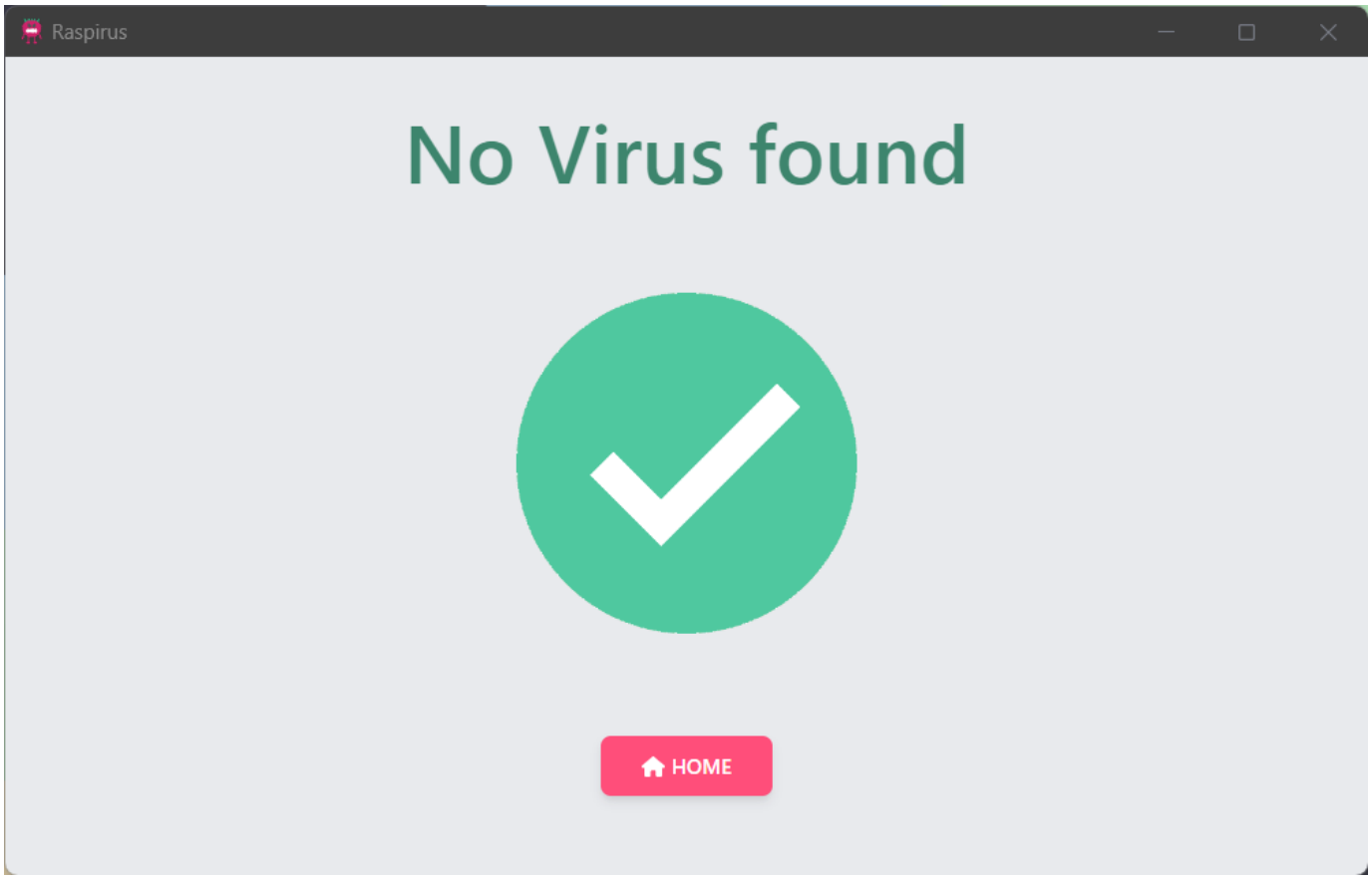
6.1 Home



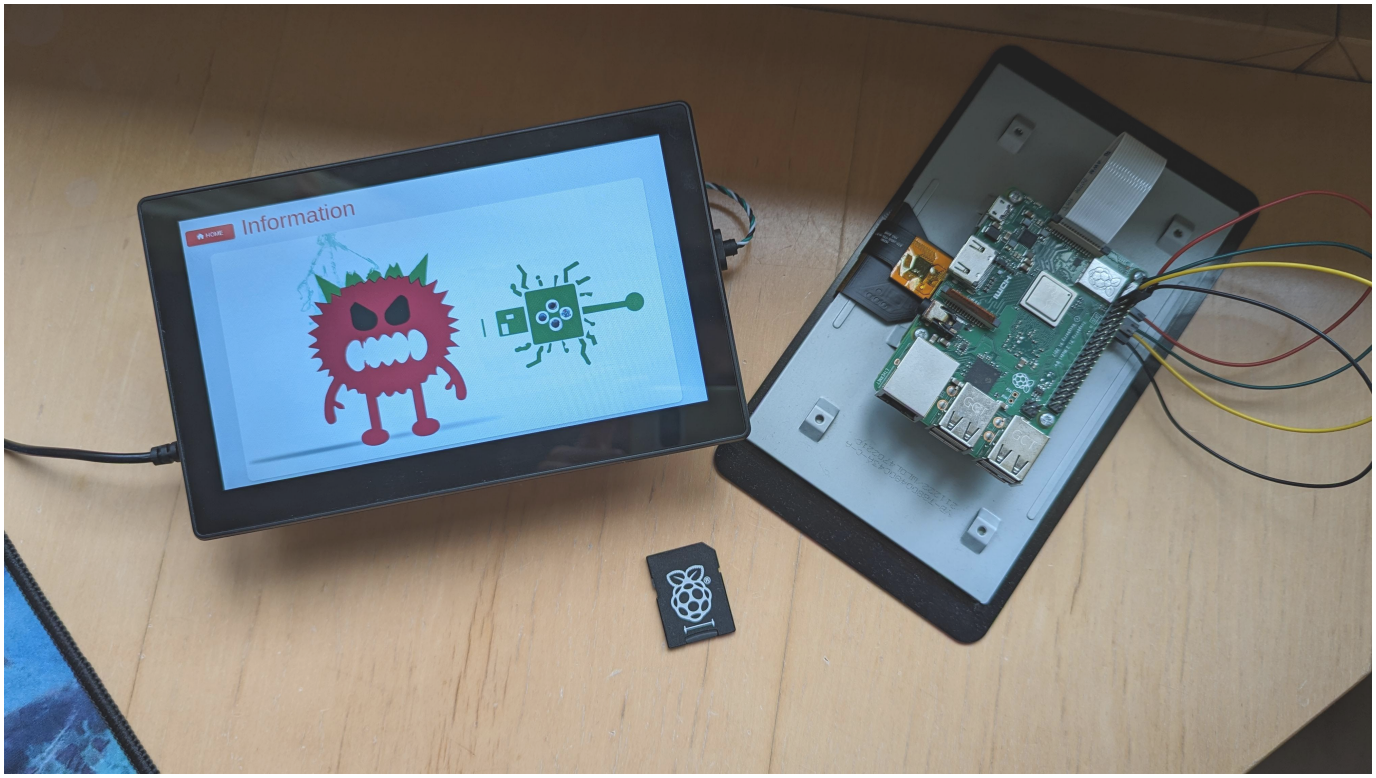
6.2 Scanning



6.3 Result



6.4 Raspberry Pi Setup



August 31, 2023

August 22, 2023

7. STARS

7.1 Maintainers

7.1.1 Benjamin Demetz

❤️ Maintainer

7.1.2 GamingGuy003

❤️ Co-Maintainer

7.2 Sponsors

7.2.1 Nurkanat Baysenkul

💰 Sponsor

7.3 Contributors

7.3.1 Matthias Dieter Wallnöfer

👩‍🏫 Mentoring

7.3.2 Zack Amoroso

📦 Linux Tester

7.3.3 Paul Guyot

💻 GitHub Action

7.4 Special Credits

7.4.1 Lairex59

💡 Ideas and Support

7.4.2 Tauri Devs

💡 Community Support

March 30, 2024

August 22, 2023