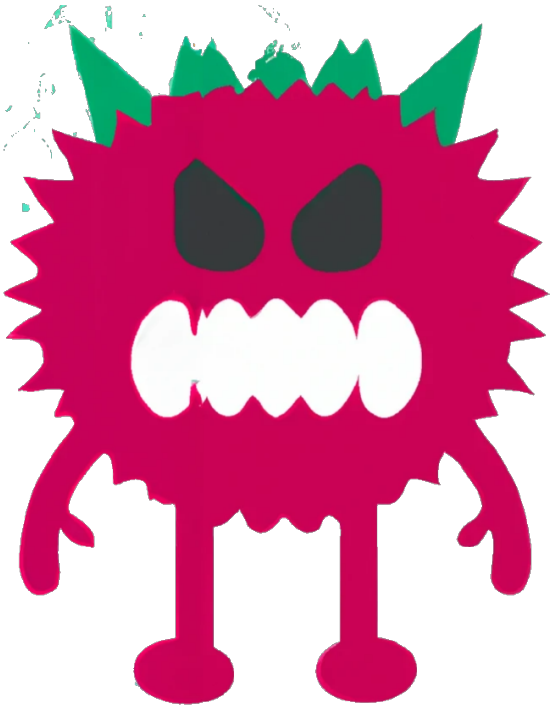


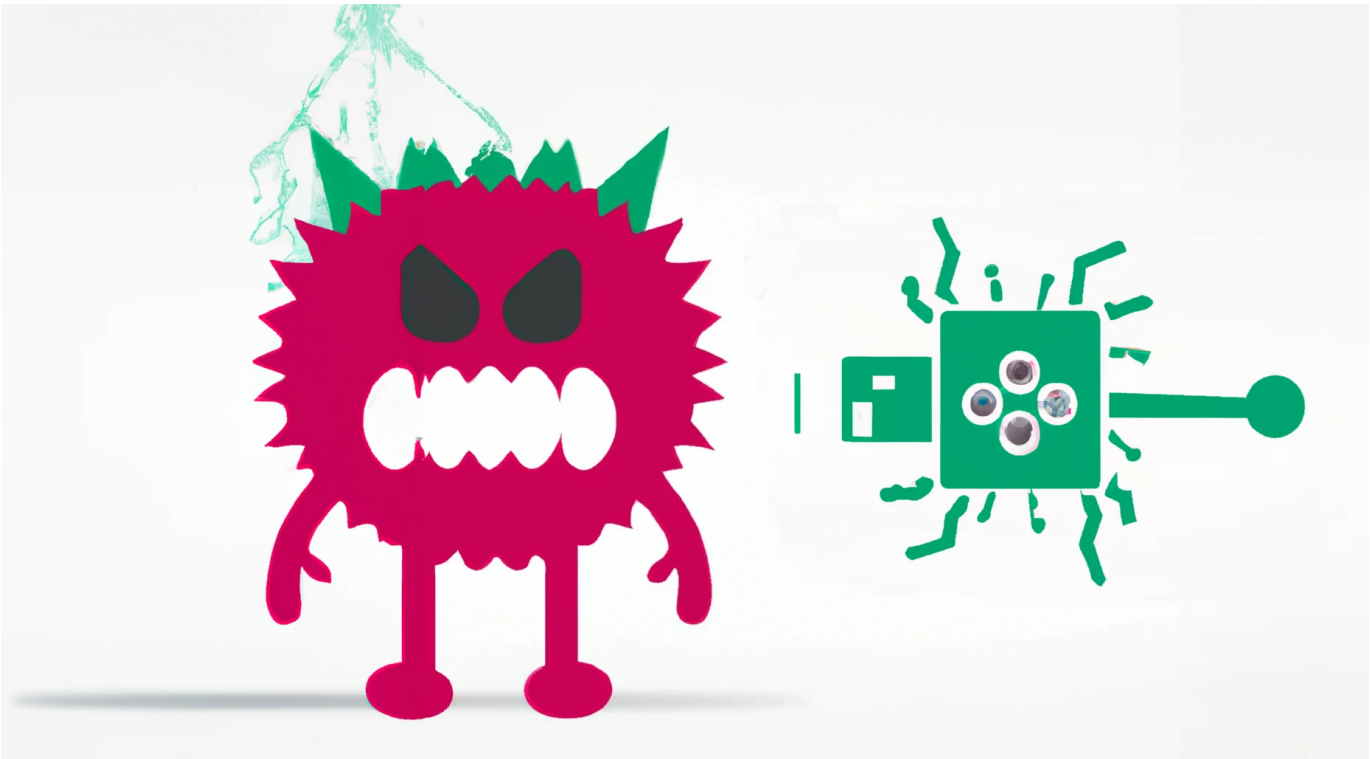
Raspirus docs

A simple hash-based virus scanner

Benjamin Demetz

Table of contents

1. HOME



1.1 Einführung

Raspirus: Verbessern Sie Ihren Malware-Schutz

Willkommen in der offiziellen Dokumentation zu Raspirus, Ihrem leichten Signaturenscanner. Ursprünglich entwickelt, um angeschlossene USB-Laufwerke mit einem Raspberry Pi zu scannen Raspirus hat sich zu einem vielseitigen Tool entwickelt, das auch lokale Dateien und Ordner scannt. Einige der herausragenden Funktionen beinhalten:

- **Kostenloser Schutz:** Raspirus arbeitet ausschließlich auf Spenden und sichert so erstklassigen Schutz ohne finanzielle Belastung.
- **Maßgeschneiderte Signaturerkennung:** Unser individueller Signatur-basierter Ansatz garantiert eine genaue Identifizierung von Malware.
- **Umfassende Datei-Scans:** Raspirus kann komprimierte Dateien effizient scannen, um sicherzustellen, dass keine Bedrohung unerkannt bleibt.
- **Datenschutz Priorisiert:** Raspirus bietet eine datenschutzfreundliche Option, schützt Ihre persönlichen Daten.
- **Cross-Platform Convenience:** Genießen Sie die Vorteile von Raspirus Schutz auf einer Vielzahl von Betriebssystemen.
- **Swift and Dependable:** Zählt auf Raspirus für schnelle und zuverlässige Malware-Erkennung.
- **Sleek Modern Interface:** Mit Benutzerfreundlichkeit im Kern verfügt Raspirus über eine schöne und intuitive Benutzeroberfläche.

1.2 Erste Schritte

1.2.1 Für reguläre Benutzer

Mit Raspirus zu beginnen ist ein Kinderspiel. Folgen Sie diesen einfachen Schritten:

1. Besuche unsere website oder gehe auf die GitHub Release Seite.
2. Laden Sie die ausführbare Datei herunter, die zu Ihrem Betriebssystem passt.

> **Notiz**
>
> Wenn du Raspirus auf der Raspberry Pi als eigenständige Anwendung verwenden möchtest, haben wir dafür einen [dedizierten Anleitung](https://raspirus.github.io/docs/guides#Raspberry-Pi).

1.2.2 Für Entwickler

Sind Sie ein Entwickler, der Raspirus aufbauen will? Wir haben dich bedeckt. Schaue dir unsere umfassenden Anleitungen für verschiedene Betriebssysteme in der Entwicklersektion an.

1.3 Fragen?

Haben Sie Fragen zu Raspirus? Wir sind hier, um zu helfen!

- Besuchen Sie unsere FAQ Sektion für Antworten auf häufige Anfragen.
- Trete unserer gedeihenden Community auf dem Discord Server bei, um sich mit anderen Nutzern zu beschäftigen.
- Wenn Sie einen Fehler gefunden haben, durchsuchen Sie die GitHub Probleme, um zu sehen, ob er bereits gemeldet ist.

Vielen Dank, dass Sie sich für Raspirus für Ihren Malware-Schutz entschieden haben. Gemeinsam machen wir die digitale Welt für alle sicherer.

22. Dezember 2023

3. April 2023

2. VERTRIBUTING

2.1 Verbessere die Codebase

Als Entwickler, der Raspirus's Funktionalität verstärken möchte, sind Ihre Beiträge von unschätzbarem Wert. Bitte halten Sie sich an die folgenden Richtlinien:

- **Überprüfe existierende Probleme:** Bevor du eintauchst, überprüfe existierende Probleme, um Duplikate zu vermeiden. Wenn keine existiert, zögern Sie nicht, eine neue zu öffnen.
- **Dokumentation ist Schlüssel:** Stellen Sie immer sicher, dass Ihr Code gut dokumentiert ist. Denken Sie daran, Tests anzupassen, um die Code-Integrität zu erhalten.
- Lies den Verhaltenskodex

2.2 Die Dokumentation bereichern

Die Dokumentation von Raspirus wird mithilfe von Markdown erstellt und von [MkDocs]betrieben (https\://www.mkdocs.org/user-guide/installation/) und Python. Um zur Dokumentation beizutragen:

- **Folge dem Guide:** Um die Dokumentation auf deinem lokalen Rechner zu starten, folge der guide.
- **Material-Theme:** Unsere Dokumentation verwendet das Material-Theme, das Markdown-Funktionalität erweitert. Entdecke mehr darüber here.

2.3 Antriebsübersetzungen

Unsere Dokumentation ist mehrsprachig dank Crowdin, einem intuitiven externen Dienst. GUI Übersetzungen werden über JSON-Dateien bearbeitet. Werfen Sie einen Blick auf die guide für umfassende Einsichten.

2.4 Kunstwerk und Medien infizieren

Das Repository für Kunstwerke (Logos, Banner, etc.) und Medien (Powerpoints, Artikel, Graphen, etc.) wird separat auf diesem Projektarchiv gehostet. Fühlen Sie sich frei, unsere visuelle Präsenz zu unterstützen und zu stärken. Beachten Sie, dass das aktuelle Logo und Banner aufgrund fehlender künstlerischer Expertise von AI generiert wurden.

2.5 Feedback geben

Jenseits der oben genannten Beiträge gibt es andere wirkungsvolle Möglichkeiten, sich einzubringen:

- **Trete Discord:** Komm mit unserer Community auf dem Discord Server um wertvolle Feedback und Ideen zu teilen.
- **Überlege dir Spenden:** Wenn du Wert in Raspirus findest, überlege donating um laufende Entwicklung zu unterstützen.
- **Spread the Word:** Eine einfache, aber effektive Möglichkeit, dazu beizutragen, ist das Herunterladen von Raspirus und die Einführung an Freunde.

Vielen Dank für Ihr Engagement für Raspirus's Wachstum und Verbesserung. Ihr Engagement ist ein Eckpfeiler unseres Erfolgs.

22. Dezember 2023

22. August 2023

3. DEVELOPER

3.1 Navigation durch die Architektur

```
graph LR
     A[Start] --> B{Scan-Position angegeben? ;
     B --> |Ja| C[Start scan];
     C --> |Start Loop| D[Datei gefunden];
     D --> E[Erstelle Hash];
     E --> F[Vergleiche Hash];
     F --> G{Hash in DB? ;
     G --> |Ja| H[Flagge als Malware];
     G --> |Nein| I[Flagge als Safe];
     H & I --> J[Iteration fortsetzen];
     J --> K{Letzte Datei? ;
     K --> |Ja| L[Scanner stoppen];
     L --> M[Ergebnisse anzeigen];
     K --> | N[No| Erneut starten];
     N --> D;
     B --> |Nein | O[Stop]
```

Raspirus gliedert sich in zwei integrale Komponenten: Frontend und Backend. Diese Komponenten, die in unterschiedlichen Sprachen und Frameworks erstellt wurden, sind über ein Drittanbieter-Framework mit dem Namen Tauri miteinander verbunden. Dieser Rahmen erleichtert nicht nur die Kommunikation zwischen Frontend und Backend, sondern ermöglicht auch die Integration von Rust in das Frontend. Darüber hinaus ermöglicht Tauri die Verteilung von Raspirus auf verschiedene Betriebssysteme.

3.2 Starte deine Entwicklerreise

\=== "Windows" Kopiere das Repository 2. Installiere Tauri and Prerequisites 3. Installieren Sie npm 4. 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` Installiere npm Abhängigkeiten mit: `npm i` 6. Beginnen Sie die Entwicklung mit `cargo tauri dev` 7. oder bauen Sie Raspirus mit "cargo tauri build"

\=== "Linux" Klone das Repository 2. Führe `make install` 3 aus. 1. Clone the Repository 2. Execute `make install` 3. Run the application with `raspirus`

\=== "macOS" Kopiere das Repository 2. Installiere Tauri and Prerequisites 3. Installieren Sie npm 4. 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` Installiere npm Abhängigkeiten mit: `npm i` 6. Beginnen Sie die Entwicklung mit `cargo tauri dev` 7. oder bauen Sie Raspirus mit "cargo tauri build"

Sollten Sie während Ihres ersten Laufs oder Builds auf irgendwelche Fehler stoßen, stellen Sie sicher, dass Sie jeden Schritt gewissenhaft verfolgt haben. Bestätigen Sie die genaue Erstellung von Logs und Konfigurationsdateien.

3.3 Erforsche das 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()
    }
```

Das Backend, ein essentielles Rad in der Raspirus-Maschine, wird in Rust sorgfältig für eine herausragende Leistung hergestellt. Die primäre Datei enthält Funktionen, die über das Frontend zugänglich sind, was JSON-kompatible Ergebnisse liefern muss. Für eine detaillierte Aufschlüsselung siehe die obige Darstellung der modularen Anordnung des Backends.

3.4 Frontend entpacken

Unsere Frontend, die mit JavaScript über das Next.js Framework entwickelt wurde, betont Benutzerfreundlichkeit und Funktionalität. Er enthält Komponenten und Seiten und spiegelt die Einfachheit und Robustheit von Next.js wider. Eine ungefähre visuelle Darstellung der Frontend-Architektur finden Sie im obigen Bild.

3.5 Testabdeckung auswerten

- Backend-Tests, die in Rust erstellt wurden, können über den Befehl "cargo test" ausgeführt werden. Greifen Sie auf diese Tests im test-Verzeichnis. Prüfen Sie die Testabdeckung auf Codecov.
- Die mit Selenium erstellten Frontend-Tests befinden sich derzeit in der Entwicklung.

Vielen Dank für Ihr Interesse, zur Entwicklung von Raspirus beizutragen. Ihre Expertise stärkt unseren Fortschritt.

22. Dezember 2023

22. August 2023

4. FAQ

???+ question "App crashes when updating" On Windows, it has been observed that the app crashes when attempting to update the database. Wir sind uns dieser Frage bewusst und arbeiten aktiv daran, sie zu lösen. Das Problem tritt auf, weil die Update-Funktion Administratorrechte benötigt, die Windows nicht automatisch zur Verfügung stellt. Um dieses Problem vorübergehend zu beheben, können Sie die App mit Administratorrechten ausführen. Klicken Sie mit der rechten Maustaste auf die App und wählen Sie "Als Administrator ausführen", um sie mit den nötigen Rechten zu starten.

her kommt das Raspirus-Logo?

Das Logo der Raspirus App enthält ein rotes Monster namens Stuart, das eine virusfressende Kreatur darstellen soll. Das Logo wurde mit DALL-E zusammen mit kreativer Bildbearbeitung und Zusammenführung generiert. Stuart ist ein freundliches Monster, außer wenn er hungrig nach Viren ist. Du kannst zusätzliche Medien und Dokumente im dedizierten Projektarchiv finden. Du kannst diese Bilder verwenden, um dein eigenes Kunstwerk zu erstellen und sie in den Diskussionsforen“ zu teilen.

??+ Frage "Mein VScode Setup gibt mir Probleme"

```
Das Rust Analyzer Plugin in Visual Studio Code sucht nach einer `Cargo.toml` Datei im aktuellen Verzeichnis oder seinem übergeordneten Verzeichnis. Um dieses
Problem zu beheben, können Sie eine Option zu den Plugin-Einstellungen hinzufügen und den Standort Ihrer `Cargo` angeben. oml` Datei.

Wie in [diesem Kommentar](https://github. om/rust-lang/rust-analyzer/issues/2649#issuecomment-691582605), können Sie die folgenden Zeilen am Ende Ihrer
Plugin-Einstellungen JSON hinzufügen. Anschließend starten Sie den Rust Analyzer neu, damit die Änderungen wirksam werden.

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

???+ question "Can't select directories/files" Unfortunately, as of this issue with Tauri, we currently can't allow users to select both files and folders. Um zwischen der Auswahl einer einzelnen Datei oder eines einzelnen Ordners zu wechseln, müssen Sie diese in den Raspirus Einstellungen ändern. Dort finden Sie einen Schalter dafür.

??+ Frage "Was ist verschleierter Modus?" Raspirus war ursprünglich für den Unternehmenseinsatz gedacht und musste daher privatfreundlich sein. Um dies zu gewährleisten, fügte es den "Obfuscation Mode" hinzu, der alles versteckt, Malware schneller erkennt und nur anzeigt: "Malware gefunden" oder "Keine Malware gefunden". Sie ist standardmäßig eingeschaltet, wenn Sie also etwas mehr über Ihren Scan wissen möchten, sollten Sie ihn wahrscheinlich deaktivieren. Sie können dies in den Einstellungen tun.

???+ question "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. Die Lösung ist einfach, wie auf der [this]erklärt (https\:// stackoverflow. om/a/76271890) Stackoverflow-Antwort, ändere einfach den Befehl um den Arbeitsbereich einzubinden, wie z.B.: `cargo install --path src-tauri/`

22. Dezember 2023

5. April 2023

4.1 Kommentare

5. GUIDES

5.1 Lokale Dokumentation einrichten

Richten Sie einfach die Dokumentation auf Ihrem Rechner mit diesen einfachen Schritten ein:

1. Clone this repository by following the instructions on GitHub.

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

1. Navigate to the cloned directory and install the required dependencies.

```
You can install the dependencies by running the following command: `pip install -r requirements.txt`.
```

1. Starte das Projekt lokal: Beginne dein lokales Projekt, indem du diesen Befehl ausführst:

```
mkdocs dienen
```

Dies wird einen Entwicklungsserver initiieren, der dir erlaubt, die Dokumentation über http://localhost:8000 in deinem bevorzugten Webbrowser zuzugreifen.

5.2 Export to PDF

If you would like to have an offline version of this documentation in PDF format, you can follow these step-by-step instructions:

1. Folgen Sie den oben beschriebenen Installationsprozess.
2. Install mkdocs and execute the build command: `mkdocs build`.
3. If everything goes smoothly, the resulting PDF file should be located in the `site/pdf` directory with the name `document.pdf`.

Please be aware that the exported PDF may have some issues with images and iFrames, but the text should be readable and suitable for sharing offline. Dennoch bleibt der Text lesbar und für die Offline-Freigabe geeignet.

5.3 Übersetzungen

Für Übersetzungen dieser Dokumentation besuchen Sie bitte Crowdin, eine gemeinschaftliche Plattform. Übersetzungen für UI-Zeichenketten werden über JSON-Dateien verwaltet. Erstelle deine Übersetzungen im locales Ordner. Behalte die Einzigartigkeit der Schlüssel bei der Übersetzung aus der `en.json` Datei.

5.4 Scanner-Einblicke

5.4.1 Scanne komprimierte Dateien

Notiz

```
Diese Option ist für die Raspberry Pi noch nicht verfügbar. Schau, warum in der [FAQ Sektion](https://raspirus.github.io/docs/faq)
```

Während Raspirus standardmäßig Ordner scannt, können Sie dieses Verhalten über die Einstellungsseite ändern. Durch Umschalten können Sie einzelne Dateien, einschließlich komprimierter Dateien, scannen. Es ist wichtig zu beachten, dass nur eine Datei gleichzeitig gescannt werden kann.

5.4.2 Protokolle und Konfigurationen navigieren

Bei unerwarteten Vorfällen oder plötzlichen Abstürzen der App kann die Prüfung von Logs und Konfigurationen Einblicke bieten. Suchen Sie diese Dateien in den folgenden Ordnern, basierend auf Ihrem Betriebssystem. Das ProjectDirs crate speichert diese an der folgenden Stelle:

Konfigurationsdatei:

Plattform	Wert	Beispiel
Linux	`$XDG_DATA_HOME` / `_project_path_` oder `$HOME` /.local/share/ `_project_path_`	/home/alice/.local/share/barapp
macOS	`$HOME` /Library/Application Support/ `_project_path_`	/Benutzer/Alice/Library/Application Support/com.Foo-Corp.Bar-App
Fenster	`{FOLDERID_RoamingAppData}` \ `_project_path_` \data	C:\Benutzer\Alice\AppData\Roaming\Foo Corp\Bar App\data

Logdateien:

Plattform	Wert	Beispiel
Linux	`$XDG_DATA_HOME` / `_project_path_` oder `$HOME` /.local/share/ `_project_path_`	/home/alice/.local/share/barapp
macOS	`$HOME` /Library/Application Support/ `_project_path_`	/Benutzer/Alice/Library/Application Support/com.Foo-Corp.Bar-App
Fenster	`{FOLDERID_LocalAppData}` \ `_project_path_` \data	C:\Benutzer\Alice\AppData\Local\Foo Corp\Bar App\data

Stellen Sie sicher, dass Sie diese Dateien beim Melden von Fehlern einfügen, da sie bei der Fehlerbehebung sehr hilfreich sind.

5.5 GitHub Integration

5.5.1 Die Anreicherung der Signaturdatenbank

KOMMEN SOON

5.6 Raspberry Pi Einsatz

Ursprünglich für den eigenständigen Einsatz von Raspberry Pi mit Touchscreen-Funktionalität (ähnlich dem Kiosk Modus) konzipiert, war dieses Projekt in erster Linie das Scannen angehängter USB-Laufwerke. Während der Umfang des Projekts erweitert wurde, bleibt diese Funktion intakt. Folge der Anleitung auf Tauri und wenn du Probleme damit hast, lass es uns wissen.

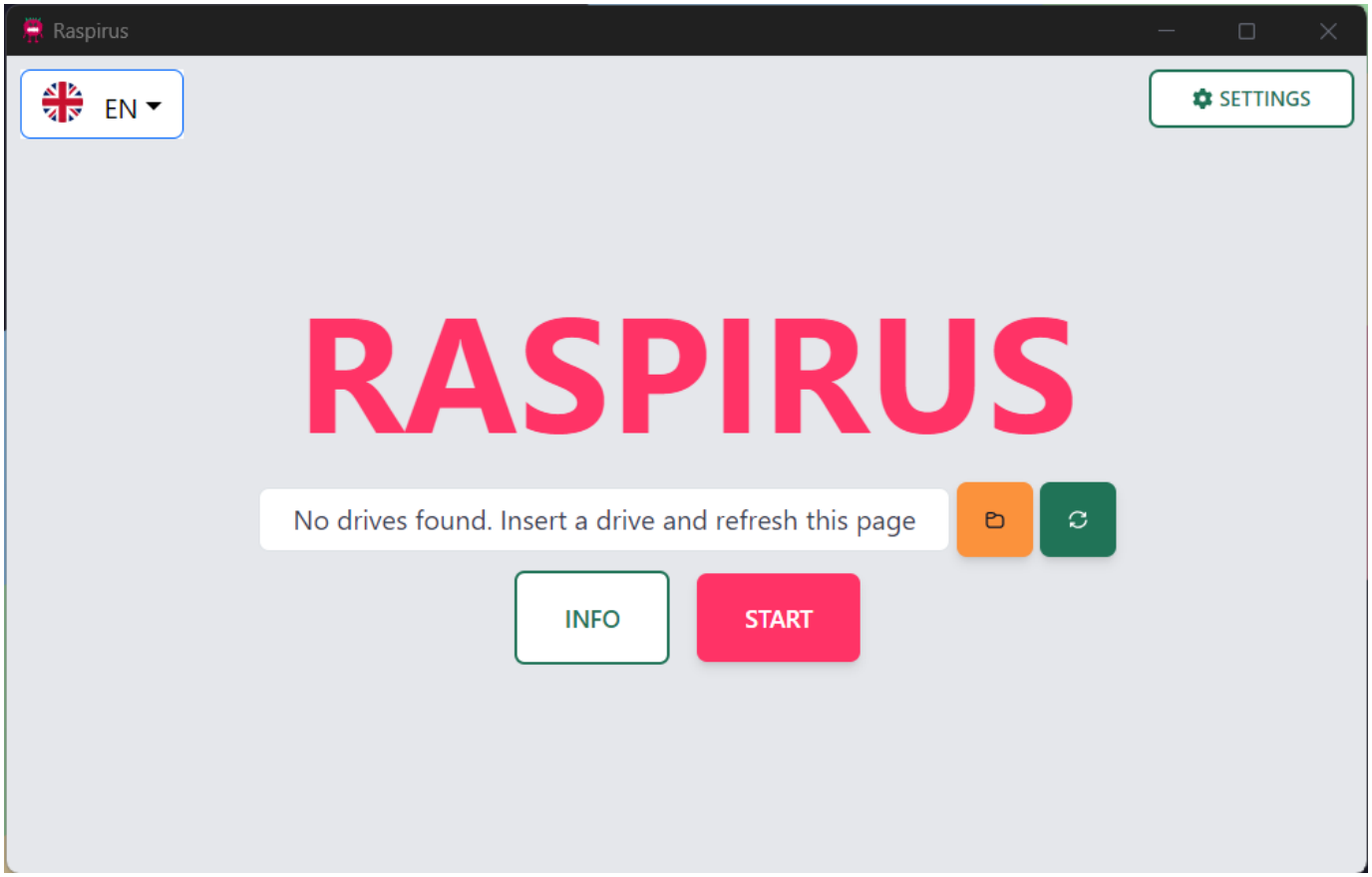
Vielen Dank, dass Sie Raspirus als Ihre Malware-Schutzlösung gewählt haben. Diese umfassenden Führer sorgen dafür, dass Ihre Erfahrung nahtlos und effizient ist.

22. Dezember 2023

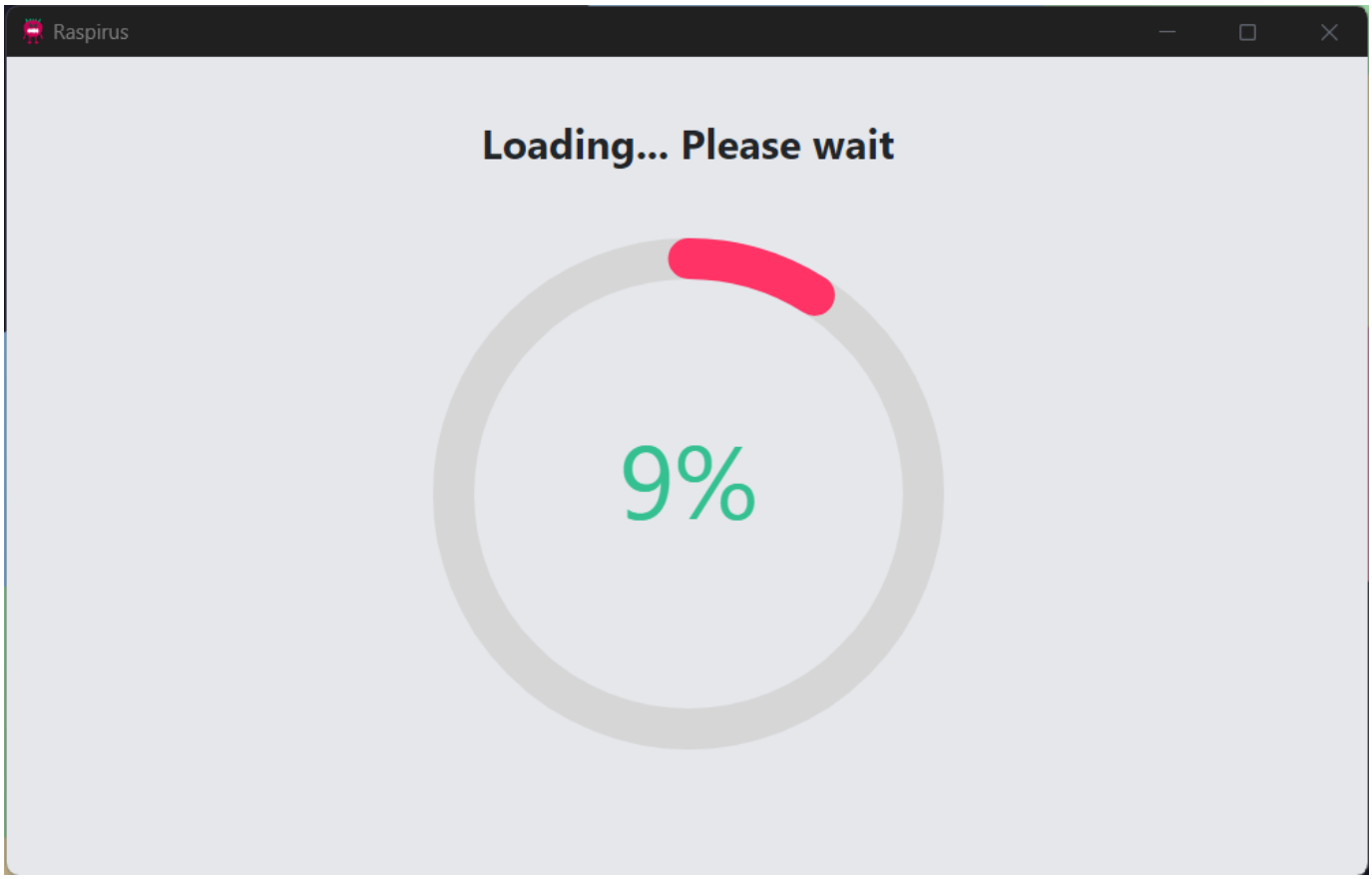
3. April 2023

6. SCREENSHOTS

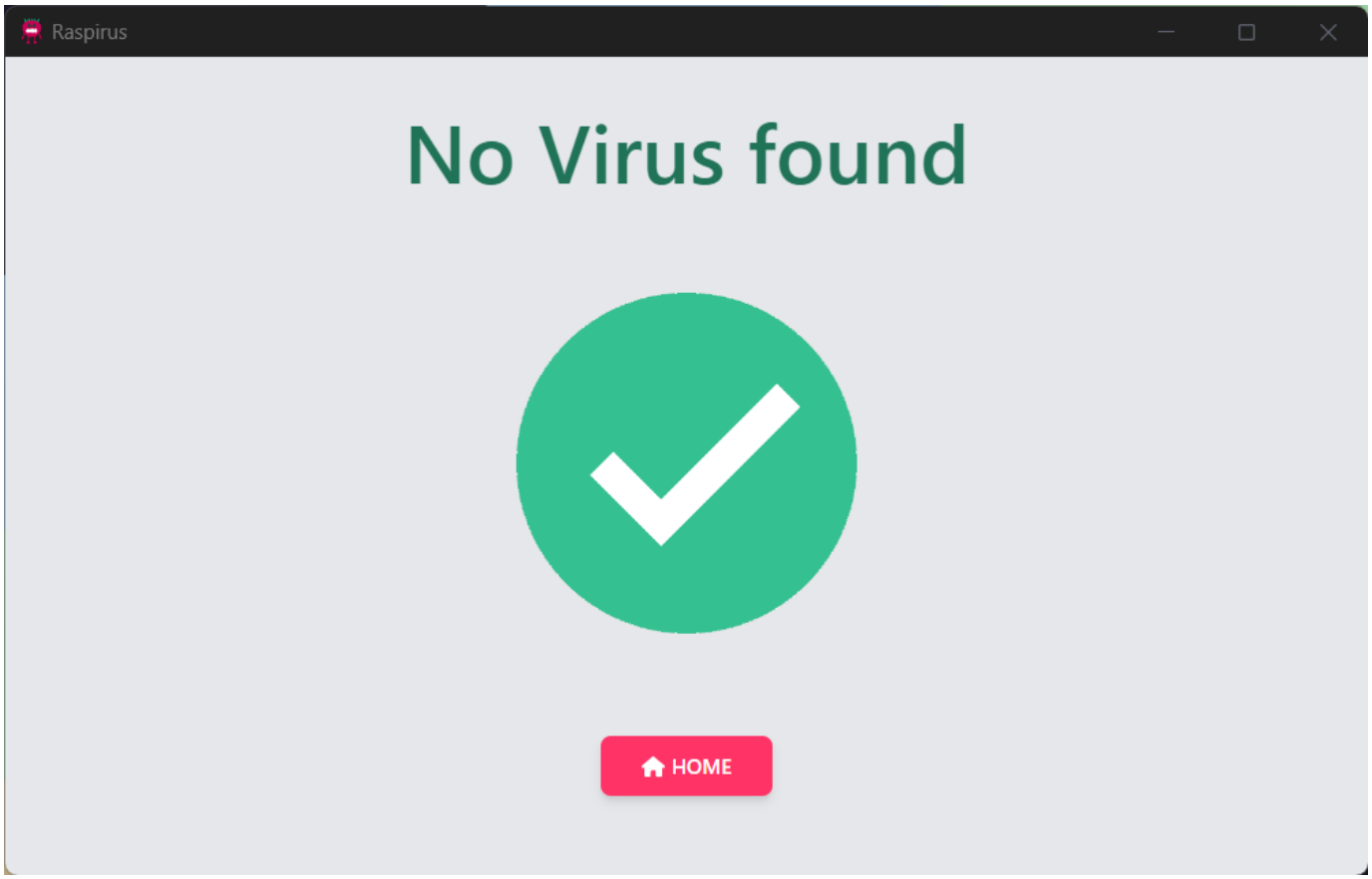
6.1 Zuhause



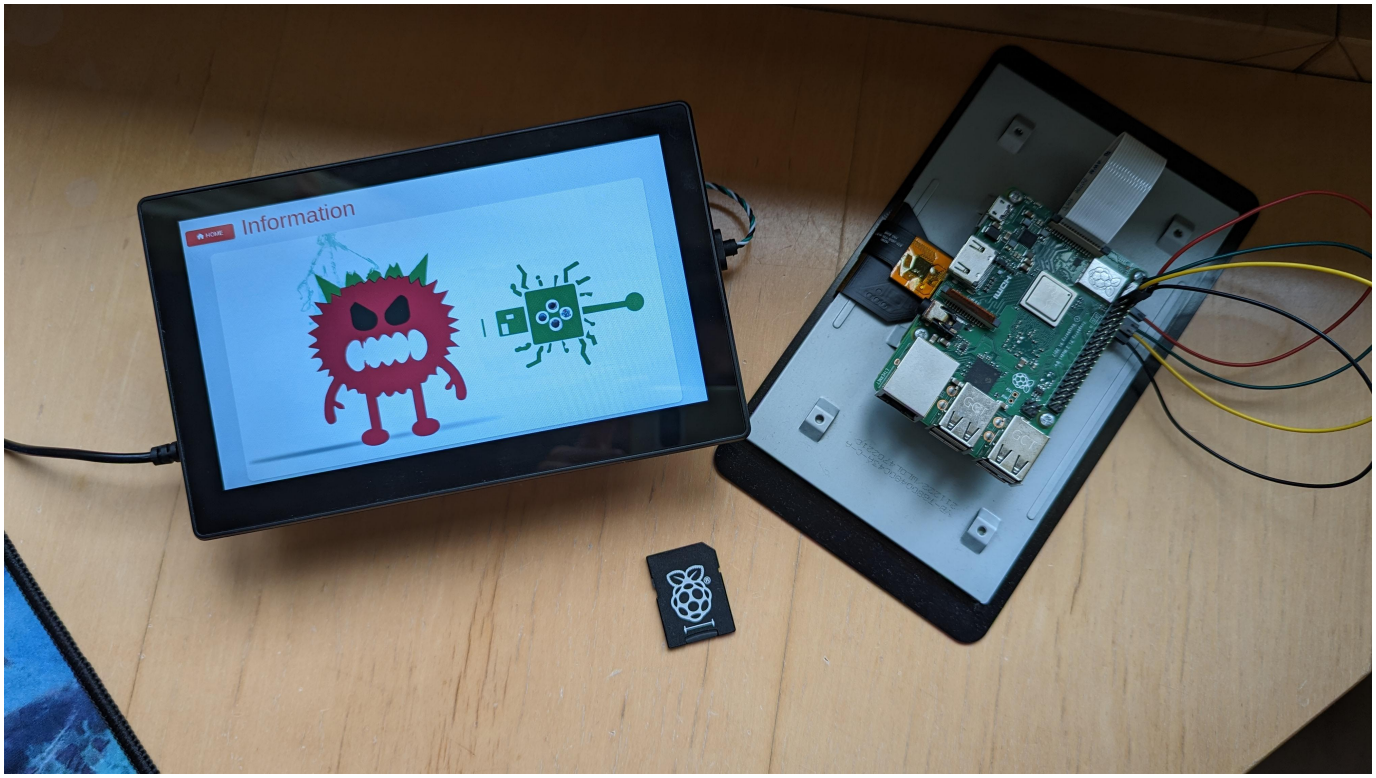
6.2 Scannen



6.3 Ergebnis



6.4 Raspberry Pi Setup



22. Dezember 2023

22. August 2023

7. STARS

7.1 Sponsoren

7.1.1 Benjamin Demetz

❤️ Maintainer

7.1.2 Nurkanat Baysenkul

💰 Sponsor

7.2 Mitwirkende

7.2.1 Matthias Dieter Wallnöfer

👩‍🏫 Mentoring

7.2.2 GamingGuy003

:Laptop: Backend-Entwickler

7.2.3 Zack Amoroso

📦 Linux Tester

7.2.4 Paul Guyot

💻 GitHub Action

7.3 Spezielle Credits

7.3.1 Lairex59

💡 Ideen und Support

22. Dezember 2023

22. August 2023