System Setup Guide

macOS & Windows

A complete guide for installing, configuring, and verifying the VIOE development environment on your workstation.

VIOE

Vulnerability Intelligence & Ownership Engine

Document Version	1.0
Date	January 28, 2026
Classification	Internal - Team Distribution
Platform Coverage	macOS / Windows 10 & 11

VIOE - Vulnerability Intelligence & Ownership Engine | Confidential

Table of Contents

1. Introduction

What This Guide Is For

This document provides step-by-step instructions for setting up the development environment required to run the VIOE (Vulnerability Intelligence & Ownership Engine) web application on your local workstation. It covers both macOS and Windows operating systems.

Who Should Use This Guide

- New team members joining the VIOE project
- Contractors or partners who need to run the application locally
- QA engineers setting up testing environments
- Anyone who needs to build or review the VIOE application

What You Will Achieve

After completing this guide, you will have:

- Node.js and npm installed and configured
- Git version control installed and configured
- A code editor ready for development
- The VIOE project cloned, configured, and running locally
- Verified that the application builds and runs correctly

> **NOTE: Time to Complete**
> Setting up the full environment typically requires completing all steps in order. If you already have some tools installed, you can skip those sections after verifying the correct versions.

2. System Requirements

Operating System Versions

The following minimum OS versions are supported:

- macOS: macOS 12 Monterey or later (macOS 13 Ventura or 14 Sonoma recommended)
- Windows: Windows 10 version 1903 or later (Windows 11 recommended)

Hardware Requirements

- Processor: 64-bit CPU (Intel or Apple Silicon for Mac; Intel or AMD for Windows)
- RAM: 8 GB minimum (16 GB recommended for smooth development)
- Disk Space: At least 2 GB free for tools and project files
- Display: 1280x720 minimum resolution

Internet Requirements

- Stable broadband connection for downloading tools and npm packages
- Access to https://nodejs.org, https://git-scm.com, and https://npmjs.com
- If behind a corporate proxy, have proxy settings ready (host, port, credentials)

Required Software Overview

The following tools must be installed. Each is covered in detail in the platform-specific sections:

Tool	Version	Purpose
Node.js	v18.x or v20.x LTS	JavaScript runtime for building and running the app
npm	v9+ (bundled w/ Node)	Package manager for installing dependencies
Git	v2.30+	Version control for cloning the project repository
Code Editor	Latest stable	VS Code recommended; any modern editor works
Web Browser	Chrome / Edge / Firefox	For testing and viewing the application

WARNING: Node.js Version

VIOE requires Node.js v18 or v20 LTS. Do not use odd-numbered versions (e.g., v19, v21) as they are unstable releases and may cause build failures. You can check your version with: node --version

3. macOS Setup Guide

3.1 Install Node.js

Node.js is the JavaScript runtime needed to install dependencies, run the development server, and build the VIOE application.

Option A: Direct Download (Recommended for Beginners)

1. Open your web browser and navigate to https://nodejs.org
2. Click the LTS (Long Term Support) download button labeled "Recommended For Most Users."
3. Open the downloaded .pkg file and follow the installer prompts.
4. Click Continue through each step, accept the license agreement, and click Install.
5. Enter your Mac password when prompted and wait for installation to complete.
6. Open Terminal (Applications > Utilities > Terminal) and verify:

```
node --version
npm --version
```

You should see version numbers like v20.x.x and 10.x.x respectively.

Option B: Using Homebrew (For Advanced Users)

If you have Homebrew installed, you can install Node.js via the terminal:

```
brew install node@20
brew link node@20
```

TIP: Multiple Node Versions

If you work on multiple projects with different Node.js versions, consider installing nvm (Node Version Manager). Install it with:

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash

Then use: nvm install 20 && nvm use 20

3.2 Install Git

Git is required to clone the VIOE project repository.

1. Open Terminal and type git --version to check if Git is already installed.
2. If not installed, macOS will prompt you to install the Xcode Command Line Tools. Click Install.

3. Alternatively, download Git from https://git-scm.com/download/mac
4. After installation, configure your identity:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@company.com"
```

5. Verify installation:

```
git --version
```

3.3 Install a Code Editor

We recommend Visual Studio Code (VS Code) for its excellent JavaScript/React support.

1. Navigate to https://code.visualstudio.com
2. Download the macOS version (.dmg file).
3. Open the .dmg file and drag VS Code to your Applications folder.
4. Open VS Code, then open the Command Palette (Cmd+Shift+P) and type 'shell command' to install the 'code' command in PATH.

TIP: Recommended VS Code Extensions

Install these extensions for the best VIOE development experience:

- ES7+ React/Redux/React-Native Snippets
- Tailwind CSS IntelliSense
- ESLint
- Prettier - Code Formatter

3.4 Clone and Configure the Project

1. Open Terminal and navigate to where you want to store the project:

```
cd ~/Projects
mkdir -p ~/Projects && cd ~/Projects
```

2. Clone the VIOE repository (or extract the ZIP archive):

```
# If using Git:
git clone <repository-url> vioe
cd vioe

# If using the ZIP archive:
unzip VIOE_UAT.zip -d vioe
cd vioe
```

3. Create the environment configuration file:

```
cp .env.example .env
```

4. The default .env is pre-configured for demo/mock mode. No changes are needed for local testing.
5. Install project dependencies:

```
npm install
```

This will download all required packages. The process requires internet access and may download approximately 200 MB of packages.

6. Start the development server:

```
npm run dev
```

7. Open your web browser and navigate to the URL shown in the terminal (typically http://localhost:5173).

> **NOTE: First Launch**
>
> The application starts in demo mode by default. On the login page, click any of the quick login buttons (Admin, Manager, Analyst, Viewer) to explore the application with different permission levels.

3.5 Verify Installation

Run the following commands to confirm everything is set up correctly:

```
# Check tool versions
node --version        # Should show v18.x or v20.x
npm --version         # Should show v9+ or v10+
git --version         # Should show v2.30+

# Run the test suite
npm run test:run      # All tests should pass

# Build for production
npm run build         # Should complete without errors
```

3.6 Common macOS Issues

Gatekeeper Blocks Installation

If macOS shows "App can't be opened because it is from an unidentified developer," go to System Settings > Privacy & Security, scroll down, and click 'Open Anyway' next to the blocked application.

Permission Denied Errors with npm

If you see EACCES permission errors when running npm install globally, do NOT use sudo. Instead, reconfigure npm's default directory:

```
mkdir ~/.npm-global
npm config set prefix '~/.npm-global'
# Add to ~/.zshrc:
export PATH=~/.npm-global/bin:$PATH
```

Port Already In Use

If port 5173 is occupied, Vite will automatically try the next available port (5174, 5175, etc.). Check the terminal output for the actual URL.

> **TIP: Apple Silicon Macs (M1/M2/M3)**
> Node.js v18+ has native Apple Silicon support. If you experience issues with older packages, try running Terminal through Rosetta: Right-click Terminal > Get Info > check 'Open using Rosetta.'

4. Windows Setup Guide

4.1 Install Node.js

Node.js is the JavaScript runtime needed to install dependencies, run the development server, and build the VIOE application.

1. Open your web browser and navigate to https://nodejs.org
2. Click the LTS (Long Term Support) download button labeled "Recommended For Most Users."
3. Run the downloaded .msi installer.
4. Follow the setup wizard:
   - Accept the license agreement
   - Keep the default installation path (C:\Program Files\nodejs\)
   - Ensure "Add to PATH" is checked (it is by default)
   - On the Tools for Native Modules page, check the box to install build tools if offered
5. Click Install, then Finish when complete.
6. Open a NEW Command Prompt or PowerShell window (existing ones won't have the updated PATH) and verify:

```
node --version
npm --version
```

You should see version numbers like v20.x.x and 10.x.x respectively.

> **WARNING: Important: Open a New Terminal**
> After installing Node.js, you MUST open a new Command Prompt, PowerShell, or Windows Terminal window. Existing terminal windows will not recognize the 'node' or 'npm' commands until you open a fresh one.

4.2 Install Git

Git is required to clone the VIOE project repository.

1. Download Git for Windows from https://git-scm.com/download/win
2. Run the installer and follow the setup wizard:
   - Select Components: keep defaults, optionally add Windows Explorer integration
   - Default Editor: choose your preferred editor (VS Code recommended, select "Use Visual Studio Code as Git's default editor")
   - PATH Environment: select 'Git from the command line and also from 3rd-party software' (recommended)
   - HTTPS Transport: select 'Use the OpenSSL library'

  - Line Endings: select 'Checkout Windows-style, commit Unix-style line endings'
  - All other options: keep defaults

3. After installation, configure your identity in a new terminal:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@company.com"
```

4. Verify installation:

```
git --version
```

4.3 Install a Code Editor

We recommend Visual Studio Code (VS Code) for its excellent JavaScript/React support.

1. Navigate to https://code.visualstudio.com
2. Download the Windows installer.
3. Run the installer and follow the prompts:
    - Accept the license agreement
    - Check "Add to PATH" (important for command-line use)
    - Check "Register Code as an editor for supported file types"
    - Check "Add Open with Code action to Windows Explorer"
4. Launch VS Code after installation completes.

> **TIP: Recommended VS Code Extensions**
> Install these extensions for the best VIOE development experience:
> - ES7+ React/Redux/React-Native Snippets
> - Tailwind CSS IntelliSense
> - ESLint
> - Prettier - Code Formatter

4.4 Clone and Configure the Project

1. Open PowerShell or Command Prompt and navigate to your workspace:

```
cd C:\Users\YourName\Projects
mkdir Projects    # (if it doesn't exist)
cd Projects
```

2. Clone the VIOE repository (or extract the ZIP archive):

```
# If using Git:
git clone <repository-url> vioe
cd vioe

# If using the ZIP archive:
# Extract VIOE_UAT.zip using Windows Explorer
# or use PowerShell:
Expand-Archive VIOE_UAT.zip -DestinationPath vioe
cd vioe
```

3. Create the environment configuration file:

```
copy .env.example .env
```

4. The default .env is pre-configured for demo/mock mode. No changes are needed for local testing.
5. Install project dependencies:

```
npm install
```

This will download all required packages. The process requires internet access and may download approximately 200 MB of packages.

6. Start the development server:

```
npm run dev
```

7. Open your web browser and navigate to the URL shown in the terminal (typically http://localhost:5173).

> **NOTE: First Launch**
>
> The application starts in demo mode by default. On the login page, click any of the quick login buttons (Admin, Manager, Analyst, Viewer) to explore the application with different permission levels.

4.5 Verify Installation

Run the following commands in PowerShell or Command Prompt to confirm everything is set up:

```
# Check tool versions
node --version        # Should show v18.x or v20.x
npm --version         # Should show v9+ or v10+
git --version         # Should show v2.30+

# Run the test suite
npm run test:run      # All tests should pass

# Build for production
npm run build         # Should complete without errors
```

4.6 Common Windows Issues

'node' is not recognized as an internal or external command

This means Node.js is not in your system PATH. Solutions:
1. Close ALL terminal windows and open a new one.
2. If it still doesn't work, re-run the Node.js installer and ensure 'Add to PATH' is checked.
3. Manually add C:\Program Files\nodejs\ to your system PATH via System Properties > Environment Variables.

Windows Defender or Antivirus Blocking Installation

Some antivirus software may flag npm package installations. If npm install hangs or fails:
1. Temporarily add your project folder to the antivirus exclusion list.
2. In Windows Security > Virus & Threat Protection > Manage Settings, add an exclusion for your project directory.

Execution Policy Prevents Running Scripts

If PowerShell blocks npm scripts, run PowerShell as Administrator and execute:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Long Path Issues

node_modules folders can have deeply nested paths that exceed Windows' 260-character limit. Enable long path support by running PowerShell as Administrator:

```
# Enable long paths in Windows
New-ItemProperty -Path "HKLM:\SYSTEM\CurrentControlSet\Control\FileSystem" `
  -Name "LongPathsEnabled" -Value 1 -PropertyType DWORD -Force
```

WARNING: Administrator Rights

Some installation steps require Administrator privileges. Right-click the terminal application and select 'Run as administrator' when needed. For day-to-day development, regular user permissions are sufficient.

5. Post-Setup Checklist

Use this checklist to confirm that your environment is fully configured. Each item should be verified before you begin working with the VIOE application.

Tools & Versions

- □ Node.js installed - node --version shows v18.x or v20.x
- □ npm installed - npm --version shows v9+ or v10+
- □ Git installed - git --version shows v2.30+
- □ Code editor installed (VS Code or equivalent)
- □ Modern web browser available (Chrome, Edge, or Firefox)

Project Configuration

- □ VIOE project cloned or extracted to local directory
- □ .env file created from .env.example
- □ npm install completed successfully (no errors)

Application Verification

- □ npm run dev starts the server without errors
- □ Application loads in browser at the displayed URL
- □ Login page is accessible and demo login buttons work
- □ Dashboard displays correctly after login
- □ npm run test:run passes all tests
- □ npm run build completes without errors

Optional But Recommended

- □ VS Code extensions installed (ESLint, Tailwind CSS IntelliSense, Prettier)
- □ Git identity configured (user.name and user.email)
- □ Tested role switching: logged in as Admin, Manager, Analyst, and Viewer

> **TIP: All Checks Passed?**
> If every item above is checked, your environment is fully set up. You are ready to develop, test, and review the VIOE application. Refer to the project README for additional development workflows.

6. Troubleshooting & Support

Common Errors and Solutions

Error / Symptom	Solution
npm install fails with ERESOLVE conflicts	Run: npm install --legacy-peer-deps This resolves dependency version conflicts.
Port 5173 already in use	Vite auto-selects the next port. Check terminal output for the actual URL. Or stop the other process using the port.
Blank page after login	Clear browser cache (Ctrl+Shift+Delete), then hard refresh (Ctrl+Shift+R). Check browser console for errors (F12 > Console tab).
Build fails with out of memory	Increase Node.js memory: NODE_OPTIONS=--max-old-space-size=4096 npm run build
npm ERR! code EACCES (macOS)	Do NOT use sudo. Reconfigure npm global directory: mkdir ~/.npm-global && npm config set prefix '~/.npm-global'
'node' not recognized (Windows)	Open a NEW terminal window. If still failing, reinstall Node.js with 'Add to PATH' checked.

Where to Get Help

- Project Lead / Tech Lead: Reach out via your team's communication channel (Slack, Teams, email).
- Internal Wiki: Check the project's Confluence or Notion documentation for additional guides.
- Node.js Documentation: https://nodejs.org/en/docs/
- Vite Documentation: https://vitejs.dev/guide/
- React Documentation: https://react.dev/

NOTE: Reporting Setup Issues

If you encounter an issue not covered in this guide, please report it to the project lead with the following information:

1. Your operating system and version
2. Node.js version (node --version)
3. The exact error message or screenshot
4. The command that caused the error

7. Conclusion

You have now completed the full setup of the VIOE development environment. Your workstation is configured to run, test, and build the application locally.

Best Practices

- Keep Node.js updated to the latest LTS version for security and performance.
- Run npm install whenever you pull new changes from the repository to ensure dependencies are current.
- Use npm run test:run before committing changes to verify nothing is broken.
- Use npm run build periodically to catch production build issues early.
- Never commit the .env file to version control. Use .env.example for sharing configuration templates.
- Keep your code editor extensions updated for the best development experience.

Application Quick Reference

Command	Description
`npm run dev`	Start the development server with hot reload
`npm run build`	Create a production-ready build in the /dist folder
`npm run preview`	Preview the production build locally
`npm run test`	Run tests in watch mode (re-runs on file changes)
`npm run test:run`	Run all tests once and exit
`npm run test:coverage`	Run tests with code coverage report
`npm run lint`	Check code for style and quality issues

Demo Login Accounts

In demo/mock mode, use these accounts to test different permission levels:

Role	Email	Password	Access Level
Admin	admin@vioe.demo	demo	Full access to all features
Manager	manager@vioe.demo	demo	Team management, reports
Analyst	analyst@vioe.demo	demo	Triage, tasks, analysis
Viewer	viewer@vioe.demo	demo	Read-only access

Thank you for completing this setup guide. If you have questions or suggestions for improving this document, please contact the project team.