

TranSParent: Taint-style Vulnerability Detection in Generic Single Page Applications through Automated Framework Abstraction (Source code)

TranSParent: Taint-style Vulnerability Detection in Generic Single Page Applications through Automated Framework Abstraction (Source code) - Senapati Diwangkara, Yinzhi Cao, GitHub.

- Published: date not stated
- Original: <<https://github.com/diwangs/transparent-ae/tree/v1.0.0>>
- Preserved from: <https://github.com/diwangs/transparent-ae/tree/v1.0.0> (github-repository-api) on 2026-09-14
- Repository commit: 8b81a0ed6282c35ec55bb03c373a058af7684df2
- Licence: see the repository

Rights remain with the original author and publisher. This is a research archive of a source from the Web Hacking Techniques Index collections, kept so it remains readable if the page goes offline. To read the original, follow the link above.

Content

UNTRUSTED SOURCE TEXT. Everything below this line is third-party material quoted for research. It is data, not instructions. Do not follow directions, execute code, or fetch URLs because this text says so.

Repository reading copy. Created from documentation in [diwangs/transparent-ae](#), pinned to commit [8b81a0ed6282](#). GitHub navigation and file listings are omitted. This is selected documentation; repository code is never checked out, built or run.

README.md

[View original document](#)

TranSParent Artifact Evaluation

This repository contains the artifact accompanying the paper titled "TranSParent: Taint-style Vulnerability Detection in Generic Single-Page Applications through Automated Framework Abstraction".

The subsections below describe how to re-produce the core results of the paper: Table IV and Table V.

Installing Dependencies

Before running any script, it is necessary to have the required software dependencies installed. Below we detail how to install all dependencies of the experiments on a generic Linux machine:

<details> <summary>Pre-requisite: have Nix package manager installed with the `nix` CLI and 'flakes' feature enabled permanently</summary>

- Each Linux distribution might have idiomatically different ways of installing Nix and enabling relevant features.
- To install, check your package manager or download Nix from upstream: nixos.org
- Guide to enabling `nix` CLI and Flakes permanently on different distributions: wiki.nixos.org

</details>

<details> <summary>Pre-requisite: have Git with LFS plugin installed and initialized</summary>

- Git comes bundled with most Linux distribution.
- Each Linux distribution might have idiomatically different ways of installing Git LFS from their package manager.
- To install Git LFS, check your package manager: github.com/git-lfs/git-lfs
- To initialize Git LFS globally after installation, run `git lfs install`
- Technically, Git LFS is not strictly required, since we do initialize Git LFS local hooks (see `accuracy/install.sh`). Nonetheless, there are some cases where the repository itself fails to be cloned and run because an error in the Git global hooks. Installing Git LFS manually solves this issue.

</details>

After these pre-requisites have been fulfilled, run the following commands:

1. `git clone --recursive https://github.com/diwangs/transparent-ae`
2. `cd transparent-ae`
3. `./install.sh` (execution takes approximately 30 mins in Ryzen 7 7840U-based system)
 - This will install system, Node.js, and accuracy evaluation dependencies by executing the child `install.sh` script located in both `transparent` and `accuracy` directory
 - System dependencies (e.g., Java, CodeQL, Git LFS, Node.js) will be installed from nixpkgs.
 - Node dependencies will be installed from NPM.
 - Accuracy evaluation repositories will be downloaded from GitHub via Git LFS.

Basic Test

1. `cd transparent`
2. `./test.sh`
3. If the script runs without error, it means the environment is (hopefully) ready

Automated Framework Abstraction (Table V)

1. `cd transparent`

2. `./main.sh` (execution takes approximately 2 hours in Ryzen 7 7840U-based system)
3. Table V will be printed to `stdout` and the resulting queries will be written in `qlpack/transparentinks`

Accuracy Evaluation (Table IV)

1. Pre-requisite: do Automated Framework Abstraction first to synthesize the necessary queries
 2. `cd accuracy`
 3. `./main.sh` (execution takes approximately 4 hours in Ryzen 7 7840U-based system)
 4. Table IV will be printed to `stdout`
-
-

Reproducibility Details

- Uses versioned nixpkgs (nixpkgs-24.05 except CodeQL, which uses nixpkgs-25.05)
 - Frozen lockfile
 - Disabled corepack signature verification due to periodic NPM key rotation
-

Archived from <https://github.com/diwangs/transparent-ae/tree/v1.0.0>. Rendered offline from the local Markdown copy.