USENIX Security '25 Artifact Appendix: High Stakes, Low Certainty: Evaluating the Efficacy of High-Level Indicators of Compromise in Ransomware Attribution

Max van der Horst
Delft University of Technology

Ricky Kho
Sogeti

Olga Gadyatskaya
Leiden University

Michel Mollema
Northwave Cybersecurity

Michel van Eeten
Delft University of Technology

Yury Zhauniarovich
Delft University of Technology

A Artifact Appendix

A.1 Abstract

The artifacts included for this paper are located in a Zenodo repository. The Zenodo repository contains the individual appendices of the paper, such as the research protocol, the codebook, the observed techniques per investigated threat actor, and the CISA reports. Additionally, the gathered data on the individual threat actors as well as the code to analyze this data, is included. To reproduce the results of the paper, the code in `code.zip` can be run by following the steps in `README.md`.

A.2 Description & Requirements

A.2.1 Security, privacy, and ethical concerns

The code included in the artifacts can exclusively process the included data and will not pose any security concerns for the users. All data has been anonymised where applicable, and all artifacts have been produced following the ethical considerations described in the paper.

A.2.2 How to access

All artifacts have been included in the Zenodo repository on https://zenodo.org/records/14732551.

A.2.3 Hardware dependencies

None

A.2.4 Software dependencies

The artifacts can be used on any operating system. However, Python 3.11+ and Python Poetry[1] is used. The file `pyproject.toml` includes all individual dependencies that are installed by Poetry. To install these, follow the steps included in the `README.md` file. Moreover, either the VSCode IDE or Jupyter Notebook should be installed, though the notebook is best evaluated using VSCode.

[1] https://python-poetry.org

A.2.5 Benchmarks

None.

A.3 Set-up

A.3.1 Installation

Before the file `data_analysis.ipynb` can be run, the system must have Python 3.11+ available and be able to run Poetry – a dependency manager for Python projects. Poetry can be installed through pipx: `pipx install poetry`. After installation of Poetry, the dependencies can be installed with the command `poetry install -no-root`. The notebook can best be evaluated in VSCode. If the reviewer chooses to use Jupyter Notebook (which is another option), appropriate dependencies have to be installed to support it.

A.3.2 Basic Test

After opening the notebook, the first cell in the notebook for the definition of the constants can be run to verify that the system is ready to execute the other cells.

A.4 Evaluation Workflow

The artifacts can be evaluated using the following steps.

1. Download the artifacts `code.zip` and `data.zip` from the Zenodo repository.
2. Ensure that the VSCode IDE is installed.
3. Ensure that all dependencies have been installed using Python Poetry.

4. Open `data_analysis.ipynb` in VSCode to open the data analysis notebook.
5. Ensure the notebook has access to the contents of `data.zip` in `../data` (relative to the working location of the notebook).
6. Execute all cells in the data analysis notebook.
7. The analysis results will be stored in `../results`, with the LaTeXtables being located in the `tabs/` subdirectory.

A.4.1 Major Claims

The major claims in this section are mainly discussed in Sections 4.2 and 5.2 of the paper. As explained in the Ethics section at the end of the paper, we only include data on the quantitative analysis for review as we cannot share some data artifacts (transcripts and reports).

High Overlap Across Different RTAs. The paper mentions a high overlap of TTPs across different RTAs. This is proven by the overlap similarity matrix in Table 6 of the paper. This table can be generated by following the "Analysis" section of the notebook, which is computed by calculating the overlap similarity for all data points.

Inconsistent TTP Usage Within the Same RTA. Attacks attributed to the same actor only shared, on average, about 37% of their TTPs. This is proven by following the steps in the "Similarity between Variants of the same TA" section in the notebook to generate the content of Table 5 in the paper.

Negative Silhouette Scores. By executing the first cell of the "Company Data" section in the notebook, the silhouette score calculated based on the clusters of RTAs is negative. This indicates that TTP sets for different ransomware actors do not define clearly separable clusters.

Coverage Gaps and Fragmentation. By executing the code belonging to the company data and the CISA data in the notebook, it can be seen that there are significantly differing overlap scores for the same RTA. Given that this is a consistent result for various actors and not an anomaly, it can be suggested that not one party has a complete overview of the data belonging to RTAs.

A.4.2 Experiments

Please find the steps to reproduce the results for the major claims below. Each section of the results is preceded by the first five steps as described in the Evaluation Workflow.

1. Download the artifacts `code.zip` and `data.zip` from the Zenodo repository.
2. Ensure that the VSCode IDE is installed.
3. Ensure that all dependencies have been installed using Python Poetry.
4. Open `data_analysis.ipynb` in VSCode to open the data analysis notebook.
5. Ensure the notebook has access to the contents of `data.zip` in `../data` (relative to the working location of the notebook).

High Overlap Across Different RTAs [Human time: 5 minutes, compute time: < 1 second]

1. Ensure the execution of the cells in the "Analysis" section of the notebook to define the functions.
2. Execute the cell "Similarity between Different Families in Company Reports".
3. The notebook will output the mean overlap similarity for different RTAs in the company reports, and a TeX table will be generated in the file `overlap_sim_of_tas_comp.tex`.
4. Overlap similarity is expressed as a value between 0 and 1, with 0 meaning that there is no overlap and 1 indicating identical values.

Inconsistent TTP Usage Within the Same RTA [Human time: 5 minutes, compute time: < 1 second]

1. Ensure the execution of the cells in the "Analysis" section of the notebook to define the functions.
2. Execute the cell "Similarity between Variants of the Same TA".
3. The notebook will output the results in the file `tavar_sim_table_comp.tex`.
4. The overlap similarity in the table is expressed as a value between 0 and 1, with 0 meaning that there is no overlap and 1 indicating identical values.

Negative Silhouette Scores [Human time: 5 minutes, compute time: < 1 second]

1. Ensure the execution of the cells in the "Analysis" section of the notebook to define the functions.
2. Execute the first cell of the section "Company Data".
3. The notebook will output the silhouette scores using both Euclidean and cosine distance metrics.
4. Silhouette score is a value between -1 and 1. A lower score indicates poorly defined clusters and a higher score indicates well-defined clusters. The silhouette score is negative for our data, indicating poorly defined clusters.

Coverage Gaps and Fragmentation [Human time: 5 minutes, compute time: < 1 second]

1. Ensure the execution of all cells in the notebook up until the "Comparison of Techniques in CISA and Company reports" cell.
2. The notebook will output the mean overlap similarity and generate a TeX table in the file `comp_cisa_sim_analysis.tex`.
3. Comparing the table generated in this step to the `overlap_sim_of_tas_comp.tex` table from the first major claim, it can be seen that the overlap similarities per TA are very different. This indicates a coverage gap and fragmentation.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2025/.