Measuring Ease of Reuse for Security Research Artifacts with Results-Reproduced Badges

Jelena Mirkovic
mirkovic@isi.edu
USC Information Sciences Institute
Marina del Rey, California, USA

Qishen Sam Liang
qishenl@alumni.usc.edu
USC Information Sciences Institute
Marina del Rey, California, USA

David Balenson
balenson@isi.edu
USC Information Sciences Institute
Marina del Rey, California, USA

Abstract

Reproducibility and reuse of scientific artifacts are critical for scientific progress. Artifacts that are easy to reuse help researchers compare results and build on prior work. Security conferences promote reproducibility through artifact evaluation and badging. The highest-level badge – results-reproduced – indicates that independent evaluators were able to use an artifact to reproduce key claims in the corresponding paper.

We perform a small-scale study to quantify the ease of reuse of security artifacts that received results-reproduced badges. We evaluate 97 artifacts from two conferences – PETS and ACSAC – from a recent year and from 2020. Our study finds that, despite passing stringent evaluation, many artifacts exhibit usability issues. Around 12% are not findable or lack key components. Among those that are complete, 27% require resources that may not be easily obtained, and 22% fail to run within an hour, leaving only 43 artifacts (44%) successfully reused. Artifacts are packaged using a variety of frameworks, such as Docker/VM images, Python environments, Rust, Go, Popper, Poetry, and Bazel; 38% of artifacts use more than one approach. Our findings highlight the need for community-wide standards for artifact packaging and evaluation to improve artifact findability and ease of reuse.

CCS Concepts

• **General and reference** → **Evaluation**.

Keywords

Reproducibility, Reuse, Ease-of-use, Artifacts, Badges

ACM Reference Format:
Jelena Mirkovic, Qishen Sam Liang, and David Balenson. 2026. Measuring Ease of Reuse for Security Research Artifacts with Results-Reproduced Badges. In *ACM Conference on Reproducibility and Replicability (ACM REP '26), July 20–22, 2026, Delft, Netherlands.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3820002.3828598

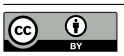
1 Introduction

Reproducible experiments and reusable scientific tools are critical for scientific progress. They enable a researcher to reproduce prior work, to compare their own work against related work in a common setting, or to build on prior work, extending and improving it to advance the science. Reusable scientific tools, such as code and datasets, greatly speed up an individual researcher's progress, because they do not have to start from scratch.

Many scientific fields suffer from reproducibility challenges, including medicine [46], social science [44], computer systems [17] and cybersecurity [50, 51]. There are many reasons why this may happen. A published paper's authors may refrain from publishing some relevant data or tools to preserve their competitive edge in the community, or these data/tools may be sensitive or proprietary. Even when authors embrace open science, full reproducibility may be hard. Scientific tool development and usage in experiments may involve work of many over an extended period of time, leading to a successful publication. Thus it may be challenging to gather necessary information even within the author team to facilitate reproducibility. An additional challenge lies in maintaining reproducibility over time. Numerous factors can degrade it, such as students graduating, faculty moving or retiring, code and data repositories being removed from public archive, and OS, programming language and third-party software evolution.

Security conferences and journals promote reproducibility by running artifact evaluation. Research artifacts (code, datasets, experiment scripts) are solicited after a paper's acceptance and evaluated by a dedicated program committee, consisting of junior researchers in the field. Successfully evaluated artifacts receive badges, which are publicized on the conference Web page. Badges often include some variants of Available (code/data are publicly available for download), Functional (code installs and runs) and Results-reproduced (code/data can be used to reproduce key results). Badges are usually cumulative, e.g., one cannot obtain a results-reproduced badge without meeting the criteria for the available and functional badges. The highest-level badge, Results-reproduced, is awarded to an artifact that could be ran by evaluators to independently reproduce key results reported in the corresponding paper. Currently, many cybersecurity venues run artifact evaluation.

In this paper we quantify how easily a researcher could *reuse* successfully-evaluated *code* artifacts. We focus on code artifacts that received Results-reproduced badges, because they are held to the highest standard. They have been installed and ran by multiple evaluators, on their own platform, and had produced results that match those in the published paper. This makes them appealing candidates for reuse. We evaluate 97 artifacts from two security venues – PETS and ACSAC – from two select years – one recent and one six years in the past. Our work makes the following **contributions**: (1) We formulate a methodology for large-scale evaluation of badged research code artifacts for findability and ease of reuse. (2) We apply our methodology to badged artifacts from two years and two security venues. (3) We find serious findability issues: out of 97 artifacts, 86 are found based on artifact links, 6 are found



ACM REP '26, Delft, Netherlands

ACM ISBN 979-8-4007-2778-8/2026/07
https://doi.org/10.1145/3820002.3828598

with additional effort. Among the 62 candidates that can be ran on commonly available hardware and software, only 45 contain a pointer to the corresponding paper publication, and only 41 papers contain a pointer to their accompanying artifact. (4) We find additional accessibility issues: out of the 85 complete artifacts, 23 have hardware or software requirements that may not be easily met. (5) We find serious reuse issues: out of 62 artifacts that we have evaluated, 43 were successfully installed and ran, 23 within 10 minutes, 9 within 30 minutes and 11 within 1 hour. Out of the 62 artifacts, 48 included a narrative (e.g., README.md) that facilitates artifact understanding and reuse. (6) We find a large diversity of packaging formats, which can impact likelihood of reuse success.

Our work sheds light on significant remaining obstacles to artifact reuse: findability, artifact completeness, access to infrastructure, and standardized packaging. We suggest that artifact evaluation processes be enriched by community-wide standardization of artifact packaging and badging criteria, to improve findability, completeness and ease of reuse.[1]

We release our dataset, data processing scripts, artifact installation scripts and resource specifications at https://doi.org/10.5281/zenodo.21089569.

2 Our Focus and Goals

We adopt standard terminology consistent with prior literature, including the National Academies report on reproducibility and replicability [46]. *Repeatability* refers to obtaining the same results by the same team using the same experimental setup, including the same data, code, and environment. *Reproducibility* refers to obtaining the same results using the original artifacts (e.g., data, code, and analysis pipeline), potentially by a different team, in order to verify the reported findings. *Replicability* refers to obtaining consistent results when addressing the same scientific question using new data, experimental conditions, or implementations, thereby testing the robustness and generalizability of the findings.

While these concepts focus on validating and generalizing scientific results, they do not fully capture how easily artifacts can be used by others in practice. We therefore define *reusability* of an artifact as the ability of a new user to obtain, execute, understand, and adapt a research artifact with reasonable effort to further their own research. This definition is consistent with ACM's Artifacts Evaluated – Reusable badge, which applies to artifacts that exceed minimal functionality and are carefully documented and well-structured so that reuse and repurposing are facilitated [3]. In this sense, reusability extends reproducibility by emphasizing usability and portability, and it enables replicability by lowering the effort required to apply artifacts in new settings.

In this work we imagine a junior researcher, whom we denote a "reuser", who looks to reproduce or extend prior work. We focus on *code* artifacts, which have been found challenging to reuse in past work [50, 51]. We seek to quantify how easy it would be for a reuser to find, obtain, understand, successfully install, and run a research artifact that received a results-reproduced badge. We focus on this highest-level badge, because it represents artifacts that passed the highest bar – they were successfully installed and run by independent evaluators, and they produced results matching those in the corresponding published paper.

We report results from two conferences – The Privacy Enhancing Technologies (PETS) Symposium and the Annual Computer Security Applications Conference (ACSAC). Both conferences have run artifact evaluation for many years. PETS focuses on privacy-preserving technologies, while ACSAC welcomes both system security and privacy works, but has traditionally published more system security-oriented works. We investigate ease-of-reuse of both recent artifacts (PETS 2025 and ACSAC 2024), and of those from six years ago (PETS 2020 and ACSAC 2020) to evaluate if reusability declines over time.

3 Artifact Evaluation

We briefly describe the process of artifact evaluation (AE) at security conferences. Authors of accepted papers are invited to submit their artifacts (code, datasets) for artifact evaluation and choose desired badges. Badge levels include some variation of Available (code/data are publicly available for download), Functional (code installs and runs) and Results-reproduced (code/data can be used to reproduce key results), and are usually cumulative. Artifact evaluation chairs recruit artifact evaluators, usually among junior researchers, and assign each artifact to at least two evaluators. Evaluation process and criteria can vary from venue to venue, and from year to year, as the community learns what works well, and evolves its processes. In some cases, the AE may seem more focused on verifying published results than on producing high-quality artifacts for the community. For example, some AE efforts may accept private artifacts and award higher badges without the available badge. Some AE efforts may accept evaluation on author-owned infrastructure, even when such infrastructure may be unavailable to the future reuser. Some AE efforts may encourage packaging of artifacts as VM or Docker images, which facilitates easy evaluation, but it may not provide sufficient foundation for others to build on and extend the artifact in the future.

Overall, AE efforts have been invaluable to the research community, because they have produced incentives and created pipelines for wider release of research artifacts. We hope our work shines light on the overlooked challenges that should be addressed to facilitate wider *reuse* of this treasure of artifacts.

4 Related Work

Definitions and conceptual foundations. Reproducibility and replicability have been widely studied across scientific disciplines, most notably in the National Academies of Sciences (NAS) report on reproducibility and replicability in science [46], which has standardized definitions of repeatability, reproducibility, and replicability. Olszewski et al. [49, 52] systematize definitions of reproducibility and replicability, and propose frameworks, such as the Tree of Validity, to better reason about these qualities in context of cybersecurity.

[1]This work was conducted in the context of the SPHERE project (Security and Privacy Heterogeneous Environment for Reproducible Experimentation), which is supported by the U.S. National Science Foundation under award number 2330066. The opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation or other supporting organizations.

Operationalizing reproducibility in computer systems and security. The computer systems community developed mechanisms to operationalize reproducibility in practice through Artifact Evaluation Committees (AECs) and artifact badging. These efforts reflect a practical response to long-standing challenges in validating experimental results in systems research, including limited artifact availability and difficulties in reproducing complex experimental setups. Over time, artifact evaluation practices were adopted and adapted by the computer security community, where they are now common at all major venues.

While AECs have helped normalize reproducibility as part of the publication process, they largely equate reproducibility with artifact availability and executability *at the evaluation time* and potentially *with help of the original author*. As a result, they address only part of the broader challenges identified in the literature, leaving open questions about how effectively artifacts can be reused in practice *over time* and *without interaction with the artifact's author*. Our study explores this gap.

Empirical studies of reproducibility. Early work by Collberg et al. focused on systems conferences and examined the availability, buildability, and executability of research artifacts, finding that many papers lacked sufficient artifacts to enable reproduction and that a substantial fraction of artifacts failed to build or run even when provided [17]. These studies established a baseline understanding of the practical barriers to reproducibility in systems research.

More recently, similar empirical analyses have been carried out in the computer security community. Olszewski et al. conducted systematic studies of artifacts across major security venues, evaluating availability, executability, and the ability to reproduce reported results. Their analysis of machine learning security papers found that 60% of papers did not provide code artifacts, and that 56% of the remaining artifacts failed to execute, with half of the successful executions producing results that did not fully reproduce corresponding publication claims [51]. A broader study of applied security conferences similarly reported low rates of runnable (35%) artifacts, even in venues with established artifact evaluation processes [50]. Olszewski et al. find that AE improves the number of available artifacts, and the chance of compilation success, but they could not confirm that it improves artifact quality. Out of 22 artifacts with Reproduced badges that they evaluated, 24% did not run, which is comparable to 30% of artifacts in our study.

Works of Collberg et al. and Olszewski et al. focus on a wide variety of research artifacts. They cover long time periods that predate full adoption of AE efforts at security conferences and examine a broader set of questions around reproducibility. Our work complements these prior works [17, 50, 51] by focusing on code artifacts that received Results-reproduced badges, which are expected to be readily accessible and of higher quality. By measuring their ease of reuse, we provide new insights into the practical usefulness of badged research artifacts.

From reproducibility to reuse. Across both systems and security, prior work has primarily focused on whether artifacts exist, can be executed, and produce results consistent with published claims. These are necessary conditions for reproducibility, but they do not capture how artifacts are reused in practice. Existing works do not evaluate the effort required to reuse artifacts, nor how this effort changes over time or how it may be influenced by artifact packaging.

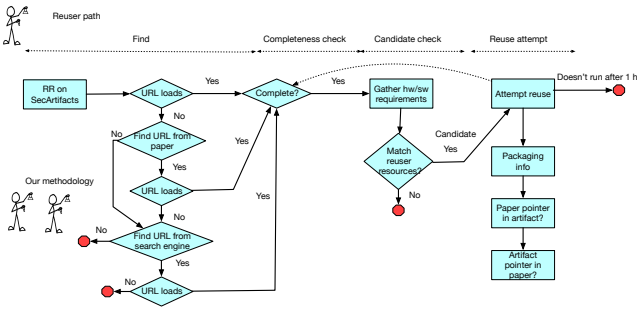


Figure 1: Reuser path and our methodology. Red octagons denote abortion points in the workflow.

In this work, we address this gap by defining and measuring the ease of reuse of artifacts that have already achieved results-reproduced status. We capture the effort required for a new user to obtain, execute, and understand an artifact. We also evaluate how this property may degrade over time and how it may depend on the artifact packaging choices.

5 Dataset and Methodology

Our evaluation methodology follows an imaginary reuser that looks to reuse an artifact. First they must *find* a candidate artifact, either by reading a related paper and finding a link to the artifact, or by browsing the artifact evaluation pages of relevant conferences. The reuser visits the artifact link and explores the contents. The artifact may be *incomplete*, for example it may need a dataset that was not released publicly, and the reuse attempt stops. If the artifact is complete, the reuser examines its *hardware and software requirements*. If the reuser has access to the necessary hardware and software the given artifact is a *candidate* for reuse. The reuser attempts to reuse it and notes the level of difficulty.

We focus on four aspects of artifact reuse from the perspective of an imaginary reuser:

- *Availability.* Can the reuser find the artifact and obtain all required components?
- *Executability under standard conditions.* Can the reuser execute the artifact in a broadly available environment?
- *Ease of reuse.* How much effort is required for the reuser to install and run the artifact?
- *Impact of packaging.* How do packaging choices affect the reuser's ability to execute and reuse the artifact over time?

Our evaluation methodology follows the imaginary reuser path illustrated in Figure 1. Two evaluators performed all steps except the reuse attempt, and they discussed and resolved any differences (e.g., around hardware requirements, findability, etc.). Reuse attempts were conducted independently, with one evaluator assigned to PETS and the other to ACSAC artifacts.

Finding and accessing the artifacts. We harvest artifact information from the Security Research Artifacts repository[2], referred to as "SecArtifacts" for short. This repository is populated directly

[2] https://secartifacts.github.io/

from artifact pages of the relevant conferences. We select the most recent year we could find – 2025 for PETS and 2024 for ACSAC [3]. We also select artifacts from the year 2020, to evaluate if ease-of-reuse degrades over time. ACSAC has performed artifact evaluation since 2017 and PETS since 2020. Since badging standards differ between conferences and even from year to year at the same conference, we show the badge descriptions used at our select venues and years in Table 2 in the Appendix. The third column shows if we included the selected artifacts in our evaluation. We focused on artifacts with *Reusable* badge from ACSAC 2020, *Code Reproducible* from ACSAC 2024 and *Artifact Reproduced* from PETS 2025. Since PETS 2020 only supported one category of artifacts, whose criteria resembled a Functional badge, we manually visited each artifact's repository and checked for presence of any code or instructions that talked about reproducing results from the corresponding paper. If such code or instructions were present, we included the artifact in our evaluation.

For each included artifact we manually visit the URL listed on the SecArtifacts. If that URL does not load, we attempt to find an alternative URL by searching in the corresponding published paper, and failing that we use a search engine with keywords consisting of the paper's title and the words "github" and "gitlab". If we find an alternative URL and can access it, we use it in the rest of the evaluation.

We perform additional findability checks on our identified candidates. We check if they contain a pointer to the corresponding paper, which can come in the form of the paper's title and the publication venue, the URL, or the citation. We further check the corresponding paper from the official proceedings, to see if it specifies the URL of the artifact, and if the URL matches the one at SecArtifacts.

It is difficult to fully assess completeness at this early stage. We refine our early assessment by removing from the complete pool those artifacts that fail to run due to missing artifact-specific files (e.g., a dataset or processing script).

Identifying evaluation candidates. We assume a reuser has access to some public research infrastructure, such as CloudLab [23], Chameleon [33] or SPHERE [43]. Such infrastructures can accommodate artifacts that use Linux or a VM with qcow2 image, do not require a specific CPU model, or paid AI APIs. Because public infrastructures differ in whether they host GPU nodes, and whether using GPU nodes requires prior approval, we excluded artifacts that required GPU nodes. We used SPHERE research infrastructure [43] in our evaluation, which enables us to deposit successfully reused artifacts into the SPHERE's artifact library, and facilitates easy reproduction of our work by others.

To preliminarily identify the hardware and software requirements of each artifact, we query a free version of Gemini AI (gemini-3-flash), using the contents of either the artifact's README file or its ARTIFACT-EVALUATION file (see Appendix for the prompt). To ensure accuracy and address potential AI errors, the authors manually verified the requirements for all artifacts. We found that AI mined most requirements correctly. It did not exhibit hallucinations, although it did occasionally misinterpret some recommended specifications as mandatory. We compared manually-verified artifact requirements with SPHERE's resource catalogue. Those artifacts whose requirements pass our resource check become *evaluation candidates*.

Identifying packaging. We identify packaging across several popular formats, and include "other" category for those artifacts that use a less popular format. We say that an artifact *uses* a given packaging approach if it provides a full path to installation with that approach. For example, an artifact that offers a Docker image, within which one should run a Bash script will be labeled as "Docker-image". Conversely, an artifact that offers a Docker image, and also offers an option to run Bash scripts for installation, will be tagged as "Docker-image" and "Bash script". Artifacts that list Bash commands in a README file are also tagged as "Bash script".

We also evaluate whether an artifact's install is fully scripted, or requires some manual activities, such as running additional commands inside of a Docker image. We further examine the artifact's narrative (e.g., README.md) to evaluate if it explains the artifact's purpose and intended use, which earns it "built-for-reuse" label. This check extends the Olszewski et al.'s check for "built for reproducibility" [50] by further requiring the artifact's narrative to goes beyond artifact evaluation goals, and to also narrate possible future reuse or modification options.

Measuring ease of reuse. Each artifact's reuse attempt starts with a fresh allocation of the required resources on SPHERE and artifact download. We attempt to reuse the artifact by installing it on SPHERE and running it according to its instructions. If one installation path fails, we attempt alternative paths. We also attempt to fix minor issues with the artifact. Both evaluators are fairly proficient in Linux, Docker, Python, Bash and CMake, and somewhat familiar with Rust and Go. We complement our knowledge with online AI tools, which we query manually to suggest possible fixes. We abort our attempts after 1 h, which replicates the criteria from prior publications [17, 50, 51]. If we are successful in reusing the artifact, we note the time required by the evaluators–from starting the artifact install on SPHERE to having the artifact completely built and able to run–in three categories: easy (< 10 min), moderately easy (10-30 min) and difficult (30 min - 1 h). This time involves active troubleshooting only, and excludes time to download the artifact onto SPHERE or to allocate SPHERE resources.

6 Results

We summarize our results in Table 1 and we provide per-artifact details in our publicly-released research artifact at https://doi.org/10.5281/zenodo.21089569.

Table 1: Summary of our results per conference and total.

metric/venue	PETS2025	PETS2020	ACSAC2024	ACSAC2020	total
highest-tier badge	41	17	29	10	97
found	41	17	24	10	92
(original+altern.)	(41+0)	(16+1)	(19+5)	(10+0)	(86+6)
complete	40	17	20	8	85
candidate	33	13	12	4	62
reused	25	6	8	4	43
easy	17	5	1	0	23
moderate	2	1	3	3	9
difficult	6	0	4	1	11
scripted	18	13	5	3	39
built-for-reuse	27	11	8	2	48
paper-link-in-art	23	12	7	3	45
art-link-in-paper	21	9	10	1	41
(secart+diff.)	(20+1)	(9+0)	(9+1)	(1+0)	(39+2)

[3] The PETS 2026 and ACSAC 2025 artifacts were published after this work began.

Findability. There are many findability issues with current artifacts: 11 out of 97 artifacts (11.3%) are not found at their official (venue-listed) artifact URL, and 5 (5%) are not found at all. Out of those that are found, 7 (8%) are incomplete. Out of 62 artifacts that are complete and do not require special hardware, GUI or paid APIs, 17 did not have the reference to the accompanying paper and 21 papers did not include a link to the artifact, while 2 included a different link than in their venue-listed URL. The largest cause of reduced findability was use of *open4science*, whose repositories expire by default in 90 days. Thus an artifact may be present during evaluation and expire afterwards, without the authors' intent.

Candidates. Out of 85 complete artifacts, 23 (27%) required hardware or software that did not match our requirements, and were not candidates for attempted reuse. Our candidates include 62 artifacts [1, 2, 4–16, 18–22, 24–32, 34–41, 45, 47, 48, 53–74].

Ease of reuse. Out of 62 candidate artifacts, we successfully reused 43 (69%): 23 (53%) were easy to reuse, 9 (21%) were moderately easy, and 11 (26%) were difficult. Additionally, out of 62 candidate artifacts, 63% (39 artifacts) had fully-scripted installation and 77% (48 artifacts) were documented for reuse.

Causes of failure to reuse. Reuse attempts failed for a variety of reasons, including changes in packaging tools and environments such as Bazel and Docker, changes in languages such as Python and Go, and changes in libraries and dependency ecosystems for Rust and Java. In all cases, we attempted to downgrade or upgrade our installation to address the issues, and we attempted to modify artifact scripts, but not the core artifact code.

These failures illustrate the reality that research artifacts can become outdated as their surrounding software ecosystems change, and that continued reuse may require ongoing maintenance. For example, *in the four months between submission and preparation of the final camera-ready version of this paper, three artifacts that we were able to reuse at submission time had become unusable.* The causes were changes in Rust packages, changes in the Go language, and changes in Java, which we could not address via downgrade actions. We discuss "living artifacts" as a possible solution to artifact upkeep in [42].

Packaging. Docker and various Python packaging approaches (mamba, conda, pyenv, requirements.txt, etc.) were by far the most common choices. Docker appeared in 24 candidate artifacts, 16 of which were successfully reused, while Python packaging approaches appeared in 26 candidate artifacts, 19 of which were successfully reused. Bash scripts were next, used by 16 artifacts, 10 of which were successfully reused. Rust and Go were used in 7 and 5 artifacts, respectively, of which 6 and 3 were successfully reused. Other packaging and build solutions, such as poetry, uv, popper, CMake, VM images, Bazel, and opam, were used in 18 artifacts, 10 of which were successfully reused.

Overall, 39 artifacts used a single packaging approach (29 successfully reused), 14 used two (9 successfully reused), 7 used three (3 successfully reused), and 2 used four approaches (2 successfully reused). The sizes of our samples and their imbalance preclude a deeper study of how packaging choices may impact reuse over time.

Impact of artifact age. 33 out of 45 recent artifacts were successfully reused, compared with 10 out of 17 artifacts from 2020. While artifact age can reduce the likelihood of successful reuse due to evolution in programming languages, operating systems, and software libraries, our sample sizes preclude statistical tests that could conclusively determine how artifact age impacts reuse.

7 Discussion

Limitations. Our study has several limitations. First, our artifact sample is small and limited to two venues across two years. A larger-scale study is needed to draw broader conclusions. Second, it would be good to extend the study to all badged artifacts, or even beyond that to all artifacts associated with published papers, to further study how badging levels may impact artifact quality. We leave this for future work. Third, our assumptions about a reuser's resources may be too strict, which may lead us to prematurely exclude some artifacts. Fourth, our reuse success and time-to-reuse measurements are inherently subjective, and may be influenced by our familiarity with specific packaging approaches or programming languages. A more robust approach would involve multiple evaluators per artifact, at different skill levels, and it would report on all their experiences.

Implications for artifact evaluation. Our findings suggest that artifact evaluation remains a highly manual process, resulting in non-standardized artifacts that are often difficult to reuse. We recommend that AE committees work toward clearer standards for both packaging and badging.

For packaging, standards could include a suggested structure for README files (as in PETS 2025), required metadata fields in artifact submissions, and fully scripted installation where possible. For badging, standards could require that artifacts remain publicly available at a persistent link after evaluation, that they are complete, and that artifact links are included in the corresponding paper. Standardizing the AE process would not only improve reuse, but also facilitate larger-scale, AI-assisted studies of badged artifacts.

8 Conclusion

Traditionally, the reproducibility of artifacts in published computer security research is evaluated at or near the time of publication, while their longer-term reusability is often assumed rather than verified. To examine this gap, we developed a methodology that mimics the path of a researcher attempting to find, obtain, understand, install, and run previously badged artifacts. We applied this methodology to the highest-tier badged artifacts from PETS and ACSAC across two years.

Although the badging systems differ across conferences and years, our results show that even highly evaluated artifacts can face significant barriers to long-term reuse, including findability problems, incomplete artifacts, unmet resource requirements, packaging challenges and programming language and software evolution.

These findings suggest that artifact evaluation should place greater emphasis on persistent availability, complete artifact-paper linking, clearer documentation, and more standardized packaging practices to support reuse beyond the original evaluation period. Future work should extend this methodology to additional venues and years, involve multiple evaluators per artifact, and explore how standardized packaging and metadata could support more scalable, AI-assisted artifact evaluation.

References

[1] Patrick Ah-Fat and Michael Huth. 2020. Protecting Private Inputs: Bounded Distortion Guarantees With Randomised Approximations. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 284–303. doi:10.2478/popets-2020-0053

[2] Ildi Alla, Selma Yahia, Valeria Loscri, and Hossien Eldeeb. 2024. Robust Device Authentication in Multi-Node Networks: ML-Assisted Hybrid PLA Exploiting Hardware Impairments . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 1172–1185. doi:10.1109/ACSAC63791.2024.00094

[3] Association for Computing Machinery. 2026. Artifact Review and Badging – Current. https://www.acm.org/publications/policies/artifact-review-and-badging-current. Accessed: 2026-06-20.

[4] Andreas Athanasiou, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2025. Enhancing Metric Privacy With a Shuffler. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 650–679. doi:10.56553/popets-2025-0081

[5] Brendan Avent, Javier González, Tom Diethe, Andrei Paleyes, and Borja Balle. 2020. Automatic Discovery of Privacy–Utility Pareto Fronts. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (Aug. 2020), 5–23. doi:10.2478/popets-2020-0060

[6] Pierre Ayoub, Romain Cayre, Aurelien Francillon, and Clementine Maurice. 2024. BlueScream: Screaming Channels on Bluetooth Low Energy . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 636–649. doi:10.1109/ACSAC63791.2024.00060

[7] Sofiane Azogagh, Marc-Olivier Killijian, and Félix Larose-Gervais. 2025. A non comparison oblivious sort and its application to k-NN. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (July 2025), 156–169. doi:10.56553/popets-2025-0093

[8] Alexander Bajic and Georg T. Becker. 2020. dPHI: An improved high-speed network-layer anonymity protocol. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 304–326. doi:10.2478/popets-2020-0054

[9] Ludovic Barman, Italo Dacosta, Mahdi Zamani, Ennan Zhai, Apostolos Pyrgelis, Bryan Ford, Joan Feigenbaum, and Jean-Pierre Hubaux. 2020. PriFi: Low-Latency Anonymity for Organizational Networks. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (Aug. 2020), 24–47. doi:10.2478/popets-2020-0061

[10] Rouzbeh Behnia, Arman Riasi, Reza Ebrahimi, Sherman S. M. Chow, Balaji Padmanabhan, and Thang Hoang. 2024. Efficient Secure Aggregation for Privacy-Preserving Federated Machine Learning . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 778–793. doi:10.1109/ACSAC63791.2024.00069

[11] Fabrizio Boninsegna and Francesco Silvestri. 2025. Differentially Private Release of Hierarchical Origin/Destination Data with a TopDown Approach. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (July 2025), 29–43. doi:10.56553/popets-2025-0087

[12] Tariq Bontekoe, Hassan Jameel Asghar, and Fatih Turkmen. 2025. Efficient Verifiable Differential Privacy with Input Authenticity in the Local and Shuffle Model. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 543–563. doi:10.56553/popets-2025-0076

[13] Andreas Brüggemann, Nishat Koti, Varsha Bhat Kukkala, and Thomas Schneider. 2025. MultiCent: Secure and Scalable Computation of Centrality Measures on Multilayer Graphs. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 944–968. doi:10.56553/popets-2025-0166

[14] Leon Böttger, Alexander Matern, Dennis Arndt, and Matthias Hollick. 2025. Okay Google, Where's My Tracker? Security, Privacy, and Performance Evaluation of Google's Find My Device Network. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 605–619. doi:10.56553/popets-2025-0147

[15] Pithayuth Charnsethikul, Almajd Zunquti, Gale Lucas, and Jelena Mirkovic. 2025. Navigating Social Media Privacy: Awareness, Preferences, and Discoverability. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 620–638. doi:10.56553/popets-2025-0148

[16] Basanta Chaulagain and Kyu Hyung Lee. 2024. FA-SEAL: Forensically Analyzable Symmetric Encryption for Audit Logs . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 716–732. doi:10.1109/ACSAC63791.2024.00065

[17] Christian Collberg, Todd Proebsting, Gina Moraila, Akash Shankaran, Zuoming Shi, and Alex Warren. 2014. *Measuring Reproducibility in Computer Systems Research.* Technical Report TR 14-01. Department of Computer Science, University of Arizona. https://cs.brown.edu/people/sk/Memos/Examining-Reproducibility/measuring-tr.pdf

[18] Daniel Collins, Simone Colombo, and Loïs Huguenin-Dumittan. 2025. Real-World Deniability in Messaging. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 320–340. doi:10.56553/popets-2025-0018

[19] Véronique Cortier, Alexandre Debant, Pierrick Gaudry, and Léo Louistisserand. 2025. Vote&Check: Secure Postal Voting with Reduced Trust Assumptions. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (July 2025), 333–348. doi:10.56553/popets-2025-0101

[20] Anna Crowder, Daniel Olszewski, Patrick Traynor, and Kevin R. B. Butler. 2024. I Can Show You the World (of Censorship): Extracting Insights from Censorship Measurement Data Using Statistical Techniques . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 1123–1138. doi:10.1109/ACSAC63791.2024.00091

[21] Marian Dietz and Florian Kerschbaum. 2025. Private Shared Random Minimum Spanning Forests. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 157–172. doi:10.56553/popets-2025-0055

[22] Daniel J. Dubois, Roman Kolcun, Anna Maria Mandalari, Muhammad Talha Paracha, David Choffnes, and Hamed Haddadi. 2020. When Speakers Are All Ears: Characterizing Misactivations of IoT Smart Speakers. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (Aug. 2020), 255–276. doi:10.2478/popets-2020-0072

[23] Dmitry Duplyakin, Robert Ricci, Aleksander Maricq, Gary Wong, Jonathon Duerig, Eric Eide, Leigh Stoller, Mike Hibler, David Johnson, Kirk Webb, Aditya Akella, Kuangching Wang, Glenn Ricart, Larry Landweber, Chip Elliott, Michael Zink, Emmanuel Cecchet, Snigdhaswin Kar, and Prabodh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the USENIX Annual Technical Conference (ATC)*. 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

[24] Stefan Dziembowski, Shahriar Ebrahimi, and Parisa Hassanizadeh. 2025. VIMz: Private Proofs of Image Manipulation using Folding-based zkSNARKs. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 344–362. doi:10.56553/popets-2025-0065

[25] Yijia Fang, Bingyu Li, Jiale Xiao, Bo Qin, Zhijintong Zhang, and Qianhong Wu. 2024. FreeAuth: Privacy-Preserving Email Ownership Authentication with Verification-Email-Free . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 336–352. doi:10.1109/ACSAC63791.2024.00040

[26] Tommi Gröndahl and N. Asokan. 2020. Effective writing style transfer via combinatorial paraphrasing. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (Aug. 2020), 175–195. doi:10.2478/popets-2020-0068

[27] Christopher Harth-Kitzerow, Ajith Suresh, and Georg Carle. 2025. SoK: Truncation Untangled: Scaling Fixed-Point Arithmetic for Privacy-Preserving Machine Learning to Large Models and Datasets. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (2025), 369–391. doi:10.56553/popets-2025-0135

[28] Christopher Harth-Kitzerow, Yongqin Wang, Rachit Rajat, Georg Carle, and Murali Annavaram. 2025. PIGEON: A High Throughput Framework for Private Inference of Neural Networks using Secure Multiparty Computation. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (July 2025), 88–105. doi:10.56553/popets-2025-0090

[29] Sébastien Henri, Gines Garcia-Aviles, Pablo Serrano, Albert Banchs, and Patrick Thiran. 2020. Protecting against Website Fingerprinting with Multihoming. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 89–110. doi:10.2478/popets-2020-0019

[30] Yasha Iravantchi, Pardis Emami-Naeini, and Alanson Sample. 2025. SoK: (Un)usable Privacy: the Lack of Overlap between Privacy-Aware Sensing and Usable Privacy Research. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 472–490. doi:10.56553/popets-2025-0026

[31] Yufan Jiang, Maryam Zarezadeh, Tianxiang Dai, and Stefan Köpsell. 2025. AlphaFL: Secure Aggregation with Malicious2 Security for Federated Learning against Dishonest Majority. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 348–368. doi:10.56553/popets-2025-0134

[32] Bailey Kacsmar, Chelsea H. Komlo, Florian Kerschbaum, and Ian Goldberg. 2020. Mind the Gap: Ceremonies for Applied Secret Sharing. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 397–415. doi:10.2478/popets-2020-0033

[33] Kate Keahey, Jason Anderson, Zhuo Zhen, Pierre Riteau, Paul Ruth, Dan Stanzione, Mert Cevik, Jacob Colleran, Haryadi S. Gunawi, Cody Hammock, Joe Mambretti, Alexander Barnes, François Halbach, Alex Rocha, and Joe Stubbs. 2020. Lessons Learned from the Chameleon Testbed. In *2020 USENIX Annual Technical Conference (USENIX ATC 20)*. USENIX Association, 219–233. https://www.usenix.org/conference/atc20/presentation/keahey

[34] Maya Kleinstein, Riad Wahby, and Yossi Gilad. 2025. Sybil-Resistant Parallel Mixing. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 639–652. doi:10.56553/popets-2025-0149

[35] Maximilian Kroschewski, Anja Lehmann, and Cavit Özbay. 2025. OPPID: Single Sign-On with Oblivious Pairwise Pseudonyms. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 629–649. doi:10.56553/popets-2025-0080

[36] Jan Lauinger, Jens Ernstberger, Andreas Finkenzeller, and Sebastian Steinhorst. 2025. Janus: Fast Privacy-Preserving Data Provenance For TLS. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 511–530. doi:10.56553/popets-2025-0028

[37] Shiyu Li, Yuan Zhang, Yaqing Song, Fan Wu, Feng Lyu, Kan Yang, and Qiang Tang. 2025. EpiOracle: Privacy-Preserving Cross-Facility Early Warning for Unknown Epidemics. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 361–378. doi:10.56553/popets-2025-0020

[38] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. 2020. The Privacy Policy Landscape After the GDPR. *Proceedings on Privacy Enhancing Technologies* 2020, 1 (Jan. 2020), 47–64. doi:10.2478/popets-2020-0004

[39] Wouter Lueks, Brinda Hampiholi, Greg Alpár, and Carmela Troncoso. 2020. Tandem: Securing Keys by Using a Central Server While Preserving Privacy. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (July 2020), 327–355. doi:10.2478/popets-2020-0055

[40] Sam Martin, Nirajan Koirala, Helena Berens, Thomas Rozgonyi, Micah Brody, and Taeho Jung. 2025. HyDia: FHE-based Facial Matching with Hybrid Approximations and Diagonalization. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 585–604. doi:10.56553/popets-2025-0146

[41] Echo Meißner, Frank Kargl, Benjamin Erb, and Felix Engelmann. 2025. PrePaMS: Privacy-Preserving Participant Management System for Studies with Rewards and Prerequisites. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 632–653. doi:10.56553/popets-2025-0034

[42] Jelena Mirkovic and David Balenson. 2026. Living Artifacts: Enabling Vertical Progress In Cybersecurity Research. In *Proceedings of the 1st Workshop on Metascience and Critical Reflections in Security & Privacy (MetaCRiSP)*. San Jose, CA.

[43] Jelena Mirkovic, Brian Kocoloski, and David Balenson. 2024. Enabling reproducibility through the SPHERE research infrastructure. In *Usenix ;login: Magazine*. https://www.usenix.org/publications/loginonline/enabling-reproducibility-through-sphere-research-infrastructure

[44] James W Moody, Lisa A Keister, and Maria C Ramos. 2022. Reproducibility in the social sciences. *Annual review of sociology* 48 (2022), 65–85. doi:10.1146/annurev-soc-090221-035954

[45] Dimitris Mouris, Christopher Patton, Hannah Davis, Pratik Sarkar, and Nektarios Georgios Tsoutsos. 2025. Mastic: Private Weighted Heavy-Hitters and Attribute-Based Metrics. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 290–319. doi:10.56553/popets-2025-0017

[46] National Academies of Sciences, Engineering, and Medicine. 2019. *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC. doi:10.17226/25303

[47] Hoang-Dung Nguyen, Jorge Guajardo, and Thang Hoang. 2025. Client-Efficient Online-Offline Private Information Retrieval. *Proceedings on Privacy Enhancing Technologies* 2025, 3 (July 2025), 192–212. doi:10.56553/popets-2025-0095

[48] Sean Oesch, Ruba Abu-Salma, Oumar Diallo, Juliane Krämer, James Simmons, Justin Wu, and Scott Ruoti. 2020. Understanding User Perceptions of Security and Privacy for Group Chat: A Survey of Users in the US and UK. In *Proceedings of the 36th Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 234–248. doi:10.1145/3427228.3427275

[49] Daniel Olszewski. 2025. On Defining Reproducible Outcomes for the Computer Security Community. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security* (Taipei, Taiwan) *(CCS '25)*. Association for Computing Machinery, New York, NY, USA, 4872–4874. doi:10.1145/3719027.3765573

[50] Daniel Olszewski, Allison Lu, Anna Crowder, Nathaniel Bennett, Seth Layton, Sri Hrushikesh Varma Bhupathiraju, Tyler Tucker, Siddhant Kalgutkar, Hunter Ver Helst, Carson Stillman, Kevin R. B. Butler, Sara Rampazzi, and Patrick Traynor. 2025. Reproducibility in Applied Security Conferences: An 11-Year Review on Artifacts and Evaluation Committees. In *Proceedings of the 3rd ACM Conference on Reproducibility and Replicability (ACM REP '25)*. Association for Computing Machinery, New York, NY, USA, 96–107. doi:10.1145/3736731.3746151

[51] Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. 2023. "Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (Copenhagen, Denmark) *(CCS '23)*. Association for Computing Machinery, New York, NY, USA, 3433–3459. doi:10.1145/3576915.3623130

[52] Daniel Olszewski, Tyler Tucker, Kevin R. B. Butler, and Patrick Traynor. 2025. SoK: Towards a Unified Approach to Applied Replicability for Computer Security. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security 25)*. USENIX Association, Seattle, WA, USA, 1235–1252. https://www.usenix.org/system/files/usenixsecurity25-olszewski.pdf

[53] Giulio Pagnotta, Dorjan Hitaj, Briland Hitaj, Fernando Perez-Cruz, and Luigi V. Mancini. 2024. TATTOOED: A Robust Deep Neural Network Watermarking Scheme based on Spread-Spectrum Channel Coding . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 1245–1258. doi:10.1109/ACSAC63791.2024.00099

[54] René Raab, Pascal Berrang, Paul Gerhart, and Dominique Schröder. 2025. SoK: Descriptive Statistics Under Local Differential Privacy. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 118–149. doi:10.56553/popets-2025-0008

[55] René Raab, Arijana Bohr, Kai Klede, Benjamin Gmeiner, and Bjoern M. Eskofier. 2025. Estimating Group Means Under Local Differential Privacy. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 236–274. doi:10.56553/popets-2025-0129

[56] Gaganjeet Singh Reen and Christian Rossow. 2020. DPIFuzz: A Differential Fuzzing Framework to Detect DPI Elusion Strategies for QUIC. In *Proceedings of the 36th Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 332–344. doi:10.1145/3427228.3427662

[57] Arman Riasi, Jorge Guajardo, and Thang Hoang. 2024. Privacy-Preserving Verifiable Neural Network Inference Service . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 683–698. doi:10.1109/ACSAC63791.2024.00063

[58] Andre Rosti, Stijn Volckaert, Michael Franz, and Alexios Voulimeneas. 2024. I'll Be There for You! Perpetual Availability in the A8 MVX System . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 520–533. doi:10.1109/ACSAC63791.2024.00052

[59] Daniel Schadt, Christoph Coijanovic, and Thorsten Strufe. 2025. Aimless Onions: Mixing without Topology Information. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 293–307. doi:10.56553/popets-2025-0131

[60] Jochen Schafer, Frederik Armknecht, and Youzhe Heng. 2024. R+R: Revisiting Graph Matching Attacks on Privacy-Preserving Record Linkage . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 699–715. doi:10.1109/ACSAC63791.2024.00064

[61] Phillipp Schoppmann, Lennart Vogelsang, Adrià Gascón, and Borja Balle. 2020. Secure and Scalable Document Similarity on Distributed Databases: Differential Privacy to the Rescue. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 209–229. doi:10.2478/popets-2020-0024

[62] Silvia Sebastián and Juan Caballero. 2020. AVclass2: Massive Malware Tag Extraction from AV Labels. In *Proceedings of the 36th Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 42–53. doi:10.1145/3427228.3427261

[63] Hussein Sheaib, Anja Feldmann, and Ha Dao. 2025. Unmasking the Shadows: A Cross-Country Study of Online Tracking in Illegal Movie Streaming Services. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 125–139. doi:10.56553/popets-2025-0053

[64] Haonan Shi, Tu Ouyang, and An Wang. 2025. Unveiling Client Privacy Leakage from Public Dataset Usage in Federated Distillation. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 201–215. doi:10.56553/popets-2025-0127

[65] Anastasia Shuba and Athina Markopoulou. 2020. NoMoATS: Towards Automatic Detection of Mobile Tracking. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 45–66. doi:10.2478/popets-2020-0017

[66] Paul Syverson, Rasmus Dahlberg, Tobias Pulls, and Rob Jansen. 2025. Onion-Location Measurements and Fingerprinting. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 512–526. doi:10.56553/popets-2025-0074

[67] Shengye Wan, Mingshen Sun, Kun Sun, Ning Zhang, and Xu He. 2020. RusTEE: Developing Memory-Safe ARM TrustZone Applications. In *Proceedings of the 36th Annual Computer Security Applications Conference* (Austin, USA) *(ACSAC '20)*. Association for Computing Machinery, New York, NY, USA, 442–453. doi:10.1145/3427228.3427262

[68] Daniel Weber, Leonard Niemann, Lukas Gerlach, Jan Reineke, and Michael Schwarz. 2024. No Leakage Without State Change: Repurposing Configurable CPU Exceptions to Prevent Microarchitectural Attacks . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 366–379. doi:10.1109/ACSAC63791.2024.00042

[69] Maximilian Weisenseel, Christoph Döpmann, and Florian Tschorsch. 2025. The Last Hop Attack: The Last Hop Attack: Why Loop Cover Traffic over Fixed Cascades Threatens Anonymity. *Proceedings on Privacy Enhancing Technologies* 2025, 2 (April 2025), 382–397. doi:10.56553/popets-2025-0067

[70] Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2020. Differentially Private SQL with Bounded User Contribution. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (April 2020), 230–250. doi:10.2478/popets-2020-0025

[71] Anurag Swarnim Yadav and Joseph N. Wilson. 2024. R+R: Security Vulnerability Dataset Quality Is Critical . In *2024 Annual Computer Security Applications Conference (ACSAC)*. IEEE Computer Society, Los Alamitos, CA, USA, 1047–1061. doi:10.1109/ACSAC63791.2024.00086

[72] Xin Yang and Omid Ardakanian. 2025. PrivDiffuser: Privacy-Guided Diffusion Model for Data Obfuscation in Sensor Networks. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 40–55. doi:10.56553/popets-2025-0118

[73] Ye Zheng, Sumita Mishra, and Yidan Hu. 2025. Optimal Piecewise-based Mechanism for Collecting Bounded Numerical Data under Local Differential Privacy. *Proceedings on Privacy Enhancing Technologies* 2025, 4 (Oct. 2025), 146–165. doi:10.56553/popets-2025-0124

[74] Cong Zuo, Shangqi Lai, Shi-Feng Sun, Xingliang Yuan, Joseph K. Liu, Jun Shao, Huaxiong Wang, Liehuang Zhu, and Shujie Cui. 2025. Searchable Encryption for Conjunctive Queries with Extended Forward and Backward Privacy. *Proceedings on Privacy Enhancing Technologies* 2025, 1 (Jan. 2025), 440–455. doi:10.56553/popets-2025-0024

9 Appendix

9.1 AI Prompt

We first harvested all ARTIFACT-EVALUATION.md and all README.md files from artifacts of interest. In some cases, there were inconsistencies, such as both of these files missing but there would be README.txt or README-evaluation.md. We manually resolved these inconsistencies and retrieved all relevant files. We then prompted Gemini by supplying the actual text of each file (first ARTIFACT-EVALUATION, and if it was missing then README) using the prompt below.

```
You are an expert requirements parser. You will be given a file path
      to a documentation file (README, ARTIFACT-EVALUATION, etc.).
Your goal is to extract information that corresponds to specific
      columns in our artifact review CSV.

**CRITICAL INSTRUCTION FOR LARGE FILES**: If the provided file is
      very large, do not attempt to read the entire file line-by-line
      . Instead, use the `start_line` and `end_line` parameters of
      the `read_file` tool to quickly sample the first 200 lines (e.g
      ., sections labeled 'Requirements', 'Getting Started', '
      Hardware', 'Software') and the last 200 lines (where claims or
      test data are usually mentioned). Your priority is to extract
      the fields below efficiently and terminate.

**STRICT ADHERENCE MANDATE**: You must base your findings ONLY on
      the provided source file. Do not hallucinate data or guess
      requirements. Do exactly what you are told.

Analyze the file and extract exactly these fields:
1) **CPU**: Specific CPU requirements (e.g. 4 cores).
2) **Mem**: Memory/RAM requirements (e.g. 16GB).
3) **Disk**: Storage requirements.
4) **GPU**: Any GPU requirements.
5) **API**: External APIs required.
6) **OS**: Operating system requirements.
7) **Software**: Software/tooling dependencies.
8) **Special reqs**: **Explicitly identify if special hardware is
      strictly required (e.g., IoT devices, specific non-x86
      architectures) versus standard Linux environments.**
9) **Topology**: Any mention of network structure, multiple nodes/
      VMs, or specific connectivity. **Explicitly identify if
      multiple VMs/nodes are required (distributed network) vs. a
      single machine.**
10) **Runs (time to set up)**: Any mention of how long it takes to
      install or run.
11) **Changes needed**: Any mention of manual steps, patches, or
      configurations the user must perform.
12) **Packaging**: Any packaging methods mentioned (e.g. Docker, .
      ova, VM image).
13) **Percent of results reproduced**: Percentage of results the
      authors claim to reproduce.
14) **Has README**: (yes/no)
15) **Has pointer to paper or citation**: (yes/no)
16) **Has license**: (yes/no)
17) **Has test data**: (yes/no)
18) **Has claims folder**: (yes/no) - Is there a folder or section
      dedicated to results/claims?
19) **Has claims scripts within the folder**: (yes/no) - Are there
      scripts to reproduce the claims?
20) **Notes**: Any other critical observations. **Scan for and
      mention if `Dockerfile`, `docker-compose.yml`, or `*.ipynb`
      files are mentioned to determine "runnability."**

Display the findings EXACTLY in this format on a single line, using
      the exact keys below:

CPU: <val> | Mem: <val> | Disk: <val> | GPU: <val> | API: <val> | OS:
       <val> | Software: <val> | Special reqs: <val> | Topology: <val
      > | Runs (time to set up): <val> | Changes needed: <val> |
      Packaging: <val> | Percent of results reproduced: <val> | Has
      README: <val> | Has pointer to paper or citation: <val> | Has
      license: <val> | Has test data: <val> | Has claims folder: <val
      > | Has claims scripts within the folder: <val> | Notes: <val>

Rules:
- If a requirement (like CPU, Mem, Disk, GPU, API, OS, Software) is
      not specified in the documentation, leave the `<val>`
      completely empty (e.g., `CPU:  | Mem:  | Disk: `), you may also
       use "not-found" if missing.
- For other fields (Changes needed, Topology, etc.), you may use "
      not-found" if missing.
- **CRITICAL**: Do NOT use the pipe character (`|`) anywhere inside
      your extracted `<val>` responses, as this will break downstream
       parsing. Use commas instead if you must separate items.
- Do not provide any other text, explanation, or markdown formatting.
       Just the one line.
```

We manually checked all outputs of Gemini AI. While it may indicate "not-found" where we could find some data manually, or mistakenly consider suggested configuration as required, it did not seem to hallucinate requirements where none were present.

9.2 Artifact Badges

Table 2 shows definitions for artifact badges for our selected four conferences. The third column shows if artifacts that were awarded a given badge were included in our evaluation. For those included, we say that they received an equivalent of the Results-reproduced badge.

Table 2: Badge descriptions for our selected conferences and years.

Conference	Badges and Descriptions	Included
ACSAC 2020	**Functional:** The artifacts associated with the research are found to be documented, consistent, complete, exercisable, and include appropriate evidence of verification and validation.	no
	Reusable: The artifacts associated with the paper are of a quality that significantly exceeds minimal functionality. That is, they have all the qualities of the Artifacts Evaluated – Functional level, but, in addition, they are very carefully documented and well-structured to the extent that reuse and repurposing is facilitated. In particular, norms and standards of the research community for artifacts of this type are strictly adhered to.	yes
ACSAC 2024	**Code Available.** This badge signals that author-created digital objects used in the research (including data and code) are permanently archived in a public repository that assigns a global identifier and guarantees persistence, and are made available via standard open licenses that maximize artifact availability.	no
	Code Reviewed. This badge signals that all relevant author-created digital objects used in the research (including data and code) were reviewed according to the criteria provided by the badge issuer.	no
	Code Reproducible. This badge signals that an additional step was taken or facilitated by the badge issuer (e.g., publisher, trusted third-party certifier) to certify that an independent party has regenerated computational results using the author-created research objects, methods, code, and conditions of analysis.	yes
PETS 2020	**Artifact.** One badge including very basic checks for the proper documentation, licensing, and compilation of artifacts, rather than doing an in-depth analysis of the source code and the reproducibility of results in the paper.	selected artifacts
PETS 2025	**Artifact Available.** This badge indicates that the artifact is publicly available at a permanent location (not be behind any kind of paywall or restricted access).	no
	Artifact Functional. For this badge the artifact should satisfy these criteria: (1) Documentation: It clearly documents how it relates to the paper, and how it should be used. (2) Completeness: It includes all the core contributions described in the paper. (3) Exercisability: It includes the scripts and data needed to run the experiments described in the paper. The software must be successfully executable.	no
	Artifact Reproduced. This badge requires the core contributions of the paper to be reproduced by the reviewers.	yes