Toward Findable, Accessible, Interoperable, and Reusable Cybersecurity Artifacts

David Balenson
david.balenson@sri.com
SRI International
Arlington, VA, USA

Terry Benzel
tbenzel@isi.edu
USC/ISI
Marina del Rey, CA, USA

Eric Eide
eeide@cs.utah.edu
University of Utah
Salt Lake City, UT, USA

David Emmerich
davidpe@illinois.edu
University of Illinois
Urbana, IL, USA

David Johnson
johnsond@cs.utah.edu
University of Utah
Salt Lake City, UT, USA

Jelena Mirkovic
mirkovic@isi.edu
USC/ISI
Marina del Rey, CA, USA

Laura Tinnel
laura.tinnel@sri.com
SRI International
Arlington, VA, USA

ABSTRACT

Researchers in experimental cybersecurity are increasingly sharing the code, data, and other artifacts associated with their studies. This trend is encouraged and rewarded by conferences and journals through practices such as artifact evaluation and badging. While these trends in sharing artifacts are promising, the cybersecurity community is still far from an ecosystem in which artifacts are FAIR: findable, accessible, interoperable, and reusable. The lack of established standards and best practices for sharing and reuse results in artifacts that are often difficult to find and reuse; in addition, the lack of community standards results in artifacts that may be incomplete and low-quality. In this paper we describe our experience in creating an online community hub, called SEARCCH, to promote the sharing and reuse of artifacts for cybersecurity research. Based on our experience, we offer lessons learned: issues that must be addressed to further promote FAIR principles in experimental cybersecurity.

CCS CONCEPTS

• **Information systems** → **Digital libraries and archives**; *Collaborative and social computing systems and tools*; • **Security and privacy**;

KEYWORDS

artifact catalog, cybersecurity artifacts, FAIR principles, reproducibility, SEARCCH

CSET 2022, August 8, 2022, Virtual, CA, USA

ACM ISBN 978-1-4503-9684-4/22/08…$15.00
https://doi.org/10.1145/3546096.3546104

ACM Reference Format:
David Balenson, Terry Benzel, Eric Eide, David Emmerich, David Johnson, Jelena Mirkovic, and Laura Tinnel. 2022. Toward Findable, Accessible, Interoperable, and Reusable Cybersecurity Artifacts. In *Cyber Security Experimentation and Test Workshop (CSET 2022), August 8, 2022, Virtual, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3546096.3546104

1 INTRODUCTION

It has become common for conference and journals to encourage researchers to share the software, datasets, methodologies, and other artifacts associated with a publication, so that the readers of the publication can (1) better understand the contributions of the paper and also (2) attempt to reproduce the results of the experiments presented, and thereby gain confidence in those results and/or insight into the research itself. While individual experiments are necessarily specific to the issues being evaluated, the artifacts associated with a study may be generalizable and thus useful to other researchers tackling similar problems. The authors of artifacts are often rewarded with badges that are affixed to their papers, indicating that the associated artifacts are publicly available, have been used to reproduce the paper's results, or have otherwise been judged to be high quality [1, 2, 21].

Because the artifacts associated with experimental cybersecurity research are mainly software and data, it is conceptually straightforward for those artifacts to support the *FAIR* principles: i.e., be *findable, accessible, interoperable,* and *reusable* [16, 22]. From a technical standpoint, it is simple to make software publicly available at little or no cost through websites such as GitHub and Software Heritage [8], and it is similarly easy to make datasets publicly available through sites such as Zenodo [11], Figshare [14], and Dryad [9]. Compared to other areas of science, performing experiments in computer science often requires little or no specialized equipment. Moreover, thanks to national investments in public testbeds [4, 10, 17] and clouds [19], any researcher can easily gain access to resources needed for experimentation.

In practice, however, the current state of artifacts for cybersecurity research is not so FAIR. The lack of clear coordination across

the research community and the lack of established standards and best practices for sharing and reuse result in artifacts that are often difficult to find and reuse, and which may be incomplete and low-quality. Consider recent surveys of the state of reproducibility of computational research in general. In 2016, Collberg and Proebsting [7] systematically evaluated the repeatability of 601 papers from ACM conferences and journals: they found that only a third of papers had artifacts that are reusable with modest effort, and 54% had artifacts that could be reused with higher level of effort, potentially involving correspondence with authors. In 2018, Flittner et al. [15] surveyed the authors of papers from CoNEXT, ICN, IMC, and SIGCOMM: they found diversity in artifact sharing and reuse across different research communities, and also found missing and dead links and incomplete artifacts.

Toward advancing FAIR principles for cybersecurity research, we created a web-based community portal, called SEARCCH,[1] that aims to improve the findability and reusability of cybersecurity artifacts. SEARCCH is both a catalog and community. As a catalog, it maintains a database of information about research artifacts that are located in different places on the Internet (e.g., GitHub, Zenodo, or other digital library). SEARCCH helps researchers find artifacts by enabling searching over domain-specific keywords and other metadata. It also stores *relationships* between artifacts, making it easier to find multiple artifacts associated with a particular effort or topic. As a community, SEARCCH allows researchers to extend the hub's content with new artifacts and discussion. SEARCCH lowers the barrier to publishing artifacts in its catalog through automated submission-assistant tools that extract and process metadata about artifacts. Online comments allow users to share their experiences with artifacts that they have used (or *tried* to use!).

Our aim in this short experience paper is to describe our motivations for creating SEARCCH, summarize our activities in developing the hub, share some of the lessons we have learned, and reflect on SEARCCH's contributions to advancing FAIR principles for cybersecurity research. One of the lessons is the importance of sustained community outreach and engagement. The development of the hub was driven by feedback from the community on early prototypes; this engagement was essential for defining the hub's features. Ongoing outreach will be required for SEARCCH to attain "critical mass" in terms of users and cataloged artifacts. A second lesson relates to the limits of current practice. We chose to work with artifacts as they exist, but this curtailed the hub's ability to automatically obtain useful metadata about artifacts. To further advance FAIR principles, we conclude that new metadata standards for artifacts need to be developed and adopted by the research community.

2 FAIR CHALLENGES

The FAIR principles [16, 22] aim to enhance the value of published research through digital artifacts that are *findable, accessible, interoperable,* and *reusable.* Findability relates to metadata, both for the artifact itself (e.g., a DOI) and for describing the content of the artifact. Accessibility means that the artifact is openly available to interested parties. Interoperability concerns the representation of the artifact, e.g., the use of standard languages and vocabularies for datasets, and the use of standard tools, libraries, and techniques for software. Reusability relates to accurate provenance, clear licensing terms, and adherence to other community standards.

There are multiple challenges to achieving FAIR principles for experimental cybersecurity, and we summarize some of these challenges below. Our discussion stems from our long work in promoting activities toward reproducibility and artifact reusability, on improving testbed-based experimentation, and from the numerous community events we organized under our SEARCCH effort.

Findability challenge: Many artifacts, but little metadata. At a basic level, computational artifacts—code and data—are easy to produce and share. Today there are many repositories of computational artifacts, shared by researchers and research labs. However, because there is no standardized format for artifact metadata, and because artifacts are shared through many different channels, it is difficult for researchers to actually find artifacts that are relevant to them. Furthermore, once an artifact is found, it is difficult for a researcher to accurately evaluate how suitable the artifact is to the researcher's goal, and what effort will be needed to reuse it.

Accessibility challenge: Identifying and documenting artifacts. Experimentation often occurs over many months or even years, as many problems require elaborate building of infrastructure (i.e., test environments) to achieve realism, scale, or both. Experiments may also include work by many students and staff, and may build on work of prior researchers in the same lab. When computational artifacts are shared, it is necessary to share all relevant details of the infrastructure used to produce them, and the entire workflow of the relevant experiments. But this knowledge may be highly distributed across the infrastructure (e.g., reside in many different files, file versions, and storage locations) and across current and past researchers.

It is often very difficult to identify all the relevant pieces of knowledge that are necessary for artifact reuse. It is also difficult to foresee and mitigate challenges that researchers may face when they attempt to reuse an artifact on a different infrastructure than the one that was used to produce it. The research community needs a common format for capturing the knowledge that is necessary to adopt artifacts. Such documentation would help researchers evaluate in advance the time and effort needed for reuse.

Interoperability and Reusability challenge: Infrastructure dependence. Code artifacts are produced in some programming language, assuming some environment dependencies. Data artifacts are also produced on some specific hardware and software infrastructure. To reuse an artifact, a researcher must often become familiar with the infrastructure used to produce it. An artifact may depend heavily on the infrastructure, e.g., computer code may only run on a *specific* version of an operating system, or it may require a *particular* version of a software library, or it may assume a certain switch queuing strategy or given GPU technology.

Artifacts may further not be interoperable. A simple example is data shared by one researcher in one format (e.g., network flow traces) while a tool shared by another researcher consumes an incompatible format (e.g., network packet traces). Code artifacts may require different execution environments: e.g., one requires Ubuntu and another requires CentOS. An added challenge to reusability is the availability of the software's environment or data. Some artifacts may be produced using private hardware or software, or

[1] Sharing Expertise and Artifacts for Reuse through a Cybersecurity Community Hub. https://searcch.cyberexperimentation.org, https://hub.cyberexperimentation.org

using private data. In these cases, researchers might only be able to partially reproduce some findings, or they might be unable to reuse an artifact without its private dependencies.

Motivational challenge: High effort for sharing and reuse. We must also consider the incentives for aiming to achieve FAIR principles in practice. Researchers who share artifacts cannot foresee the demand for their artifact or the expertise of researchers that may want to reuse it. Packaging artifacts is hard, and if there is low demand for a specific artifact, the producer has wasted effort. For this reason, most shared artifacts have minimal instructions for reuse, are incomplete, and may have undocumented dependencies. While many conferences offer badges for artifact sharing, the concrete benefits of receiving a badge may be minimal.

An artifact consumer may spend weeks or even months recreating the necessary environment for a code artifact. This work goes unrewarded: it is simply the cost of doing research. It is often easier for researchers to evaluate their own systems in a setting of their choice than it is for those researchers to reproduce the environments in which published artifacts were evaluated, for fair comparison. Even when an artifact consumer spends time making an artifact work reliably in a new context, there may be no clear way for that consumer to communicate improvements back to the community and receive recognition.

A FAIR example. To conclude our discussion of FAIR principles, we share the true experience of a graduate student[2] who worked to reuse a cybersecurity artifact: Genius [12], a binary-similarity tool.

The Genius publication [13] appeared at ACM CCS 2016. The paper did not contain a link to an artifact and there was no artifact released through the venue. The student searched on Google and located two artifacts: the original [12], released by the author, and an improved artifact [18], released by another researcher. The original artifact contained a minimal README file, without instructions for installation and execution, and with partial code (providing two out of four functionalities of Genius). The improved artifact had a longer README file, with instructions for installation and execution, and it provided the complete Genius functionality.

The student continued with the improved artifact. The README specified that code needed IDA Pro v7.0, a commercial disassembler, which requires a license. The student asked Hex-rays (the owner of IDA Pro) for an educational-use license and obtained it. The student then discovered that the latest version of IDA Pro (v7.7) was not compatible with the artifact, which used deprecated APIs. The student replaced these with the newer APIs. The student also read the Genius code and discovered hard-coded parameters, which they replaced with parameters better suited to their task.

Datasets from the Genius paper were not released, but the authors specified the names and version numbers of the public binaries they used, which allowed the student to locate them. The authors of Genius experimented with a random selection of functions from the binaries, but they did not provide details which functions were selected, and thus their results could not be fully reproduced.

After resolving the dependencies, the student moved the code from their laptop to a lab server, which was accessible remotely.

[2]Advised by one of the authors.

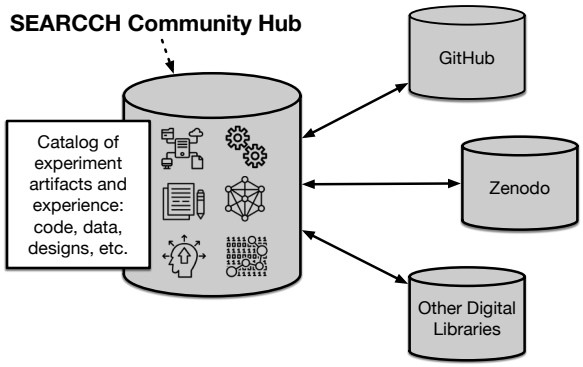

Figure 1: SEARCCH catalogs metadata about cybersecurity artifacts that are stored in separate repositories.

They spent another week learning how to make IDA Pro run in headless mode (i.e., without a GUI). In total, it took almost four weeks to reuse this artifact.

3 SEARCCH

We created SEARCCH, a new web-based portal, with the goal of improving the findability and reusability of experiment artifacts within the cybersecurity research community. One can think of SEARCCH as a community center where artifacts are advertised, where researchers search for and locate artifacts that are relevant to their interests, and where the authors and consumers of artifacts can meet to discuss those artifacts and share experiences. SEARCCH is *not* a new repository for artifacts: rather, it is a *catalog* of artifacts that are stored across the Internet, e.g., in GitHub, Zenodo, arXiv, institutional repositories, or elsewhere.

The concept of the SEARCCH community hub is illustrated in Figure 1. SEARCCH is based on a catalog of experiment artifacts. The catalog contains *metadata* describing the artifacts, which are stored in places such as GitHub. The metadata in the catalog is intended to promote the *findability* of artifacts by people who use the SEARCCH hub. This means that in addition to basic information about each artifact—e.g., name, authors, kind (software, dataset, paper), web location, and license—the catalog also contains keywords, snippets of free-form documentation, venue information, and the set of awarded badges for each artifact.

The ways that researchers interact with SEARCCH are illustrated in Figure 2. By using text-based or structured search, a visitor can quickly locate artifacts—potentially, multiple artifacts—that match their interests. SEARCCH consults its metadata store to locate the artifact records that match the user's search criteria. When a user views the information about an artifact, the browser also displays links to related catalog items, e.g., the paper or papers in which a software artifact was used. SEARCCH keeps information about the relationships among artifacts in a knowledge graph. From the SEARCCH display of an artifact record, a user can follow a link to the actual artifact, e.g., stored in GitHub. A user can also rate an artifact (1–5 stars), mark it as a personal favorite ("like") for easy future access, or enter and publish a comment on the artifact. Ratings and comments are stored in the SEARCCH database.

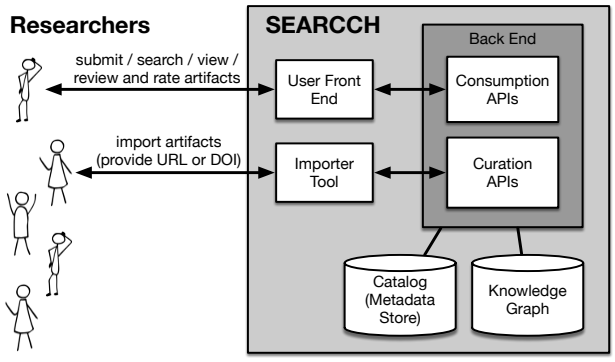

Figure 2: Researchers interact with SEARCCH through consumption and curation APIs.

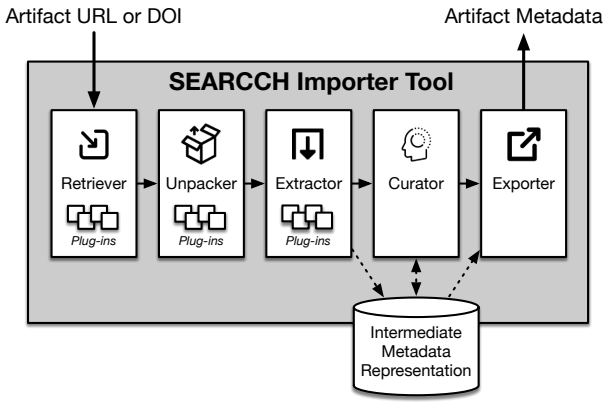

Figure 3: The SEARCCH importer tool uses modules to obtain metadata about artifacts stored in a variety of locations and formats.

A researcher can add to SEARCCH by *importing* an artifact into the SEARCCH catalog. Requiring a user to manually enter lots of metadata about an artifact would be a high barrier to contributions, as well as error-prone. To address this barrier, SEARCCH incorporates an *importer tool* that automatically obtains metadata about an artifact: the user provides the URL or DOI of the artifact, and the importer examines the artifact to extract its metadata. The importer is designed in a modular fashion, illustrated in Figure 3, to support a variety of artifact locations and formats. One set of modules obtains artifact files and metadata from their original locations: e.g., version control systems, websites, or digital libraries. A second set of modules performs "unpacking" of various data formats, and a third set of "extractor" modules examines the unpacked components of an artifact to obtain additional metadata and discover additional artifacts that a user may also want to import. The current SEARCCH importer can import artifact metadata from GitHub repositories, Zenodo, the ACM Digital Library, IEEE Xplore, USENIX conference paper webpages, ACSAC and NDSS conference paper webpages, ACSAC conference artifact webpages, arXiv, and Papers With Code. Some of these sites provide APIs for obtaining metadata, while others require "screen scraping" the content of HTML pages.

While the automated importer tool can greatly reduce the effort required to add an artifact to the SEARCCH catalog, it is ultimately heuristic and imprecise, and the metadata that it obtains is often incomplete. For this reason, SEARCCH allows a user to inspect and edit the information about an artifact before the information is published in the catalog. We refer to this step as *curation* (Figure 3). SEARCCH stores a user's changes to the metadata separately from the artifact record itself. This allows changes to be "replayed" if the artifact is later re-imported, say, because a new version of the artifact was made available.

SEARCCH is a *community* hub and relies on its users to build and maintain the quality of its catalog. The importer tool aims to make it easy for users to create new catalog entries and update existing ones, but users can also create and update records "manually," and the hub does not set any minimum quality standard for catalog entries or the artifacts they reference. This is a deliberate choice to encourage contributions. At the same time, through community engagement activities, we encourage the authors of cataloged artifacts to "take ownership" of their artifact records in SEARCCH. In the future, this will allow authors to better manage their catalog entries. Artifact records in SEARCCH are versioned, so they can be updated (by creating new versions) as the artifacts themselves evolve over time. A record can also be "deleted" from SEARCCH—hidden from public view—if the underlying artifact becomes unavailable, i.e., a "dead link." As the SEARCCH hub grows, we expect that its content-management features will need to evolve to meet the expectations of the cybersecurity research community.

We opened SEARCCH to beta testers in August 2021 and to the public in December 2021. The catalog currently contains information about 393 cybersecurity artifacts and publications: 206 software artifacts, 55 datasets, and 132 publications.

4 LESSONS LEARNED

We continue to enhance SEARCCH's features, expand the number of artifacts in its catalog, and increase its user base. Although the hub is still quite new, we have drawn lessons in two main areas.

The importance of community outreach and engagement. Our outreach and engagement goal was to develop a diverse and vibrant community of cybersecurity researchers who use SEARCCH to actively share their work and reuse the work of others. One of the most effective means for building an active community is to instill a sense of ownership; our outreach and engagement activities were designed to do just that. Outreach activities work to promote and raise awareness of SEARCCH (e.g., via posters, presentations, and social media), and engagement activities work to elicit input and feedback into the design of SEARCCH as well to encourage contributions (e.g., via birds-of-a-feather sessions and artifact "parties").

One of our main takeaways from community engagement is how critically important it is to engage researchers earlier in the development process. We tried anticipating researcher desires in the earlier phases and found that we either underestimated the importance of, or completely missed, desired features. We also needed to understand the relative importance of different features so we could prioritize our development. For example, we learned that not all researchers want to work with SEARCCH in the same way: the

portal must support a diverse set of workflows. We also learned that researchers strongly want to minimize the time required to include their artifacts in SEARCCH. This led to building simpler user interfaces and providing easy-to-use tools that help automate the process. Finally, we underestimated the community's desire to be able to edit an artifact's metadata. Had we engaged in more discussion around this topic, this feature would have been implemented sooner. We did a better job of estimating the kind of simple search mechanism needed. Demonstrating this feature to the community helped us evolve the feature to store the right sets of metadata and relationships among artifacts.

Another important lesson is that word choice really matters in communicating ideas. We found that our initial language for describing SEARCCH incorrectly gave the impression that the portal housed artifacts. We have since revised how we describe SEARCCH.

The need for improved community metadata standards. Achieving the FAIR principles for cybersecurity artifacts critically depends on the metadata used to describe those artifacts. To achieve a high degree of findability, we must go beyond "typical" publication attributes such as title, authorship, and venue: we need structured attributes that describe an artifact's domain of concern (e.g., "DDoS mitigation" or "malware analysis") and the contexts in which the artifact is applicable (e.g., particular software platforms). To support findability, we need metadata to describe relationships between artifacts, e.g., the papers in which a software or dataset artifact is presented or reused. Maximizing reusability depends not only on packaging (e.g., build scripts and encapsulated execution environments) but also on metadata that documents the requirements of an artifact (e.g., its software dependencies) and how it executes.

The SEARCCH importer tool obtains metadata about the artifacts that are imported into its catalog. In designing the hub, we made a deliberate decision to work with artifacts "as they exist." While this approach significantly reduced the barrier to importing artifacts to the hub, our experience has highlighted the limits of this approach: it is often difficult or impossible for the importer tool to automatically obtain complete metadata that would promote the artifact's findability and reusability via the hub. Based on our experience, we conclude that further advancing FAIR principles for cybersecurity artifacts will require the design and adoption of new community metadata standards and practices for artifacts.

5 RELATED WORK

SEARCCH was motivated by the conclusions of the NSF-funded Cybersecurity Experimentation of the Future (CEF) community-based study of expected needs for experimentation infrastructure [3] and subsequent community-engagement workshops. The participants in these workshops indicated strong interest in community infrastructure to facilitate the sharing and reuse of experimental designs, methodologies, tools, and artifacts.

Papers with Code [20] is a platform that couples code and data artifacts with published papers about machine learning. While SEARCCH may include ML-relevant artifacts, it would do so only in the context of cybersecurity and networking applications. Zenodo [11] packages and shares papers with code artifacts for a broad set of domains; thus, unlike SEARCCH, only a small fraction of Zenodo's content is relevant to cybersecurity. Unlike Papers with Code and Zenodo, SEARCCH does not provide any storage for artifacts themselves. Instead, it relies on general-purpose, open-access repositories such as GitHub, Zenodo, and HAL-Inria [5] to store artifacts, and the SEARCCH metadata catalog points to those other sites. In this regard, SEARCCH is somewhat like Google and other search engines.

SEARCCH is also similar to FindResearch.org [6], a website that catalogs artifacts related to computer science publications. Both FindResearch.org and SEARCCH provide links to artifacts that are stored elsewhere. The SEARCCH hub is distinguished, however, in its focus on cybersecurity, the greater amount of metadata it collects about artifacts (toward achieving the FAIR principles), and its focus on community-building features, e.g., direct contributions and comments from users.

SEARCCH supports and complements the growing trend toward artifact evaluation and sharing associated with cybersecurity conference publications. The Annual Computer Security Applications Conference (ACSAC) [2] and the annual USENIX Security Symposium [21] both feature artifact-evaluation processes that are based on the ACM's Artifact Review and Badging guidelines [1].

6 CONCLUSION

SEARCCH aims to improve the findability and reusability of cybersecurity artifacts. Community input was essential for defining the features of the hub, and more community involvement will be needed to further advance FAIR principles through the development of new metadata standards for experiment artifacts.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and help in improving this paper. We especially thank our colleagues and the community members who participated in our alpha/beta test programs and in our community outreach and engagement activities, providing valuable input and feedback on the design of SEARCCH. This material is based upon work supported in part by the National Science Foundation under Grant Numbers 1925564, 1925588, 1925616, and 1925773.

REFERENCES

[1] ACM. 2020. Artifact Review and Badging Version 1.1 - August 24, 2020 (website). https://www.acm.org/publications/policies/artifact-review-and-badging-current.
[2] ACSA. 2022. ACSAC Paper Artifacts (website). https://www.acsac.org/2022/submissions/papers/artifacts/.
[3] David Balenson, Laura Tinnel, and Terry Benzel. 2015. Cybersecurity Experimentation of the Future (CEF): Catalyzing a New Generation of Experimental Cybersecurity Research. https://cef.cyberexperimentation.org/.
[4] Terry Benzel. 2011. The Science of Cyber Security Experimentation: The DETER Project. In *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC)*. 137–148. https://doi.org/10.1145/2076732.2076752
[5] Center for Direct Scientific Communication (CCSD). 2022. HAL-Inria (website). https://hal.inria.fr/.
[6] Christian Collberg and Todd Proebsting. 2021. FindResearch.org (website). http://www.findresearch.org/.
[7] Christian Collberg and Todd A. Proebsting. 2016. Repeatability in Computer Systems Research. *Commun. ACM* 59, 3 (Feb. 2016), 62–69. https://doi.org/10.1145/2812803
[8] Roberto Di Cosmo and Stefano Zacchiroli. 2017. Software Heritage: Why and How to Preserve Software Source Code. In *iPRES 2017: 14th International Conference on Digital Preservation* (2017-09-25). 10 pages. https://hal.archives-ouvertes.fr/hal-01590958
[9] Dryad. 2022. Dryad (website). https://datadryad.org/.

[10] 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 Prabudh Mishra. 2019. The Design and Operation of CloudLab. In *Proceedings of the 2019 USENIX Annual Technical Conference (ATC)*. 1–14. https://www.usenix.org/conference/atc19/presentation/duplyakin.

[11] European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. https://doi.org/10.25495/7GXK-RD71

[12] Qian Feng. 2016. Gencoding. https://github.com/qian-feng/Gencoding. The original Genius code artifact.

[13] Qian Feng, Rundong Zhou, Chengcheng Xu, Yao Cheng, Brian Testa, and Heng Yin. 2016. Scalable Graph-based Bug Search for Firmware Images. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 480–491. https://doi.org/10.1145/2976749.2978370

[14] Figshare, LLC. 2022. Figshare (website). https://figshare.com/.

[15] Matthias Flittner, Mohamed Naoufal Mahfoudi, Damien Saucez, Matthias Wählisch, Luigi Iannone, Vaibhav Bajpai, and Alex Afanasyev. 2018. A Survey on Artifacts from CoNEXT, ICN, IMC, and SIGCOMM Conferences in 2017. *SIGCOMM Comput. Commun. Rev.* 48, 1 (April 2018), 75–80. https://doi.org/10.1145/3211852.3211864

[16] Daniel S. Katz, Morane Gruenpeter, and Tom Honeyman. 2021. Taking a fresh look at FAIR for research software. *Patterns* 2, 3 (2021), 100222. https://doi.org/10.1016/j.patter.2021.100222

[17] 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 *Proceedings of the 2020 USENIX Annual Technical Conference (ATC)*. 219–233. https://www.usenix.org/conference/atc20/presentation/keahey.

[18] Yunlong Lyu. 2020. Genius. https://github.com/Yunlongs/Genius. The improved Genius code artifact.

[19] Jeffrey Mervis. 2021. U.S. law sets stage for boost to artificial intelligence research. Article in ScienceInsider blog. https://doi.org/10.1126/science.abg4337

[20] Meta Platforms, Inc. 2022. Papers with Code (website). https://paperswithcode.com/.

[21] USENIX Association. 2022. USENIX Security '22 Call for Artifacts (website). https://www.usenix.org/conference/usenixsecurity22/call-for-artifacts.

[22] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3 (2016), 160018. https://doi.org/10.1038/sdata.2016.18