SoK: Towards a Unified Approach of Applied Replicability for Computer Security

Daniel Olszewski
University of Florida

Tyler Tucker
University of Florida

Kevin R. B. Butler
University of Florida

Patrick Traynor
University of Florida

Abstract

Reproducibility has been an increasingly important focus within the Security Community over the past decade. While showing great promise for increasing the quantity and quality of available artifacts, reproducibility alone only addresses some of the challenges to establishing experimental validity in scientific research and is not enough to move forward our discipline. Instead, replicability is required to test the bounds of a hypothesis and ultimately show consistent evidence to a scientific theory. Although there are clear benefits to replicability, it remains imprecisely defined, and a formal framework to reason about and conduct replicability experiments is lacking. In this work, we systematize over 30 years of research and recommendations on the topics of reproducibility, replicability, and validity, and argue that their definitions have had limited practical application within Computer Security. We address these issues by providing a framework for reasoning about replicability, known as the Tree of Validity (ToV). We evaluate an attack and a defense to demonstrate how the ToV can be applied to threat modeling and experimental environments. Further, we show two papers with Distinguished Artifact Awards and demonstrate that true reproducibility is often unattainable; however, meaningful comparisons are still attainable by replicability. We expand our analysis of two recent SoK papers, themselves replicability studies, and demonstrate how these papers recreate multiple paths through their respective ToVs. In so doing, we are the first to provide a practical framework of replicability with broad applications for, and beyond, the Security research community.

1 Introduction

Prompted by growing concerns of a reproducibility crisis across all of science and engineering, the Security Community has recently enacted Artifact Evaluation Committees (AECs) in many of its most important venues. The promises of AECs appear to be many, from creating an opportunity for current authors to demonstrate that their claims are computationally reproducible to providing future researchers with artifacts to more easily compare their results to prior efforts. While the measured benefits of reproducibility have not yet been fully demonstrated [40], the widespread practice of making artifacts available and reproducible is likely to improve the value of results published by this community.

Even with the potentially substantial increase in the number of artifacts available to the community, significant challenges remain. For instance, while the term *computationally reproducible* is rigidly defined to mean a different team using the same code and data to achieve the same result [38], there are few meaningful definitions that assist in reasoning about the differences between two efforts when reproducibility of a study is not possible, such as when the underlying data or the experimental setup are unavailable or the environmental conditions are unattainable.

In such cases, *replicability* tests the limits of a hypothesis under new conditions, providing a new understanding of the experiments that reproducibility alone cannot achieve. Replicability advances a new understanding of areas and shows a trend across multiple independent studies. Other areas of science (e.g., Medicine and Psychology) primarily focus efforts on replicability. As reproducibility and replicability control for different outcomes, the lack of reproducibility does not mean a study is not replicable and vice versa. While this community's efforts have primarily focused on reproducibility, we introduce formal frameworks and methods to discuss replicability. For the remainder of this paper, we will refer to the broad field of reproducibility/replicability as *Validity* to be precise in our definitions and avoid confusion.[1]

In this paper, we make the following contributions:

- **Systematization of Validity:** Previous work, not only in computer security but all of computational science, does not develop a comprehensive or directly applicable definition of replicability. We systematize 30 years of

[1] Validity has been used to describe reproducibility [15], but it has never to our knowledge been used to describe the field. Borsboom et al. [17] discuss an ontological viewpoint in Psychology to describe Validity as testing the integrity and bounds of the hypothesis.

meta-science on this topic with the goal of providing a practical understanding of replicability. Further, we examine the unique challenges and aspects of computer security replicability.

- **Framework of Validity:** We provide a new approach to classifying reproducibility/replicability using a binary tree of available artifacts, the Tree of Validity (ToV). This framework unifies with previous definitions of validity and applies a structure for comparing validity studies. We show that this framework can adapt to the needs of the Security Community.

- **Application to Case Studies:** We provide two case studies of an attack and a defense paper to demonstrate the utility of our framework to Security Research. Further, we demonstrate using the framework on individual papers that have recognized Distinguished Artifact awards [16, 59], showing how each can be mapped to a ToV and how reproducibility for both is limited in spite of their award status. We then more broadly show the application of our framework to previous SoK papers [34, 52], which show how our ToV framework unifies multiple experimental efforts that move toward improved claims of validity.

The benefit of this work is that it not only conceptually systematizes the topics of replicability and validity, but it also provides a practical framework adapted and guidance allowing researchers to better contextualize their contributions to a research area and ideally facilitate a more rapid advancement of Security Research.

The remainder of this paper is organized as follows: Section 2 provides definitions of terms; Section 3 systematizes this space; Section 4 discusses the problems and challenges that security research faces; Section 5 introduces the ToV framework, its goals, and visualizations of multiple possible ToVs; Section 6 performs case studies on an attack and defense paper as well as award-winning artifact papers; Section 7 offers discussion and open problems; and Section 8 provides concluding remarks.

2 Definitions

We provide definitions of various parts of experimental methodologies to avoid confusion.

We refer to *validity* as the field of science that reproducibility and replicability are a subset of. Our systematization demonstrates the differences between the two definitions but focuses on the complexity of replicability. A validator is a group or team that conducts experiments to gauge the validity of another author's work, through reproducing or replicating the experiments. A validity experiment is any such experiment that is in part based on the original authors' work.

For experiments, we define the *setting* as composed of a problem and domain. We define the *problem* as a scientific question that the experiments provide evidence to (e.g., deepfake detection). The *domain* is the environment of the experiment. It can include but is not limited to the population studied, time, software systems, hardware systems, etc. As an example of a setting, detecting malicious network traffic is the problem with the domain being a specific network.

The *process* is the experimental methodology conducted in a setting. A *method* is the approach to gather and/or manipulate data. The *data* is the collection of measurements or observations within the setting. Finally, the *analysis* is conducted on the data and provides a quantifiable measure. To follow the above example, the data is the network traffic (e.g., TCP connections). The method is how the malicious traffic is detected (e.g., a machine learning detector). The analysis is then a gauge of detecting the performance of the method (e.g., false positives).

3 Systematization

To motivate the development of our framework, we provide a systematization of the numerous proposals for defining reproducibility and replicability. While the discussions on reproducibility have resulted in a clear definition, replicability remains vague and largely unexamined. As such, our systematization shows that there remain open problems with the definitions of replicability, especially in adapting these definitions to Security Research. This systematization is not exhaustive of every available definition of reproducibility. Rather, it seeks to highlight research into reproducibility and replicability that aim to clarify misunderstandings of validity and build frameworks to address challenges within validity. Figure 1 shows the relative relation between all of the definitions systematized in this work.

3.1 Claerbout Terminology

The *Claerbout Terminology* [45] refers to some of the first papers to formally define computational reproducibility. We highlight the two papers associated with building the basis for these definitions: Claerbout et al. [18] and Peng [44].[2]

Claerbout et al. [18] - With the proliferation of computational capabilities, Claerbout et al. [18] identified that anyone should be able to validate computational results. This is due to the innate ability of the original authors to code experiments, requiring that the "validator" only runs the program. As such they structured their classes and research group to promote reproducible research and identified reproducibility as being able to run the same software on the same input and obtaining the same results. Claerbout et al. [18] proposed several goals

[2] Plesser [45] is the first to refer to this taxonomy as the Claerbout Terminology, but Barba [14] and Liberman [37] also group these papers together.

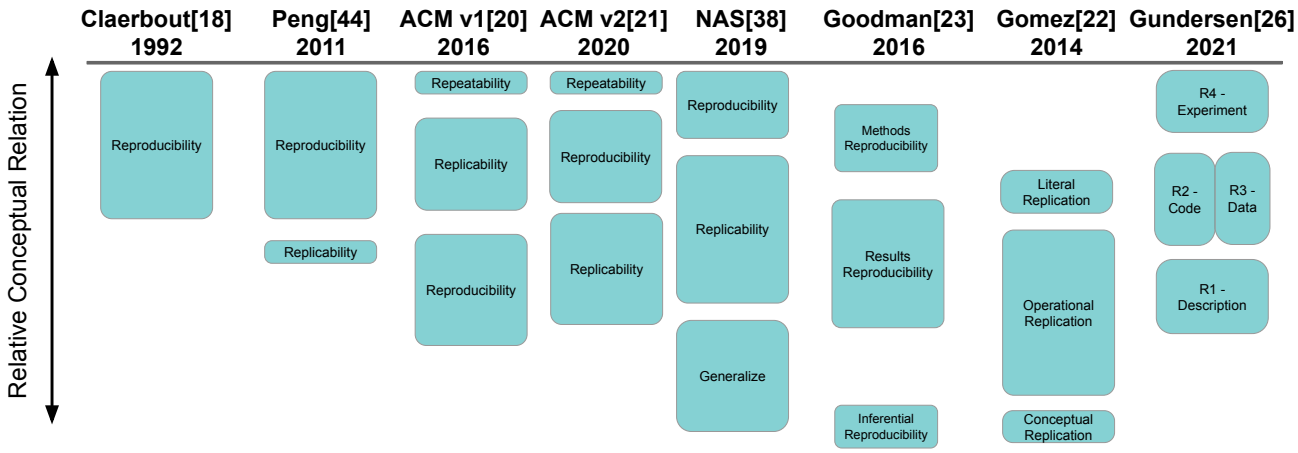

Figure 1: We can only represent a relative alignment of definitions proposed in this systematization, as no grounding framework is present to compare between the definitions. The box around each term is a conceptual spectrum and covers multiple meanings. Similar figures exist in [14, 22, 37, 45].

for reproducible research: first, to "teach researchers how to ... reproduce their own research results a year or more later" (i.e., longevity); second, to "learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button" (i.e., iteration); and third, to "prepare a complete copy of ... a local software environment so that graduating students can take their work away... to other sites ... and reproduce their work." They achieve this goal by outlining a systematic way to prepare research artifacts, where documents are created using software that runs the code for each experiment in the compilation of the document.[3] This is one of the first definitions of reproducible research and self-contained experiments.

Peng [44] - In discussing the challenges faced by computational reproducibility, Peng argues that replicability is the ultimate standard. Replicability is where "independent investigators address a scientific hypothesis and build up evidence for or against it." This constitutes an independent verification of the results proposed by the research. He concedes that replication is often unavailable due to costs associated with fully recreating an experiment. In such cases, reproducibility can act as a minimum standard where the software and data can be re-analyzed.

Peng builds upon Claerbout et al. [18] by introducing reproducibility as a spectrum. He argues reproducibility exists in terms of the availability of the artifacts starting from not reproducible with nothing made available to code available to code and data available to linked and executable code and data. Full reproducibility, while not a substitute for "gold-standard replicability," can provide further proof for the validity of a claim. Thus, Peng defines replicability as a fully independent attempt to build further evidence for or against a scientific hypothesis and reproducibility as the re-analyzing of available code and data. While Claerbout et al. [18] provide a strict definition for reproducibility, Peng [44] provides a strict definition for replicability (e.g., fully independent experiments) and reproducibility as a spectrum. Counterintuitively, Peng's definitions put fully reproducible (i.e., only using the original artifacts) as the closest substitute to replicability. Yet replication experiments would exist at the opposite end of the reproducibility spectrum because Peng's replicability is fully independent of the original authors. Thus, the closer one's experiment to full reproducibility, the farther from the replicability standard. This spectrum does not provide context for reproducing results by using only part of the available artifacts.

Takeaway *The Claerbout Terminology provides the foundational understanding of reproducibility, yet fails to define replicability. Fundamentally,* **reproducibility and replicability are interchangeably treated as both properties and actions.**

3.2 Definitions by Organizations

The growing concerns for reproducibility in the mid-2010s prompted organizations to undertake investigations into defining reproducibility. Although numerous organizations have addressed and proposed reproducibility definitions, we discuss two prominent organizations that influence the Security community's understanding of reproducibility, the Association for Computing Machinery (ACM) and the National Academies of Science (NAS).

ACM - The ACM released their first definitions for reproducibility derived from the International Vocabulary of Metrology in 2016. In contention with the Claerbout terminol-

[3] They modernized this work into a formal framework called ReDoc in Schwab et al. [51].

ogy, they define *Repeatability* as "same team, same experimental setup", *Replicability* as "different team, same experimental set up" and *Reproduciblity* as "different team, different experimental setup" [20]. After identifying the contentions between the ACM's definitions and broader computational sciences, this version was later unified with the Claerbout terminology in 2020 with *Reproducibility* as "different team, same experimental setup" and *Replicability* as "different team, different experimental setup" [21].[4] The ACM measures these properties, which results in badges awarded to published papers.

The ACM provides three badges, *Artifacts Evaluated*, *Artifacts Available*, and *Results Validated*, each with levels of the property contained within the badge [21]. The Artifacts Evaluated badge contains two levels, Functional and Reusable. A functional award is "documented, consistent, complete, and exercisable." The associated repository must contain an inventory of the artifacts included and a sufficient description to enable the artifacts to be exercised (i.e., documented). A consistent artifact is relevant to the associated paper, contributes to the main results of a paper, and is complete if "all components relevant to the paper in question are included" [21]. Finally, a functional artifact is exercisable if all included software runs and generates results. The ACM defines a Reusable artifact as "significantly exceeds minimal functionality." With no additional properties beyond Artifact Evaluated - Functional, the Reusable badge contains "carefully documented and well-structured to the extent that reuse and repurposing is facilitated" [21] artifacts. An Artifact Available badge is awarded when the authors make the artifacts available through a publicly accessible archival repository.

The Results Validated badge contains two levels, Results Reproduced and Results Replicated [21]. Results Reproduced is awarded when a subsequent study by an independent person or team obtained the main results "using, in part, artifacts provided by the author" [21]. Results Replicated is awarded when an independent person or team obtains the main results "without the use of author supplied artifacts" [21]. We note that the badging does not fully reflect their definitions. The ACM provides a strict definition of reproducibility, but the badges do not follow this strict definition. For example, the Results - Reproduced badge only requires using "in part, artifacts provided by the authors" [21], but the revised definition of reproducibility "means that an independent group can obtain the same result using the author's own artifacts" [21]. Thus, these definitions do not fully align.

Takeaway *The practice of reproducibility and replicability does not always align with the conceptual definitions, leading to inconsistent application and miscommunication.*

NAS - As the growing concerns for reproducibility became more public, in 2017 the United States Congress entreated the National Science Foundation (NSF) to engage the NAS to investigate reproducibility and replicability and provide recommendations to improve the field. The NAS released a book on reproducibility and replicability in 2019 [38]. This body of work represents a synthesis of over two years of exploratory committees to define reproducibility and replicability. The NAS explored numerous areas of research (e.g., Biology, Biomedical, Economics, Psychology, and Medicine) to understand current undertakings of reproducibility and aims to unify definitions across disciplines. This report highly focuses on meta-science research and analyzes the applications of reproducibility and replicability, providing extensive recommendations for fields of science as well as guidelines for conducting reproducibility studies.

To differentiate between reproducibility and replicability, the NAS first focuses on the underlying concepts. They pose the following iterative questions related to reproducibility and replicability:

1. "Are the data and analysis laid out with sufficient transparency and clarity that the results can be checked?"
2. "If checked, do the data and analysis offered in support of the result in fact support that result?"
3. "If the data and analysis are shown to support the original result, can the result reported be found again in the specific study context investigated?"
4. "Finally, can the result reported or the inference drawn be found again in a broader set of study contexts?"

From these questions, the NAS defines the demarcation between reproducibility and replicability between questions (2) and (3). Reproducibility is focused on verifying the existing claims of a paper by checking the available artifacts of a study. Replicability is defined as testing the limits of the study and broader implications. This introduces two new properties of reproducibility and replicability, indirect and direct. Direct studies imply that there was an experimental procedure that was conducted, whether the exact same artifacts of the authors (i.e., reproducibility) or new data and experiments (i.e., replicability). Olszewski et al. [40] performed an indirect and direct reproducibility study on ML Security.

We note that the usage of indirect is not the same between reproducibility and replicability. While an indirect study of reproducibility assesses "the extent of the availability of computational information" [38], an indirect replicability study is not defined. Upon examining the studies the report classifies as indirect replicability studies, these studies appear to focus on assessing the validity of the methodology used across several studies (e.g., "49.6% of the articles with null hypothesis statistical test (NHST) results contained at least one inconsistency" [38]). Further, the NAS report claims that direct reproducibility is rarer than indirect reproducibility due to the difficulty of conducting a study, showing that 9/13 surveyed reproducibility studies are indirect. Although no formal definition for indirect replicability is provided, they highlight 19/22

[4] Liberman [37] identifies that the opposing definitions were introduced to computational science by Drummond [19].

direct replicability and 3/22 indirect replicability studies.

The NAS definitions motivate two ideas. First, it provides a term for a property implicitly identified by Peng, indirect; there is a limit to the reproducibility of a paper due to the availability of artifacts. Second, it motivates the difference between replicability and reproducibility by the purpose of conducting the study. For example, reproducibility is checking that the artifacts support the main results (e.g., Question 2), while replicability is testing the finding in new contexts (e.g., Questions 3 and 4).

Takeaway *The current taxonomies do not have a unified convention for desired properties. There is a missing connection between the purpose, outcome, and realization of reproducibility and replicability.*

3.3 Expanded Definitions

The Claerbout Terminology provided the foundation of computational reproducibility, and organizations adopted versions of it to promote reproducible research. As noted above, there are several limitations within this theoretical understanding of reproducibility and its adaptation to computational science. In this section, we discuss works that build upon the previous work by relying on different concepts to derive a taxonomy.

Gomez et al. [22] - In synthesis of a survey of 20 replication classifications, Gomez et al. derive a framework for replicability motivated by the purpose of the replication. They define the operationals (i.e., parts) of an experiment as the protocol, the operationalization, the population, and the experimenter. The protocol is the experimental methodology followed to perform the experiment. The operationalization is the control and response and how this effect is measured. The population is the studied population, and the experimenters are the ones performing the experiment. Gomez et al. designed the framework for the function of the replication experiment. By modifying the operationals, the function of the experiment changes. They list six functions of replication: (1) to control for sampling error; (2) to control protocol independence; (3) to understand operationalization limits; (4) to understand population limits; (5) to control experimenter independence; and (6) to validate hypotheses. As an example, modifying the experimenters of the replication fulfills the function (5), to control for experimenters' independence. From these modifications, Gomez et al. [22] define literal replication as "new runs of the experiment on another sample of the same population", operational replication as a broader category of replications that modify operations of the experiment, and conceptual replication as changing every operation of the experiment. Operational replication can further be denoted by the changes in the experiment (e.g., changing the experimenters would be Operational Experimenter Replication).

Of particular interest in this framework is the function of the replication. Gomez et al. [22] primarily derive these functions from Schmidt [50]. The purpose of a replication study can be defined before conducting the experiment. Function (1) controlling for sampling errors determines that the results do not happen by chance. This function is achieved through literal replication (i.e., running the same experiment on a new collection of data from the same population). Function (2) identifies that changing the protocol does not modify the results. Schmidt [50] formulates this as "controlling for artefactual results (internal validity)." Thus, Function (2) ensures that the result is not due to a tool or methodology (e.g., faulty thermometer). Function (3) controls the operationalizations of the experiment ensuring that the result is not due to how the response is measured. For example, Function (4) assesses the extent the result is unique to the studied population. This is formulated as generalizability by the NAS [38]. Function (5) determines whether the result is independent of the original researchers. Schmidt [50] calls this a control for fraud. Function (6) seeks to validate the hypothesis. Gomez et al. [22] define conceptual replication to meet this function.

This framework focuses on the function of replication to derive their definitions of replicability. We note that this framework does not consider Claerbout's reproducibility. As such, the functions of replicability are not comprehensive to validity. Although designed for Software Engineering, this framework does not intuitively map to computational experiments. As Peng points out, some methodologies rely on massive datasets that additional studies cannot recreate. Further, this framework focuses on complete replicability that does not use any experimental artifacts from the original authors. As such, although they map literal replication to repetition and conceptual replication to reproduction, these definitions are not consistent with the Claerbout terminology.

Takeaway *The purpose of a validity study is inherently connected to the changes made within the experiment, yet no taxonomy currently unifies the function, the availability, and the actualization of the experiment.*

Goodman et al. [23] - Goodman et al. define three types of reproducibility: method reproducibility, results reproducibility, and inferential reproducibility. Method reproducibility is the provision of "enough detail about study procedures and data so the same procedures could, in theory or in actuality, be exactly repeated." [23] Although they focus their discussion on the theoretical ability to reproduce experiments, they capture a similar property as the NAS, that there exists a theoretical ability to be methods reproducible (i.e., indirect) and an actualization of method reproducibility (i.e., direct). They describe results reproducibility as their version of replicability. It is "obtaining the same results from the conduct of an independent study whose procedures are as closely matched to the original experiment as possible" [23].

They briefly mention robustness (i.e., the stability of results across variations of the experiment) and generalizability

(i.e., "the persistence of the effect outside of the experimental framework" [23]). Inferential reproducibility refers to the interpretation of the results after a study is conducted (i.e., a paper is inferential reproducible if an independent team comes to the same conclusions regardless of methodology). While an independent team may run similar experiments and achieve results reproducibility, they may fail to be inferential reproducible if the independent team draws different conclusions. This discussion uses Bayesian statistics. "If a finding can be reliably repeated, it is likely to be true, and if it cannot be, its truth is in question" [23]. Finally, they identify problems that result in why an experiment may not be reproducible. This implicitly points to Gomez's functions. Further, they discuss that the results of reproducibility studies often cause disagreements between the original authors and the reproducer. This is caused by the misconception of terminology and misrepresented methodology choices. The demarcation between these three definitions appears to be in what part of speech they are. Method reproducibility is described as a property of published research, results reproducibility is an action taken by other researchers, and the unclear terminology can result in disagreements about the reproducibility study.

Takeaway *The outcome of validity experiments builds evidence towards a scientific theory. The current terminology does not reflect that replicability is iterative and cannot compare between two independent validity studies.*

Gundersen [26] - While the previously discussed frameworks focused on properties or functions to derive functions of reproducibility, Gundersen [26] derives their framework for AI/ML reproducibility from the scientific method. They motivate the design of their framework by breaking the MNIST dataset [36] into the step-by-step tasks needed to re-implement the methodology. For example, to collect handwritten digits, one would have to first gather writers, second, have them write the numbers, and finally digitize the numbers. Each other task within an experiment can be further broken down into sub-tasks (e.g., data processing can be broken into removing outliers and normalizing the data).

Gundersen defines three degrees of reproducibility: outcome reproducible, analysis reproducible, and interpretation reproducible. Each of these definitions is defined as the outcome of an experiment (i.e., research is not "X reproducible" until the experiment has been attempted). Outcome reproducible is defined as the same outcome from the original experiments. Analysis reproducible is when the outcome is not the same, but the same analysis leads to the same interpretation. Interpretation reproducible is when both the outcome and analysis are different but lead to the same interpretation.

The three degrees of reproducibility are affected by what is made available by the original authors. For this, Gundersen defines four reproducibility types: R1 - Description, where only the description of the experiment is used; R2 - Code, where the code and description are used to reproduce the experiment; R3 - Data, where the description and the data are used; and R4 - Experiment, where the documentation, data, and code are all used to reproduce the experiment. Gundersen is one of the first to provide definitions that reflect what is used from the original experiment (e.g., the same data is used in the reproduction experiments). The reproducibility type is iterative though and, as such, does not handle varied methodologies. This framework demonstrates that reproducibility and replicability are affected by what is used from the original experiments. While an improvement to the ACM's definition, the same data oversimplifies the rest of the experiment. Although it addresses several deficiencies in previous frameworks, it does not account for what the original paper can achieve. Thus, Gundersen treats reproducibility as an outcome of an experiment and fails to unify reproducibility and replicability as properties and actions.

Takeaway *Current frameworks are not robust to the varying implementations of the scientific process. Further, reproducibility as an outcome is affected by what artifacts are used from the original experiments.*

3.4 Lessons and Problems

This body of work summarizes over 30 years of work, yet several deficiencies remain with how validity is defined and formalized. In this systematization, we see that reproducibility is described as a property and as an action of an experiment. For example, an experiment could be described as reproducible because one *could* run the experiments or reproducible because one *has* run the experiments. We note that this is part of the confusion one runs into when discussing validity. While Gundersen attempts to address this by explicitly stating that reproducibility is based on the conclusion of a validity experiment, this ignores the limits to the extent the experiments could be reproducible. Further, this does not allow comparison between subsequent validity studies. As such, *there is no way to reason or compare between two independent validity studies*. We address this limitation in our framework.

No taxonomy explicitly establishes that reproducibility and replicability are not mutually exclusive. An experiment could be reproducible but not replicable, replicable but not reproducible, or both reproducible and replicable. For example, if a paper has provided all artifacts, one could reproduce all of their experiments by running their code and ensuring that the artifacts run and output a similar result. One could also replicate the study by coding these experiments from scratch to demonstrate that it is replicable at some level. These taxonomies treat reproducibility and replicability as an ill-defined hierarchy when they control for different outcomes. None of the previous work is meaningfully deployed, as they fail to provide a robust and actionable framework for assessing and comparing experimental validity across studies.

Finally, none of these taxonomies adapt to the needs of the security community.

4 Replicability in Security

All fields of science face challenges in achieving reproducibility and replicability. The sensitive, adversarial, and complex nature of Security Research exacerbates existing obstacles. In this section, we highlight unique aspects of Security Research, including threat models, and notable challenges in Computer Security. Further, we provide an exploration of the primary proposed solution, artifact evaluation committees (AECs).

4.1 Security Challenges

Threat Model. There is a distinct challenge in adapting the previous frameworks discussed in Section 3. Unique to Security Research, a threat model is often used to declare the capabilities of an adversary and inform the experimental procedure. This model often informs system configurations, analysis methods, and the scope of experiments in ways that are not accounted for in reproducibility and replicability taxonomies. For example, an attack paper expresses the capabilities of the adversary within the threat model, allowing certain actions or attack vectors. In a reproducibility study, this means that the threat model must be the exact same. For replicability, For replicability, the threat model does not need to be the same but clearly defines what differences were made. If a replication study adopts a different threat model, either implicitly through environmental constraints or explicitly through design, the replication may fail, not due to flaws in the original work, but due to a divergence in assumptions about the adversary. For example, Provo et al. [47] propose creating honeypots to identify adversarial capabilities. Replicating these experiments will fundamentally change as the adversary may have knowledge and adapt to the honeypot.

Further, Security does not contain the ability to express this replication beyond a different threat model. If a publication proposes a defense to a vulnerability, validators could test the bounds of the defense by exposing the system to different adversarial threats. Characterizing the extent to which a defense or attack is applicable is an important part of evaluating Security Research, and the Security Community lacks a robust communication framework for replication studies.

Challenges. As Security Research increasingly becomes interdisciplinary, it shares several challenges for validity that other fields experience. While traditional fields focus research efforts in one direction, Security Research will identify a new facet of the problem. A network protocol researcher will address the speed of the protocol, but a Security Researcher will focus on the integrity of the protocol. In another example, traditional machine learning optimizes the performance of the task given, but Jia et al. [29] propose a defense for model attribution. The experimental process for both these problems will be similar. Security Research will face many of the traditional problems in other fields, albeit from a different perspective. However, the adversarial nature of Security Research creates additional difficulties in achieving reproducibility or replicability such as: time to impact of research, effect on stakeholders, transition to practice, complexity of systems, and adversaries using developed research. The impact of Security Research is often quicker and can directly affect multiple stakeholders. Security Researchers focus on current research problems, but the rate at which technology changes is rapid. The complex systems that Security Researchers measure and deploy are inherently difficult to reproduce and replicate [30]. The systems studied contain numerous stakeholders that can be affected. For example, Reaves et al. [49] collected over 400,000 text messages sent to public online SMS gateways over the course of 14 months and identified inherent problems in two-factor authentication. Publishing the dataset could result in massive privacy problems by inadvertently disclosing users' phone numbers, verification codes, and other private data. Preparing data for reproducibility or replicability efforts has adverse effects on multiple stakeholders. While this is observed in other fields (e.g., computational chemistry), the impact can directly affect everyday users. For example, a vulnerability to Bluetooth devices could affect billions of devices [11]. Security Research can often have an immediate impact on billions of users.

Security Research usually performs responsible disclosure, where the authors disclose to a technology vendor an identified vulnerability. Even after disclosure, publishing or replicating attacks can have adverse effects on stakeholders. For example, a published attack can result in a vendor issuing a patch, but if a user does not apply the patch, the attack can adversely affect the user. For example, Tian et al. [55] identified attacks using AT-commands present in over 2,000 Android devices across 11 vendors. They worked with the vendors to patch these vulnerabilities, but as the problem is in firmware, publishing an attack could comprise millions of devices. Thus, there is often an incentive for Security Researchers to not publish a proof-of-concept attack. In the case of identifying the bounds of Security Research, replicating experiments can cause adverse harm to other stakeholders. Unlike fields such as medicine, where the transition from controlled experiments to real-world application is mediated through formal phases (e.g., clinical trials), Security Research often moves from lab experiments to practical deployment with little formalization. For example, a patch to vulnerable software may be implemented without additional testing, which has resulted in the addition of new vulnerabilities [39].

Takeaway *The Security Community does not have a robust framework for communicating validity studies.*

4.2 Artifact Evaluation Committees

In response to growing calls to address reproducibility concerns, artifact evaluation committees (AECs) accept artifacts from accepted papers at a conference. The goal of the AEC is to first ensure a level of validity through reproducibility, and second, consolidate research artifacts for potential future use. AECs were first introduced in computer science at the ACM Conference on Foundations of Software Engineering (FSE) in 2011 [32]. While FSE did not continue with AECs, other software conferences adopted AECs such as: European Conference on Object-Oriented Programming (ECOOP) in 2013, Object-Oriented Programming, Systems, Languages & Applications (OOPSLA) in 2013, Principles of Programming Languages (POPL) in 2015, and Programming Language Design and Implementation (PLDI) in 2014 [1].

The first security conferences to adopt AECs were ACSAC and WiSec in 2017 [3, 4]. ACSAC notably required that any artifact that undergoes the artifact evaluation process must commit to making the artifact publicly accessible online. While ACM CCS and NDSS created AECs in the years 2023 [9] and 2024 [10], respectively, USENIX Security was the first Tier-1 security conference to adopt an AEC in 2020 [5]. In 2022, USENIX Security adopted a standardized artifact appendix [8]. This resulted in a fully published artifact appendix in the proceedings [7] containing the artifact appendices submitted to the AEC. USENIX Security in 2025 adopted an Open Science Policy, requiring authors to make available the associated artifacts for a publication available online [6]. This access must be made permanently available and as such, websites such as GitHub, are not allowed. IEEE S&P will be adopting an AEC for 2026 [12]. We note that AECs in their current forms only address reproducibility should everything be made available.

5 Framework

From our systematization, we identify that there are several shortcomings within reproducibility and replicability frameworks. As such, the current definitions of reproducibility and replicability do not attain a unified framework and cannot describe the complexity of security research. Our goal is not to provide new "words" or "adjectives" and provide a commentary on how researchers should favor one definition over the other. This becomes increasingly complex to adapt threat models into. Instead, we introduce a new framework for characterizing validity. Our goal is to provide a framework that addresses the misunderstandings and innate properties of previous frameworks that can be used in a security context (e.g., adversarial attacks). Rather than treating adversarial models as fixed or background context, our framework should allow authors and validators to formally specify the threat scope under which findings hold. Thus, any framework should enable clearer comparisons across studies and facilitate meta-

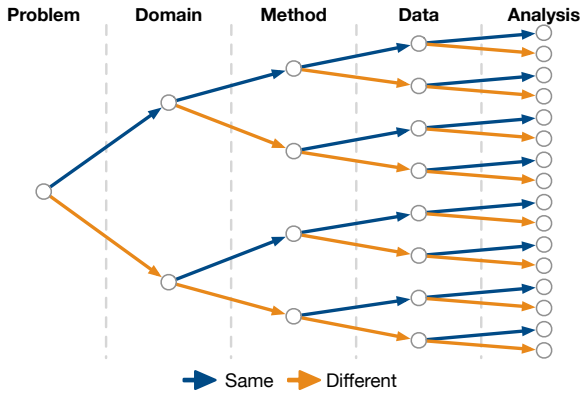

Figure 2: The Tree of Validity shows every possible validity experiment by using the same or different parts of an experiment. We emphasize that this tree can be modified to express different methodologies by swapping or creating new layers in the perfect binary tree.

analyses that account for differences in threat modeling. In this section, we discuss the goals we aim to achieve and the properties this framework should have, formalize the framework, and show it unifies with previous understanding of reproducibility and replicability. We will then use this framework to map prior efforts in Section 6.

5.1 Goals

From the previous taxonomies and identified security challenges, we define reproducibility as a form of validity of the physical manifestations of experiments (e.g., code or data). Replicability is designed to build evidence towards or against the initial hypothesis. To address the limitations in previous frameworks, our framework should meet the following goals:

- *Unify with previous frameworks*. The previous taxonomies and frameworks establish a foundation for the field of validity. As such, any proposed work should not oppose these definitions. Our framework should aim to consolidate these definitions into one structure.
- *Reconcile validity as a property and as an action*. Much of the confusion between the taxonomies is due to treating published work as reproducible in that it has been reproduced and as reproducible in that it could be reproduced again in the future. Our framework should address the potential for validity and reconcile the inherent limitations due to the available experimental artifacts.
- *Accommodate different fields and varied methodologies*. The Security Community encompasses numerous subfields, and security research is growing in complexity. As such, the proposed framework should accommodate the adversarial features of security research and complicated methodologies.

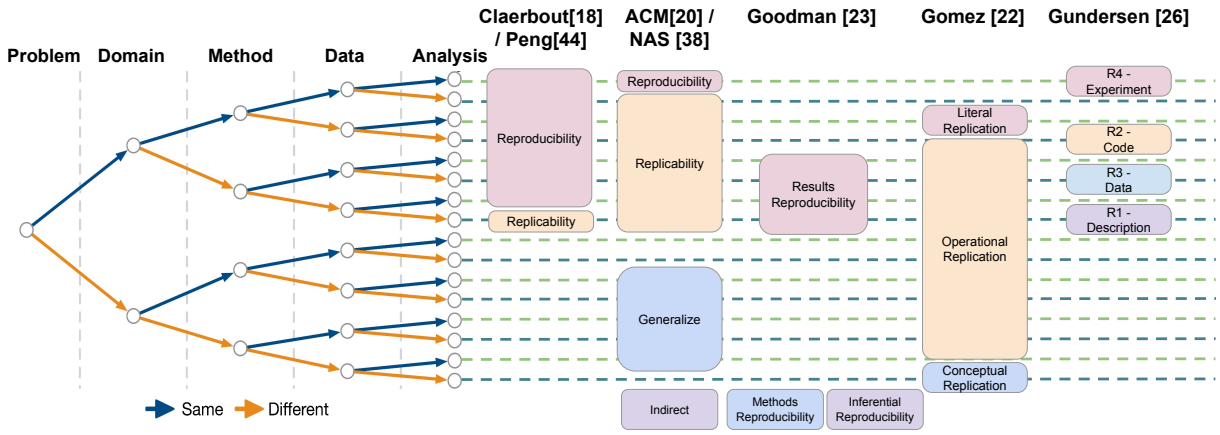

Figure 3: This figure shows where all of the previous definitions map within our framework. We can see that no framework covers every definition and that several definitions can mean similar things. For example, the NAS definition of replicability maps to several paths through the **Potential** Tree of Validity.

- *Quantify differences between validity studies*. When validity studies are conducted, they indicate the changes made by calling the study either reproducible or replicable. Mapping such complex tasks to two or three words is incomplete and does not allow for direct comparisons to be made. As such, our framework should succinctly express the differences between studies.

5.2 Tree of Validity

To motivate the construction of our framework, consider the perfect case for reproducibility where the published research provides all experimental artifacts. In this case, it would be possible to use any part of the original experiment to conduct a validity experiment. We broadly formulate our framework around the following experiment: an experiment addresses a *problem* within a *domain*. We refer to this as the *setting* as seen in Figure 2. A *method* is applied to create *data* and then *analysis* is conducted on the data, referred to as the *process*. Each of these parts of an experiment will create the layers of a perfect binary tree. The decision of the binary tree is whether the part remains the *same* or *different*. We call this the Tree of Validity (ToV) and visualize it in Figure 2. This Tree of Validity shows every iteration of possible validity experiments. For example, following the upper path through the ToV (i.e., keeping everything the same) would create a validity experiment that Claerbout would describe as reproducibility. We show how each definition from Section 3 maps to the Tree of Validity in Figure 3. Thus, the comparison between definitions is no longer relative but quantifiable.

The layers of the Tree of Validity are not static, although changes to the *setting* are most likely not to occur. There are numerous enumerations of using the same parts of the original experiment depending on how granular the tasks of the experiment are expressed. For example, the steps to the experiment could be a *methodology* that consists of a software and hardware component. Thus, the *method* layer can be split into a *software* and a *hardware* layer. In another example, an experiment may wish to split the *data* into a *train* and *test* layer for a machine learning problem. Further, the layers can be swapped or removed. For example, in some experiments, it may make sense to move *data* before *method* or remove either layer. Thus, this formulation allows the ToV to be dynamic to the experiment it is describing. A ToV can be constructed for all experimental methodologies, and we show examples of applying the ToV in Section 6.

5.3 PEC

By formulating our framework around the Tree of Validity, we can now describe several properties of validity. We define the **Potential**, **Execution**, and **Conclusion** as aspects of the framework. **Potential** - Often reproducibility and replicability are inherently limited by what experimental artifacts are made available, a notion identified by Peng, the NAS, Goodman et al., and Gundersen. While some assign definitions or provide a hierarchy, they fail to provide a cohesive formulation that identifies the **Potential** for validity that each published paper inherently has. The Tree of Validity shows the **Potential** for each validity experiment. Figure 4 shows the construction of a Tree of Validity for an experiment. In practice, the authors may not make every experimental artifact available (e.g., a massive or private/sensitive data set). Figure 4a shows a shorthand way of describing the experimental artifacts available which we refer to as the Seed of Validity (SoV). We construct this SoV, but the SoV can be constructed by the original authors. We further discuss this in Section 7.

In Figure 4a, we see that the *method* is not available. This could be, for example, the system code to run an experiment. As it is not available, this limits the paths one could take

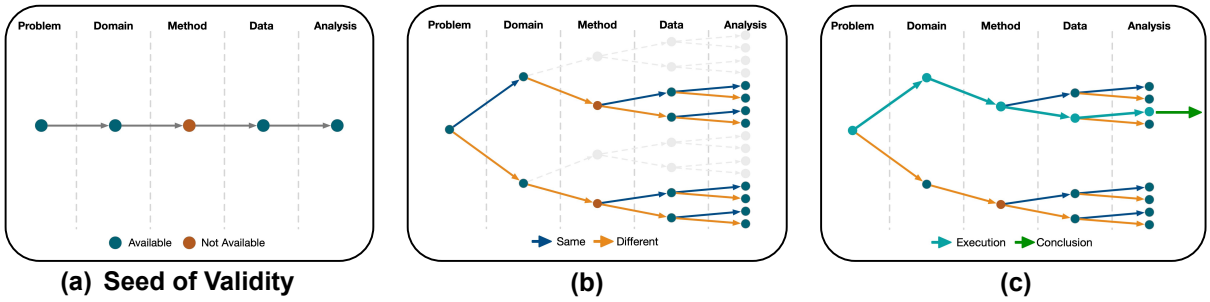

Figure 4: (a) We see what is made available (e.g., problem, domain, data, and analysis) and what is not available (e.g., method). This shortened figure is referred to as the Seed of Validity (SoV). (b) We can construct the **potential** tree from (a). Since the method (e.g., experimental code) is not available it limits what end nodes we can reach. (c) An **execution** through the **potential** tree is a validity experiment that results in an outcome that we can draw **conclusions** from.

through the tree. We show the **Potential** tree in Figure 4b which reflects the possible **Executions** a researcher could take to confirm validity. Since the *method* is not available, it is impossible to reproduce the experiment. This does not limit other forms of validity (e.g., replicability). For example, a validator could implement their own method on the same data and analysis.

Execution - The **Execution** of a validity experiment manifests as a path from the root node of the **Potential** tree to a leaf node. Thus, an **Execution** is the act of conducting a validity experiment. Figure 4c shows an execution of a validity experiment in black. We can now compare validity experiments as paths through the **Potential** tree. An execution that follows the path where every edge is the *same* as the original experiment is an execution of reproducibility. If two executions have the same path through the potential tree of validity and at least one edge of the path walks *different* experimental artifacts, then the executions are not inherently the same. For example, if two executions test the method and analysis in a *different* domain, the two domains do not have to be the same. Every execution describes an experimental methodology and, thus, can create its own Tree of Validity.

Conclusion - After executing a path through the **Potential** tree, the output results in a **Conclusion**. With varying experiments, this can result in a figure (e.g., a bar graph showing results) or a number (e.g., relative differences in accuracies).

Our framework is conceptually simple, yet describes validity in a different way than previous definitions. As such, we can express how the availability of experimental artifacts affects the potential validity studies. Furthermore, we can describe previous definitions in terms of our ToV. An experiment's ToV describes the potential. We can modify the ToV such that it fits any experimental process. This framework provides a unified communication strategy between validity studies.

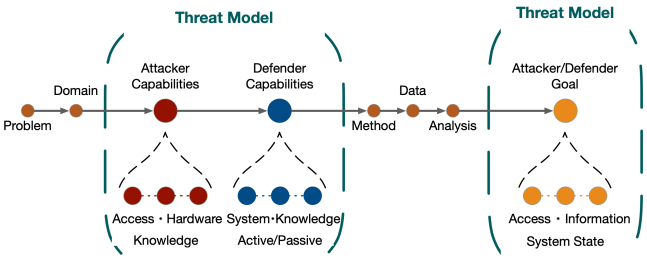

Figure 5: Demonstrating the threat model in a ToV. The attacker and defender capabilities are included in the ToV, including but not limited to: knowledge, access, hardware, system state, active or passive defense, etc.

5.4 Security Utility

As discussed in Section 4, Computer Security faces several unique challenges to reproducibility and replicability. The threat model is a fundamental aspect of most security papers that describes attacker capabilities and defender capabilities. These underlying assumptions shape the methodology and analysis of a security paper. The ToV provides a simple way to include these underlying assumptions and an expression for various adaptive attacks or defenses. We consider two perspectives: the adversary's and the defender's. We emphasize that the goal and utility of the ToV are in replicability studies.

When demonstrating a novel attack, authors typically specify the target system and delineate the capabilities of the attacker. This defines the operational boundaries within which the adversary demonstrates the exploit. For example, in vulnerability discovery using fuzzing, the assumed adversary may have access to binaries, source code, or system interfaces. A replication study using a different system configuration or set of attacker capabilities may not yield the same results, not due to flaws in the methodology, but due to a misalignment in threat modeling. From the defender's point of view, the threat model encapsulates how the adversary can interact with the

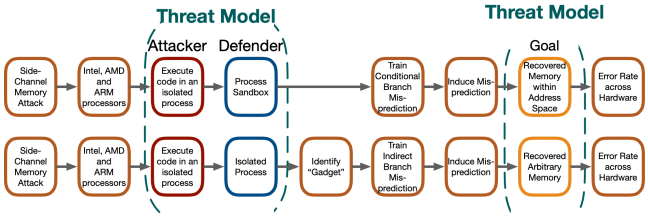

Figure 6: The Spectre attacks from Kocher et al. [31] can be modeled as executions through a ToV.

system. For example, a detection mechanism (e.g., intrusion detection system) will have specific requirements of what constitutes a vulnerability, what the attacker can perform, and how the system can measure the problem space. An adversary's goal is explicit in the threat model and decides how the system is evaluated. In modeling defenses for distributed denial-of-service (DDoS) attacks, the attack model encapsulates the performance of the targeted system under a DDoS attack. In contrast, an authentication mechanism is optimized for minimizing the false rejection rate and the false acceptance rate. By using a ToV, we can define the capabilities of an attacker that a defense mechanism is robust to. This is visualized in Figure 5.

Of primary interest in security research is how a defense functions under an adversary aware of the defense (i.e., adaptive adversary), that tests the bounds of the research hypothesis. An adaptive adversary can be modeled as an execution through the ToV with different attacker assumptions and modifications within the methodology. The adaptive adversary can interact with the defense system to modify the attack. The flexibility of the ToV can adapt to multiple scenarios. Thus, the ToV can accommodate a threat model and communicate the design choices of an experiment.

6 Case Studies

To demonstrate our framework, we examine several case studies that demonstrate the challenges with Security Research replicability. First, we create SoVs for a well-known attack and a machine learning defense. Second, we examine replicating a USENIX-award-winning artifact.

6.1 Spectre Attacks

Modern processors use branch prediction and speculative execution, and Kocher et al. [31] leverage this to leak sensitive information from a processor. Specifically, the Spectre attacks leverage that a processor will attempt to guess the destination and execute the thread to improve performance. The Spectre attacks exploit two variants using conditional branches and indirect branches. In this paper, Kocher et al. demonstrate the attacks using native code, JavaScript, and eBPF.

Security. Kocher et al. [31] demonstrate several threat models and attacker capabilities. For example, one variation requires running code on the target processor to leak information (e.g., an AWS server). Defender capabilities do not exist in this threat model since they leverage inherent features of the processor. As such, the processor has a simple defender capability, process isolation. Replicating this attack remains an open and interesting problem that would identify to what extent processors defend against Spectre attacks. Furthermore, the Spectre attacks affect billions of devices using Intel, AMD, and ARM processors. Thus, publishing proof-of-concepts in the interest of replicability can raise ethical concerns. This is further discussed in Section 7.

The several variations of the Spectre attacks can be modeled in a SoV demonstrating the different variations across several threat models. The two variations declare different capabilities in the threat model, and the three practical demonstrations demonstrate different attacker goals. Figure 6 depicts two of the attack combinations consisting of exploiting branch misdirection JavaScript and poisoning indirect branches in native code. We note that the use of the ToVs here shows the differences between the experimental methodologies of the two variants of the attack. The two SoVs demonstrate different defense capabilities and attacker goals. The attack variations exploit different parts of a process. The goal of the JavaScript branch misdirection is to violate the sandbox which is demonstrated by leaking memory from another sandbox. This is done by inducing the CPU to speculatively execute the incorrect direction of a branch. The goal of poisoning the indirect branch is to violate the process isolation. This is accomplished by identifying the memory location (i.e., "a gadget") the attacker wishes to leak, then training the Branch Target Buffer (BTB) to mispredict a branch from an indirect branch instruction. We note that, while similar to an attack tree, the SoV and potential ToV include the experimental methodology for evaluating the system, which is outside the scope of an attack tree.

Appendix C of Kocher et al. [31] shows a proof-of-concept attack for the x86 processor in C, but any other experiment associated with the attack is not made available. As the attack is not made publicly available, the potential ToV does not include an execution that would coincide with reproducibility. The ToV describes the environment and assumptions that inherently affect the experimental methodology. This is informed by the measurement of the attack. For example, Kocher et al. identify that there are error rates within the recovery of a memory cache, due to identifying the memory location done through timing. While they provide a workaround to severely limit the error rate of the Spectre attacks, the statistical nature of the attack can be replicated. In so doing, a validator would replicate the distribution of the attack success rate. Any future validator would consider the hardware, attacker capabilities, and goal of the attack, including measuring the error rate across different hardware setups. When

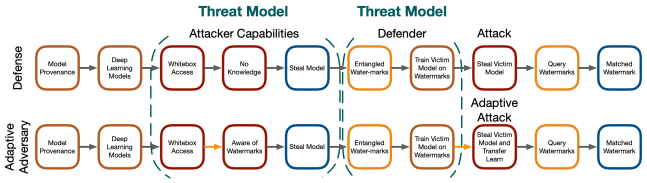

Figure 7: Entangled Watermarks by Jia et al. [29] demonstrate defender capabilities. The threat model is easily adaptable to show differences in an adaptive adversary. Thus, replicating their results can clearly be shown.

running a new execution through the potential ToV, the validator can modify the defender capabilities to identify whether a proposed defense subverts the Spectre attacks.

Takeaway *The ToV can demonstrate the threat model with the potential for replicability. No previous taxonomy can account for different assumptions of the experimental environment.*

6.2 Entangled Watermarks

Due to the expensive nature of collecting data, processing data, and training a model, machine learning models are primary targets for being stolen. Watermarks, unique input/output pairs known to the defender, can be employed to identify whether a model has been stolen. Jia et al. [29] propose entangled watermark embedding that improves on previous defenses by pulling the watermarks from the task distribution of the data, compared to random watermarks. This results in an overall improvement against model stealing attacks.

Security. This case study demonstrates how a defense can be modeled in a ToV. Jia et al. [29] evaluate the entangled watermarks through several adaptive adversaries. These adaptive adversaries are modeled as executions through the ToV as the capabilities of the adversary are different. Further, the artifacts available dictate the level of potential ToV, as the entangled watermark system is available, but the experiments that evaluate the entangled watermarks are not. Thus, any replication of the experimental methodology can train models on entangled watermarks.

The ToV for Entangle Watermarks can be seen in Figure 7. The top SoV demonstrates the original threat model, attack, and defense proposed. Through the ToV, we can capture the attacker capabilities (e.g., white-box), proposed defense (e.g., watermarks), and the attack leveraged. Jia et al [29] additionally demonstrate the robustness of entangled watermarks by applying four adaptive adversaries. We show the transfer-learning adaptive adversary that can be seen in Figure 7. In this case, the adversary applies an additional step in their attack and is fully aware of the presence of a watermark. This changes the underlying assumptions of the system. The adversary uses similar data to the training distribution and finetunes the model on this new data to remove the entangled watermark. We can see that the attacker knowledge is different as well as the model extraction.

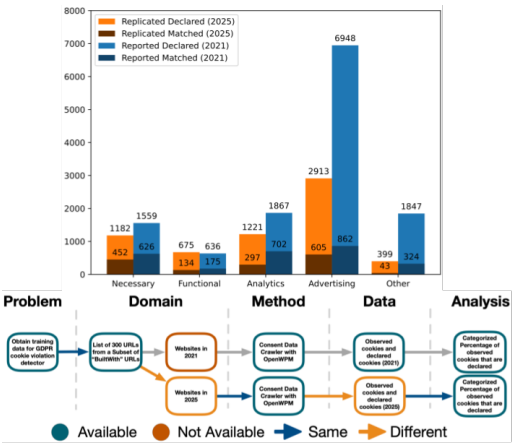

Figure 8: After running Bollinger et al.'s artifacts, we obtain fewer cookies than their reported results due to updates to Cookie policies and websites no longer active. Beneath the graph is our walk through the ToV of this experiment.

Jia et al. [29] provide the code artifacts to train and test the model in a CleverHans [43] implementation. The repository includes the code to watermark the experiments they perform on several models including CNNs, RNNs, and ResNet [27] across several datasets including MNIST [35], Fashion MNIST [57], CIFAR-10 [33], CIFAR-100 [33], and Google Speech Commands [56]. Although the code is available to train the various models on their associated tasks, the repository does not include the code associated with creating figures or running transfer learning experiments. Thus, the potential ToV is limited by the available artifacts. The authors discuss how various implementations of hyperparameters can affect the results. Thus, they provide an intuition on testing the bounds (i.e., replicating) of the experimental methodology.

Takeaway *The ToV includes differentiating between the perspective of an attacker and a defender. Further, the ToV can represent several experiments as executions through the tree.*

6.3 Artifact Awards and Systematizations

In this section, we demonstrate one practical walk-through of replicating a USENIX artifact award winner [16]. We provide other examinations of USENIX artifact award winners [59] in Appendix A.1. Further, we demonstrate that Systematizations of Knowledge (SoK) can be modeled as repeated executions of ToVs using two recent SoKs on audio deepfakes [34] and web crawlers [52] in Appendix A.2.

Bollinger et al. [16] automate GDPR violation detection of cookies. This work was awarded the 2022 Distinguished Artifact award and demonstrates several interesting aspects of our framework and challenges within Security Research. Their methodology consists of collecting the websites from their domain and training an ensemble decision tree on the data. From this data, they map each cookie to a purpose and identify potential violations from the various policies (e.g., Google Analytics cookies not being labeled for analytics). They create a baseline model, which acts as an execution through the Tree of Validity, and construct a baseline model by querying Cookiepedia [41], which contains labels from manual analysis on the purpose of cookies. The execution of the baseline model only differs in the method.

Security. When measuring the state of a complex system, the potential ToV is limited by the domain. The experiments can be massively different, as networks have convoluted interactions. Further, walking through a ToV with everything the same can be practically infeasible.

One of the interesting aspects of this case study is the domain. The domain considers websites taken from the Tranco [46] ranking of May 5th, 2021 that use a consent management platform (CMP). Time is an inherent part of the domain. The authors dutifully note that reproducing these experiments from scratch is infeasible, because of changes in the state of the internet.

We run the experimental artifacts and find that 1,032 of the 6,940 (14.9%) of the URLs no longer resolve.[5] This is most likely due to websites no longer being hosted at the listed URL if they even exist at all. We show the replicated data collection in January of 2025 in Figure 8. Of interest, are the functional and advertising cookies. The advertising cookies in our run show only 42% of the original declared cookies. In functional, we see a slight increase in this number. We show the path through Bollinger et al.'s SoV in Figure 8 underneath the graph. As such, reproducibility experiments conducted on their artifacts will only result in execution paths that start at an internal node. For example, we can train their model with the processed training data, but we cannot recollect the complete dataset. The experimental process in Bollinger et al. cannot be recreated from start to finish as this changes. They discuss this in their experimental artifact and provide intuition on how the data collection will change over time.

Takeaway *Even when* **all** *experimental artifacts are made available, reproducibility is not always possible to execute. However, authors can proactively address where experimental methodologies will vary over confounding factors.*

[5] They limited the number of crawled websites in the available artifact to 6,940 to demonstrate the efficacy of the crawler.

7 Discussion

As the Security Community increasingly embraces open science practices and reproducibility, a critical gap remains: the absence of explicit definitions for replicability and reproducibility. This ambiguity complicates both communication and evaluation across research efforts. In response, our work introduces the Tree of Validity (ToV), a communication framework designed to help validators articulate differences in experimental methodologies for replicability and reproducibility. The ToV is adaptive and intended to integrate into existing research and teaching workflows by making the reasoning behind methodological choices explicit.

7.1 Replicability in Security

The Security Community is increasingly seeking to address concerns over reproducibility. While this has primarily focused on implementing AECs, we aim to provide an additional tool that can provide a clear distinction between reproducibility and replicability to promote a guided presence of validity. Not only are previous taxonomies limited in computational reproducibility, but they cannot accommodate the complexity of security research. Our framework allows for the perspective of a defender and an attacker that previous taxonomies could not demonstrate.

While defenses are often made available, attacks are not necessarily so. Making an attack publicly accessible can have serious real-world consequences, potentially enabling the exploitation of billions of devices. For instance, the Spectre attack affects Intel, ARM, and AMD processors. Thus, releasing a working exploit could facilitate widespread harm and exposure to liability. However, from a research perspective, the availability of such attacks can be valuable; they enable the study of attack chains, helping researchers understand how seemingly minor vulnerabilities may be leveraged as stepping stones to more severe vulnerabilities. Additionally, replicating known attacks in new contexts can help validate whether proposed defenses remain effective under variation or adaptation. As one recent study highlights, poorly implemented patches can inadvertently create new vulnerabilities [39]. Further, Hernandez et al. [28] showed that patches were reverted in some firmware images, causing vulnerabilities to reappear.

7.2 Current Progress and Adoption

The Security community introduced AECs to address growing concerns about reproducibility. Within our framework, this would be walking the top path of the Tree of Validity. Although USENIX is requiring the publication of experimental artifacts with the paper, this does not address the broader concerns of Validity. Reproducibility shows that the results presented in published research were obtained within the associated artifacts. Due to complex domains or limitations in cost,

validators cannot always execute the full reproducibility path and instead rely on the collected artifacts (e.g., measured data). Even in the best-case scenario, code does not always work. Systems are complex, and we need a language to discuss how our experiments can be affected. Our framework provides a communication path between original authors and validators. It presents a consistent way to communicate experimental methodologies over generic definitions of reproducibility.

Furthermore, our framework focuses on applying replicability standards to the Security Community. Our framework promotes exploration of the underlying hypothesis by multiple experimenters, incorporating a variety of differences (e.g., methods, data). By expressing experiments as ToVs, comparisons to prior work can be more precisely made and greater evidence towards the validity of a hypothesis can be achieved. Multiple efforts in an area can more naturally demonstrate progress in that space, and more clearly delineate novelty.

We recognize that the nodes within the SoV are subjective to the authors and validators. Fundamentally, it does not matter whether the choices made are subjective. Rather, if the original authors know the determinative factors in the experimental process and the validators agree, then *they are communicating in a consistent fashion*. If disagreements occur, this can be expressed by constructing new SoVs.

We encourage authors to include a SoV in the Appendix of their work. Modifying a pipeline figure, something most authors could include, can result in a detailed SoV. Further, they should identify areas that would cause variation in their results. For example, identifying time-dependent domains or sources of randomness. We expect that the SoV diagram can be expanded to include these variations. This expresses where the sources of variation occur. Further, conferences can include our framework in the publication process, and AECs can include this as a part of the artifact appendix. The additional figure presents limited difficulty for authors and a chance to argue for both the contribution (e.g., independent confirmation of another study) and novelty (i.e., a new contribution such as method or data) of their work.

7.3 Applications Beyond This Study

We applied our framework to a diverse set of Security papers, including side-channel attacks, machine learning defenses, audio deepfakes, cookie analysis, device integrity, and web crawlers. The framework is flexible, and we anticipate it can provide benefits to areas outside of Security. For example, human-computer interaction can use our framework to express changes to experimental methodologies or specific populations involved in a study. Machine Learning researchers can integrate this framework into their research to compare to similar work.

This framework could also be applicable beyond computer science. For example, biological studies are often complex and not deterministic (e.g., population sampling). One could use this framework to model the experimental process and discuss where their SoV introduces variability and how that can affect future replicability studies. Similarly, areas such as psychology could account for factors not included in prior studies (e.g., stress, fatigue, motivation) using this framework.

7.4 Open Problems

Our work proposes a framework for replicability that can be adopted throughout the Security community and acts as a foundation for the field of meta-science within the Security community. As such, we identify several future work areas. We define the conclusion as an outcome of an execution through the ToV, but we do not address what quantifies as a "successful" execution. While prior work [40,48] assigns arbitrary thresholds (e.g., within 10% of the results), this is subjective and falls into fallacies identified within null-hypothesis statistical testing. For example, there can be numerous reasons for results falling outside of 10% (e.g., time). Interpreting the results of validity experiments is an open and *difficult* problem, and future research should be conducted to identify these measures. Security research often operates in controlled environments to limit disruption of real-world settings. An avenue for future meta-science research is quantifying the variability from lab-controlled. Finally, we see potential in identifying avenues for automating replicability analysis and studies. This approach is made easier by testbeds (e.g., SPHERE [2]), but broader implications exist within how these processes generalize to different environments. For example, one may be interested in testing environments outside the domain of their original experiments. The development of tools to automate this process can help create better mechanisms for transparent research.

8 Conclusion

Reproducibility is a growing concern within the Security community and as such, conferences and authors are starting to address this. Yet reproducibility is only a small part of the broader field of validity. Replicability is the testing of the underlying hypothesis. In this work, we provide a systematization of the field of computational reproducibility and replicability. We then provide the first framework to unify definitions and address the limitations in prior work, specifically in how replicability is ill-defined. Our framework is flexible and conceptually simple, built around a binary tree of all of the available experimental processes, the Tree of Validity. This formulation allows consistent communication of experimental methodologies and comparisons of reproducibility and replicability studies, which we demonstrate through several case studies of known attacks, defenses, award-winning artifacts, and systematizations of knowledge from USENIX Security. We encourage future authors to adopt our framework in their research to promote open science goals.

9 Ethics Consideration

We did not make any ethical considerations for our work, but there are ethical ramifications of our work. Security research often occurs in sensitive areas, such as privacy, surveillance, and data protection. Replicating a study can inherently affect the privacy and security of the studied area. For instance, a sensitive dataset may be kept from the public, making a reproducibility study impossible.

Our proposed framework aims to help researchers navigate these challenges by offering a structured way to describe the claims and evidence in their studies, even when full disclosure is ethically or legally constrained. Thus, our framework enables comparison without requiring sensitive components to be made public, the Tree of Validity (ToV) encourages ethical transparency. Further, this framework can help prompt reflection on the ethical boundaries of replication, encouraging authors to document what was done.

10 Open Science

We provide all of the ToVs produced as part of this paper. Experimentally, we conducted one main experiment of replicability (e.g., analyzing cookie traffic). We will provide the code to conduct this with the submission, but note that it primarily relies on previous artifacts of submission. To calculate our end results we provide a script to run the collection and generate Figure 8 at https://zenodo.org/records/15616973.

11 Acknowledgements

This work was supported in part by the National Science Foundation grants CNS-2446321 and CNS-2206950. Any findings and opinions expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies. Finally, the authors would like to thank the members of the Artifact Evaluation Committee for their efforts in improving our open-source artifact.

References

[1] PLDI: Artifact Evaluation Committee. https://pldi15.sigplan.org/.

[2] Security and Privacy Heterogeneous Environment for Reproducible Experimentation. https://sphere-project.net/.

[3] ACSAC Call for Artifacts. https://www.acsac.org/2017/artifacts/, 2017.

[4] WiSec Artifactiation Evaluation Committee 2017. https://wisec2017.ccs.neu.edu/reproducibility.html, 2017.

[5] USENIX Security '20 Call for Artifacts. https://www.usenix.org/conference/usenixsecurity20/call-for-artifacts, 2020.

[6] USENIX Security '25 Call for Artifacts. https://www.usenix.org/conference/usenixsecurity25/call-for-artifacts, 2021.

[7] Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium. https://www.usenix.org/sites/default/files/sec22_full_artifact_proceedings.pdf, 2022.

[8] USENIX Security '22 Artifact Appendix. https://www.usenix.org/conference/usenixsecurity22/artifact-appendix-guidelines, 2022.

[9] ACM CCS Call for Artifacts. https://www.sigsac.org/ccs/CCS2023/call-for-artifacts.html, 2023.

[10] NDSS Symposium 2024 Call for Artifacts. https://www.ndss-symposium.org/ndss2024/submissions/artifacts/, 2024.

[11] Bluetooth device shipments worldwide from 2015 to 2018. https://www.statista.com/statistics/1220933/global-bluetooth-device-shipment-forecast/, 2025.

[12] IEEE S&P Call for Papers. https://sp2026.ieee-security.org/cfpapers.html, 2026.

[13] Davide Balzarotti and Wenyuan Xu. Message from the USENIX Security'24 program co-chairs. In *33rd USENIX Security Symposium, USENIX Security 2024*, 2024.

[14] Lorena A Barba. Terminologies for Reproducible Research. *arXiv preprint arXiv:1802.03311*, 2018.

[15] Jennifer Urbano Blackford. Leveraging Statistical Methods to Improve Validity and Reproducibility of Research Findings. *JAMA Psychiatry*, 74(2):119–120, 2017.

[16] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. Automating Cookie Consent and GDPR Violation Detection. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2893–2910, 2022.

[17] Denny Borsboom, Gideon J Mellenbergh, and Jaap Van Heerden. The Concept of Validity. *Psychological review*, 111(4):1061, 2004.

[18] Jon F Claerbout and Martin Karrenbach. Electronic Documents Give Reproducible Research a New Meaning. In *SEG Technical Program Expanded Abstracts 1992*, pages 601–604. Society of Exploration Geophysicists, 1992.

[19] Chris Drummond. Replicability is Not Reproducibility: Nor is it Good Science. In *Proceedings of the Evaluation Methods for Machine Learning Workshop at the 26th ICML*, volume 1. National Research Council of Canada Montreal, Canada, 2009.

[20] Association for Computing Machinery. Artifact review and badging - Version 1.0 (not current), 2016.

[21] Association for Computing Machinery. Artifact review and badging - current, Aug 2020.

[22] Omar S Gómez, Natalia Juristo, and Sira Vegas. Understanding Replication of Experiments in Software Engineering: A Classification. *Information and Software Technology*, 56(8):1033–1048, 2014.

[23] Steven N Goodman, Daniele Fanelli, and John PA Ioannidis. What does research reproducibility mean? *Science translational medicine*, 8(341):341ps12–341ps12, 2016.

[24] Google. Overview of CrUX. https://developer.chrome.com/docs/crux.

[25] Thomas Gregory, Ulrich Hansen, Monica Khanna, Celine Mutchler, Saik Urien, Andrew A Amis, Bernard Augereau, and Roger Emery. A CT Scan Protocol for the Detection of Radiographic Loosening of the Glenoid Component After Total Shoulder Arthroplasty, 2014.

[26] Odd Erik Gundersen. The Fundamental Principles of Reproducibility. *Philosophical Transactions of the Royal Society A*, 379(2197):20200210, 2021.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin Butler. FIRMWIRE: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Network and Distributed Systems Security Symposium (NDSS) 2022*, 2022.

[29] Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled Watermarks as a Defense Against Model Extraction. In *30th USENIX security symposium (USENIX Security 21)*, pages 1937–1954, 2021.

[30] Ivo Jimenez, Michael Sevilla, Noah Watkins, Carlos Maltzahn, Jay Lofstead, Kathryn Mohror, Andrea Arpaci-Dusseau, and Remzi Arpaci-Dusseau. Standing on the Shoulders of Giants by Managing Scientific Experiments Like Software. *USENIX*, 41(4):20–26, 2016.

[31] Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom. Spectre Attacks: Exploiting Speculative Execution. 2019.

[32] Shriram Krishnamurthi. Artifact Evaluation for Software Conferences. https://cs.brown.edu/~sk/Memos/Conference-Artifact-Evaluation/esec-fse-2011.html.

[33] Alex Krizhevsky, Geoffrey Hinton, et al. Learning Multiple Layers of Features from Tiny Images. 2009.

[34] Seth Layton, Tyler Tucker, Daniel Olszewski, Kevin Warren, Kevin Butler, and Patrick Traynor. SoK: The Good, The Bad, and The Unbalanced: Measuring Structural Limitations of Deepfake Media Datasets. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1027–1044, 2024.

[35] Yann LeCun. The MNIST database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[37] Mark Liberman. Replicability vs. Reproducibility - Or is it the other way around? *Language Log*, 2015.

[38] National Academies of Sciences Engineering and Medicine and others. *Reproducibility and Replicability in Science*. National Academies Press, 2019.

[39] Lily Hay Newman. Sloppy Software Patches Are a 'Disturbing Trend'. https://www.wired.com/story/software-patch-flaw-uptick-zdi/.

[40] Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. "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*, pages 3433–3459, 2023.

[41] OneTrust. Cookiepedia. https://cookiepedia.co.uk/.

[42] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR Corpus Based on Public Domain Audio Books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[43] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768*, 2018.

[44] Roger D Peng. Reproducible Research in Computational Science. *Science*, 334(6060):1226–1227, 2011.

[45] Hans E Plesser. Reproducibility vs. Replicability: A Brief History of a Confused Terminology. *Frontiers in neuroinformatics*, 11:76, 2018.

[46] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. *arXiv preprint arXiv:1806.01156*, 2018.

[47] Niels Provos et al. A Virtual Honeypot Framework. In *USENIX Security Symposium*, volume 173, pages 1–14, 2004.

[48] Edward Raff. A Step Toward Quantifying Independently Reproducible Machine Learning Research. *Advances in Neural Information Processing Systems*, 32, 2019.

[49] Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin RB Butler. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 339–356. IEEE, 2016.

[50] Stefan Schmidt. Shall we really do it again? The Powerful Concept of Replication is Neglected in the Social Sciences. *Review of general psychology*, 13(2):90–100, 2009.

[51] Matthias Schwab, N Karrenbach, and Jon Claerbout. Making Scientific Computations Reproducible. *Computing in Science & Engineering*, 2(6):61–67, 2000.

[52] Aleksei Stafeev and Giancarlo Pellegrino. SoK: State of the Krawlers-Evaluating the Effectiveness of Crawling Algorithms for Web Security Measurements. 2024.

[53] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-End Anti-Spoofing with Rawnet2. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6369–6373. IEEE, 2021.

[54] Hemlata Tak, Massimiliano Todisco, Xin Wang, Jeeweon Jung, Junichi Yamagishi, and Nicholas Evans. Automatic Speaker Verification Spoofing and Deepfake Detection Using wav2vec 2.0 and Data Augmentation. *arXiv preprint arXiv:2202.12233*, 2022.

[55] Dave Jing Tian, Grant Hernandez, Joseph I Choi, Vanessa Frost, Christie Raules, Patrick Traynor, Hayawardh Vijayakumar, Lee Harrison, Amir Rahmati, Michael Grace, et al. {ATtention} Spanned: Comprehensive Vulnerability Analysis of {AT} Commands within the Android Ecosystem. In *27th USENIX security symposium (USENIX security 18)*, pages 273–290, 2018.

[56] Pete Warden. Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[57] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[58] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. ASVspoof 2021: Accelerating Progress in Spoofed and Deepfake Speech Detection. In *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Coutermeasures Challenge*, 2021.

[59] Zhiyuan Yu, Yuanhaur Chang, Shixuan Zhai, Nicholas Deily, Tao Ju, XiaoFeng Wang, Uday Jammalamadaka, and Ning Zhang. XCheck: Verifying Integrity of 3D Printed Patient-Specific Devices via Computing Tomography. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2815–2832, 2023.

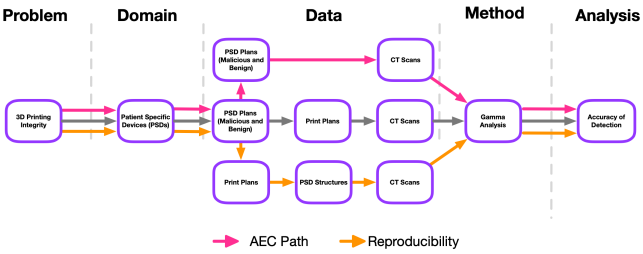

Figure 9: This figure shows the SoV for XCHECK by Yu et al. [59] and the execution path the AEC took to reproduce the results.

A Appendix

A.1 Additional Case Studies

A.1.1 XCHECK

Yu et al. [59] proposed XCHECK which verifies the integrity of the 3-D printed patient-specific devices (PSD)s. They verify the integrity of the PSDs using various methods (e.g., gamma analysis). In their experimental artifacts, they provide all of the PSD plans and CT scans of the PSDs used in the study. Thus, a validator can run Yu et al.'s gamma analysis on the CT scans and PSDs to obtain their experimental results, but this is not full reproducibility. As seen in Figure 9, a validator would need to walk an execution through the ToV: using the PSD plans, print the PSDs; then scan the PSDs using a CT scanner, which results in usable CT scans; then the validator can apply the gamma analysis to identify malicious attacks. XCHECK was awarded a Results - Reproduced badge, but the execution path does not use the same collection methodology. Instead, it relies on the prior conducted data collection pathway. Thus, the execution of a validity experiment for the AEC starts at an internal node of the ToV.

The operation of the data collection methodology may affect the final results. This is confirmed in Yu et al.'s discussion section. For example, prior work [25] shows that there are optimal object orientations for CT scanning. Although XCHECK is cost-efficient compared to the cost of printing organs (e.g., millions of dollars), validating the full experimental process would be expensive for a validator that does not have access to the equipment. The validator would need to buy or rent a 3D printer and a CT scanner to run all of the associated experiments. Specialized testbeds (e.g., SPHERE [2]) do not currently have this expensive hardware.

Takeaway *The Security community's understanding of reproducibility is sometimes only a partial execution of a reproducibility pathway starting at an internal node of the ToV.*

A.2 Systematizations

Section A.1 demonstrated how Trees of Validity are built from existing papers and their experimental artifacts. To show how our framework allows for comparison between replicability studies, we rely on SoKs that conducted several experiments around a central problem. USENIX Security introduced SoKs in 2024 [13] and published eight SoKs. We selected two SoKs that demonstrate replicating several . First, we consider deepfake datasets from Layton et al. [34] that iteratively run experiments to identify problems with the construction of deepfake datasets. Second, we demonstrate our framework on Stafeev et al. [52], where they systematize web crawlers. In their experiments, they run the largest experimental evaluation of web crawlers. Both of these papers demonstrate several executions through a tree of validity.

A.2.1 Deepfake Datasets

Layton et al. [34] conduct several experiments to identify existing problems in deepfake datasets. Although their systematization focuses on deepfakes in general, their experiments are targeted at audio deepfake detection. They present three research questions around audio deepfake datasets that are answered through experiments. First, are models built to detect audio deepfakes reproducible? Second, are the metrics representing the performance behavior of the models sufficient? Third, how does the current construction of the dataset affect the model performance? The third research question is answered in two parts: (1) by changing the training set and evaluating the performance and (2) by changing the evaluation set to a new domain and evaluating performance. This experimental setup can be modeled by a ToV and shows that each of these research questions is targeted at a subset of the ToV. Within their work, they highlight seven models across five experiments.

For succinctness, we highlight and construct the Tree of Validity from one of the audio deepfake detectors they use in the paper, RawNet2 [53]. Then, we define the executions of the three experiments using one of the other models, wav2vec [54]. There are five other models used in the paper, but these models would appear as the same execution path through the ToV as wav2vec. In Figure 10, we see the executions through the Tree of Validity that map to the research questions presented in their paper. The Problem is detecting audio deepfakes in the Domain of the ASVspoof2021 dataset [58]. As we build the Tree of Validity from the perspective of the RawNet2 model, the Method is the model of detection of the RawNet2 Model. The data it is trained on is the ASVspoof Train set and the ASVspoof Eval set to generate the Analysis metrics of EER, FPR, and TPR.
RQ1 They reproduce the RawNet2 model using the same experimental artifacts and walk the path where all of the experimental methodology remains the same. Thus, this execution walks the path of the upper-most, as visualized in Figure 10.

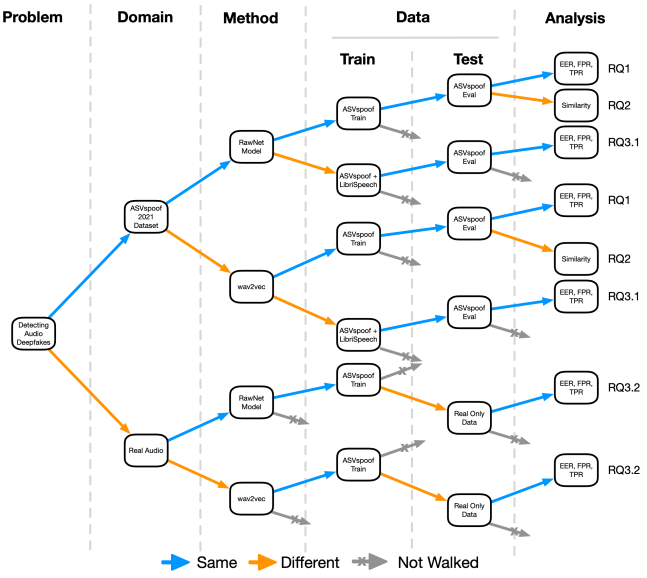

Figure 10: This figure shows the executions of the experiments Layton et al. [34] conduct. We model RawNet2 as the original experiment and wav2vec as the second model. The grey arrows represent sub-trees that are not a part of any execution conducted in the paper.

When reproducing wav2vec, they walk the path where the only difference is the method. The other models that are reproduced would also take the same path as wav2vec.

RQ2 In this research question, they are assessing the performance of the models with a new analysis technique. Layton et al. compute the similarity between the models, shown as the execution from the root node to the leaf nodes of similarity.

RQ3 The third research question tests the limitations within the datasets. As such, the two experiments that show this are (1) changing the training set and (2) changing the test set. To do this, they modify the train set with more real audio from Librispeech [42] for (1). This path manifests at two leaf nodes. The first is with the RawNet2 model on a different train set but the same test and analysis. The second is with wav2vec on a different train set but the same test and analysis. For (2), they show that the models are biased towards predicting deepfakes by only giving the models real audio. This changes the domain with which the experiment is conducted, and the two paths through the different domains are the executions for wav2vec and RawNet2 on the new domains.

We can apply our framework to these experiments and identify how the individual research questions differ experimentally. In previous frameworks identified in Section 3, RQ3.1 and RQ3.2 fall under replicability with no meaningful difference. While these frameworks would label RQ1 as reproducibility, they cannot differentiate between RQ2 and RQ3. In our framework, we visually see that there are differences between the research questions and can define the differences by the path taken from the root node to the desired leaf node.

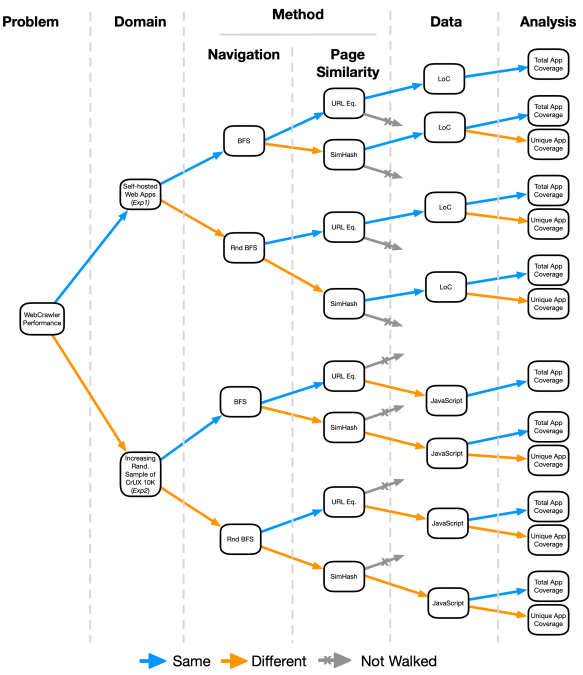

Figure 11: This figure shows the executions of the experiments in Stafeev et al. [52]. We model the baseline navigation and page similarity algorithm configuration (e.g., BFS and URL Eq.) as the Potential Tree of Validity and show the executions of the possible combinations with Rnd BFS and SimHash.

SoKs map a space and provide validity studies. They show where Security research has been conducted. Therefore, by applying our framework to Layton et al., we can denote the differences in the research questions and highlight how the different experimental methodologies affect the replicability of the results.

Takeaway *Our framework provides a clear distinction between the research questions and shows the potential for further replicability studies to improve the understanding of the problem area.*

A.2.2 Web Crawlers

Stafeev et al. [52] compare web crawlers that perform page navigation with a page similarity algorithm. The final tool, *Arachnarium*, implements seventeen-page similarity algorithms and six navigation strategies. There are over 100 possible combinations of similarity algorithms and navigation strategies. To reduce the load on real-world websites, they down-select by performing *Exp1* on self-hosted web applications and taking the ten best page similarity algorithms to perform *Exp2* on an increasingly large random samples from disjoint CrUX Top 10k [24] buckets. While each experiment

could be mapped to this Tree of Validity, we demonstrate the baseline of Breadth-First Search (BFS) and URL Equal (URL Eq.) path navigation as the potential ToV. We show the combinations of BFS and URL eq. with Random BFS (Rnd BFS) and the SimHash navigation algorithm as shown in Figure 11.

Exp1 The first experiment implements the over 100 combinations of the seventeen-page similarity algorithms and the six navigation strategies. In this experiment, the data collected is the lines of code, and the code coverage is considered for the analysis. In the top path through the ToV, we see that the domain is self-hosted web applications, the method is broken up into the page navigation algorithm, BFS, and the page similarity algorithm, URL Eq. The total app coverage is considered for each combination. For every combination, a unique app coverage is calculated as a different metric to compare against.

Exp2 Once the top ten page similarity algorithms are identified, the second experiment runs the web crawlers on the increasingly-large random samples from disjoint CrUX Top 10k buckets. This experiment collects the unique JavaScript lines run by crawling the website and represents the bottom sub-tree of the root node. We note that more algorithms could be implemented within this ToV, but they would implement in the same paths as the *different* pathways.

From this case study, we see the numerous executions Stafeev et al. conducted in this work. Further, the analysis can be augmented to create new executions through the ToV to increase the coverage through the ToV. By proposing the framework in this way, we can reason not only about singular validity studies, but any number of validity studies.

Takeaway *Our framework allows a comparable analysis of multiple validity studies by providing a standardized approach.*