"Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences

Daniel Olszewski
University of Florida
Gainesville, FL, USA
dolszewski@ufl.edu

Allison Lu
University of Florida
Gainesville, FL, USA
allison.lu@ufl.edu

Carson Stillman
University of Florida
Gainesville, FL, USA
carson.stillman@ufl.edu

Kevin Warren
University of Florida
Gainesville, FL, USA
kwarren9413@ufl.edu

Cole Kitroser
University of Florida
Gainesville, FL, USA
colekitroser@ufl.edu

Alejandro Pascual
University of Florida
Gainesville, FL, USA
alejandropascual@ufl.edu

Divyajyoti Ukirde
University of Florida
Gainesville, FL, USA
divyajyotiukirde@ufl.edu

Kevin Butler
University of Florida
Gainesville, FL, USA
butler@ufl.edu

Patrick Traynor
University of Florida
Gainesville, FL, USA
traynor@ufl.edu

ABSTRACT

Reproducibility is crucial to the advancement of science; it strengthens confidence in seemingly contradictory results and expands the boundaries of known discoveries. Computer Security has the natural benefit of creating artifacts that should facilitate computational reproducibility, the ability for others to use someone else's code and data to independently recreate results, in a relatively straightforward fashion. While the Security community has recently increased its attention on reproducibility, an independent and comprehensive measurement of the current state of reproducibility has not been conducted. In this paper, we perform the first such study, targeting reproducible artifacts generated specifically by papers on machine learning security (one of the most popular areas in academic research) published in Tier 1 security conferences over the past ten years (2013-2022). We perform our measurement study of indirect and direct reproducibility over nearly 750 papers, their codebases, and datasets. Our analysis shows that there is no statistically significant difference between the availability of artifacts before and after the introduction of Artifact Evaluation Committees in Tier 1 conferences. However, based on three years of results, artifacts that pass through this process work at a higher rate than those that do not. From our collected findings, we offer data-driven suggestions for improving reproducibility in our community, including five common problems observed in our study. In so doing, we demonstrate that significant progress still needs to be made in computational reproducibility in Computer Security research.

CCS '23, November 26–30, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0050-7/23/11…$15.00
https://doi.org/10.1145/3576915.3623130

CCS CONCEPTS

• **General and reference** → **Measurement**; **Validation**; • **Computing methodologies** → *Machine learning*; *Cross-validation.*

KEYWORDS

reproducibility; machine learning; security; meta-science

ACM Reference Format:
Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. 2023. "Get in Researchers; We're Measuring Reproducibility": A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 27 pages. https://doi.org/10.1145/3576915.3623130

1 INTRODUCTION

Advances in science do not come solely from novel exploratory studies. As pointed out in a recent report by the National Academy of Sciences [443], scientific progress requires confirmatory research to validate and expand the limits of new knowledge. This observation has crucial importance across all branches of science and engineering and makes it clear that closing the gap between initial discovery and widespread adoption of claims or methods requires significant effort to be spent on reproducibility. As a result of prioritizing exploratory over confirmatory studies, diverse fields ranging from medicine [184, 256, 591] and economics [122, 239], to chemistry [16, 219] and psychology [35, 126] have suffered very public and damaging reproducibility crises.

Research in Computer Science, and Computer Security specifically, have a rare advantage in their ability to create reproducible science. Specifically, because much of the work in our community produces computational artifacts as a side-effect of their methodology (e.g., code, data, and figures), the ability to perform confirmational studies should be strictly simpler than fields in which methods are less portable, potentially dangerous, or more expensive (e.g., recreating an experimental pharmaceutical compound and testing it on a large population). An increasing number of papers appear to be making their code available to the broader community as a means of supporting such analyses; however, outside of largely anecdotal evidence, a comprehensive study of the availability of artifacts and the ability of independent researchers to confirm their computational reproducibility has not been conducted in our community. Without such a study, it is unclear if Computer Security is truly creating reproducible and ultimately broadly applicable science, or if it is having a reproducibility crisis of its own.

In order to better characterize the current state of computational reproducibility in Computer Security, we perform an extensive measurement study and make the following contributions:

- **Comprehensive Longitudinal Study:** We perform the largest known longitudinal study of reproducibility in the Security community. We focus on the sub-discipline of machine learning (ML) security as published at Tier 1 security venues over the past 10 years, yielding observations over a total population of nearly 750 papers. We find that 60% of these published papers do not include code to run their experiments. Moreover, 56% of the provided artifacts do not run at all.
- **Measure Impact of Artifact Evaluation Committees:** Using data collected from the longitudinal study, we analyze whether the introduction of artifact evaluation committees in 2020 has had an impact on the availability of code artifacts. We show that there is no statistically significant difference between the two groups, meaning that Artifact Evaluation Committees have yet to achieve their intended goals.
- **Recommendations Based on Measurement:** We highlight five recommendations based on the most common problems that we observe in our analysis that impeded both our indirect and direct reproducibility studies. We believe that explicitly addressing these issues will result in a substantial improvement of reproducibility in Tier 1 Security conferences.

Computational reproducibility efforts are often discounted in their value when compared to exploratory/novelty-focused papers, as the former often requires less time or expense than their exploratory counterparts. We note that conducting this study required extensive resources and time, with an estimated 8 person-years of effort and well over 10,000 hours of computational time to recreate results. Only through such comprehensive measurement and analysis of our community can actionable steps for improvement be offered.

The remainder of the paper is organized as follows: Section 2 provides background information including explicit definitions of reproducibility; Section 3 states our null hypothesis; Section 4 details our research questions and methodology; Section 5 discusses the results and implications of our Indirect Reproducibility study; Section 6 presents the observations and results of our Direct Reproducibility study; Section 7 provides discussion of critical issues and offers recommendations based on our observations; Section 8 presents limitations of our study and future considerations; Section 9 highlights related work from a broad set of communities; and Section 10 provides concluding remarks.

2 BACKGROUND

We briefly discuss the formal study of reproducibility and current artifact evaluation in the Security community.

2.1 Reproducibility

Although *ACM* harmonized its definition of reproducibility in 2020 [1], we use the National Academy of Science's definition [60, 443] for computational reproducibility, replicability, and generalizability and discuss the nuances between each.

Computational Reproducibility: Computational reproducibility (i.e., *reproducibility*) refers to recreating a study's results with the study's artifacts. Thus, reproducibility occurs when an independent team can obtain consistent results using the same data, computational environment, and code [212]. Reproducible results confirm that the phenomenon described by a paper is present under the study's environment.[1] Further, bit-wise reproducibility entails reproducing the exact numeric results. Oftentimes, this strict definition is relaxed, albeit marginally, for areas that rely on complex computational processes or use some modicum of randomness (e.g., machine learning).

A computational reproducibility study can be either *indirect* or *direct*. An indirect study assesses whether the authors of a study made their artifacts available. It considers the transparency of the study and thus requires fewer resources to conduct. A direct study runs, if available, the same code, data, and analytical methodology to check that running the provided code recreates the results of the paper. This inherently requires a greater amount of resources. In this paper, we conduct both a direct and indirect study to understand and measure the current state of computational reproducibility within the Security community.

Replicability: While computational reproducibility uses the same code and data for a study, *replicability* studies seek to address the same research question with different methodologies. Using different collected data, a replicability study aims to confirm the results of a previous study, subject to the inherent uncertainty of the underlying studied system. Due to the statistical nature of observation, a failure to replicate a study does not necessarily mean that the original study's results are false, nor does success indicate that the original study's results are true. Replicability is achieved through numerous attempts that provide a preponderance of evidence for the existence of the observed phenomenon.

Generalizability: All scientific exploration occurs in some unique environment. *Generalizability* defines how the identified trends apply to other unique environments [60]. There are numerous reasons a study may not generalize. An unsuccessful attempt to show generalization may not come from the study but from outside conditions adversely affecting the underlying system. Similarly to replicability, generalizability is not shown by a single study, but by numerous studies across multiple conditions that collectively show the same phenomenon.

2.2 Artifact Evaluation

Conferences seek to address concerns about reproducibility by providing artifact evaluation committees (AECs) and giving authors

[1] It is important to note that errors within the computation are not addressed. Bad code that leads to erroneous claims will propagate throughout a reproducibility study.

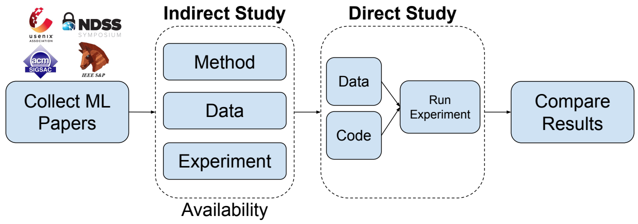

Figure 1: The pipeline for the methodology of our study. We collect machine learning papers from the Tier 1 Security conferences. Then we perform an indirect study of each paper considering the availability of the Method, Data, and Experiment. Finally, from the available Data and Experiments, we run a direct study running their code to attempt to recreate their results.

the option to submit to their call for artifacts. The committee accepts everything including tools, test suites, models, proofs, and even videos of the artifact working to evaluate whether the artifacts are consistent with the claims or procedures of the associated paper [313]. These committees award badges depending on the success of running. *USENIX Security* as of 2023 assigns three badges: *Artifact Available* where some portion of the artifact is publically available; *Artifact Functional* where the artifact runs; and *Artifact Reproduced* where the artifact directly reproduces the results of the paper.

While these committees face many issues [48], AECs are becoming increasingly popular in many communities [678]. In 2017, both *WiSec* and *ACSAC* added artifact evaluations committees after calls from the community [50, 454]. Of the Tier 1 conferences, *USENIX Security* introduced its AEC in 2020, and *CCS* introduced its AEC in 2023. As of the time of writing, *NDSS* and *IEEE S&P* have not introduced AECs. Although outside the scope of our work, it is important to note that *NeurIPS* open-sourced the reproducibility of papers in a reproducibility challenge [485]. However, unlike *NeurIPS*, Tier 1 Security conferences do not require artifact evaluation. As such, this inherently affects the state of reproducibility. We aim to provide an analysis of how AECs have affected reproducibility in Section 6.

3 H_0 NULL HYPOTHESIS

A null hypothesis, along with an alternative hypothesis, conjecture relationships about a population. These hypotheses are tested against a statistical model of collected data to show statistical significance. A null hypothesis claims that there is no causal relationship resulting in differences between two subpopulations and that any observation is due to random chance. The alternative hypothesis is the inverse stating that there is a statistically significant relationship. Although traditional work considers a p-value less than 0.05 as statistically significant, modern experts in the "post $p < 0.05$" era encourage classifying $0.005 < p < 0.05$ as merely "suggestive" and $p < 0.005$ as beginning to indicate statistical significance [49].

We state the null hypothesis for this paper as:

H_0 · There is no difference in whether code from published papers is available before vs after the introduction of AECs to Tier 1 Security Conferences (2020).

4 METHODOLOGY

To understand the current state of reproducibility in the Security community, we conduct a measurement study where we collect machine learning papers from the Tier 1 conferences over the past 10 years. We select this sub-community because it is large and long-lived. Moreover, it requires less specialized equipment than other areas (e.g., wireless) giving us the best opportunity to capture reproducible science. In this section, we outline what criteria a paper must meet to be considered a part of this study and how we analyze each paper according to its Method (i.e., a complete description of its methodology), Data, and Experiment (i.e., code), outlined in Figure 1. We perform both an Indirect and Direct Study of reproducibility. We propose the following research questions to guide our study:

RQ1 (Indirect Study) Do studies provide the details of their method?
RQ2 (Indirect Study) To what extent is collected data made available? Where are studies sourcing their data?
RQ3 (Indirect Study) To what extent are experimental artifacts made available?
RQ4 (Direct Study) Of available experimental artifacts, how many run and produce consistent results?

4.1 Paper Selection

We consider papers from the four Tier 1 Security conferences (*ACM-CCS*, *IEEE S&P*, *NDSS*, and *USENIX Security*) ranging from the years 2013 to 2022 (10 years). We exclude all workshops associated with each conference as well as any poster talks. As we aim to quantify the state of reproducibility in machine learning security, we attempt to select every paper that uses ML in its system design.

To make this process as objective as possible, we select papers according to the following criteria: (1) machine learning is mentioned in the Abstract, Introduction, Background, Methodology, or Conclusion; (2) the paper creates a training procedure based on data available to the study authors, usually mentioned in the Methodology or Results section (e.g., "We train a multi-layer perceptron on our collected data."); (3) their Results section clearly outlines a metric for an ML model that is not from previous work (e.g., "Our RFC classifier achieves 99% accuracy"). We enact a consensus protocol where each year is reviewed by two separate team members. Each reviewer independently applies the selection criteria to find a list of papers, and we take the union of the two lists of papers. Note that our selection process goes beyond AECs and considers every published paper at the conference. After applying our criteria for inclusion in this study, we identify 744 papers. We make all of our data available[2], which includes a list of the papers and all of their associated URLs.

4.2 Indirect Study

After finalizing the list of papers, we evaluate each paper according to its Method, Data, and Experiment. Previous work [50, 212, 496] reinforces that these three factors are the foundation for reproducibility analysis. We outline in detail the factors and each variable in Table 1 and discuss them in the following sections. Each paper is reviewed twice in the Indirect Study. The Indirect Study only

[2]https://github.com/reproducibility-sec/reproducibility

Factor	*Variable*	*Description*
Method	Model	What model did it use?
	Set Up	Were the hyper-parameters described?
	Training	Does it explicitly outline its training?
Data	Available	Is the data made available?
	Reason	If not, why is it not provided?
	Data Split	Is the data split into training/validation/test in a deterministic way?
Experiment	Available	Was the code made available?
	Instructions	Does the code have explicit instructions on how to run?
	Trained	Is there a trained model? Is there training code?
	Works	Does the code work based on the instructions? Was further coding needed?
	Output	What is the output of the code? Does it match the metrics in the paper?
	Results	Is the code output consistent with the claims of the paper?

Table 1: An outline of the factors, variables, and details we use to analyze each paper. The three factors are Method, Data, and Experiments. Each variable represents a column in our data frame and forms the foundation for our study.

assesses the availability of the identified factors. We do not run any code during the Indirect Study. A measure for inter-rater reliability is Cohen's Kappa coefficient, which balances the agreement between two raters against the random chance that they would agree. Generally, a Cohen's Kappa coefficient above 0.7 is considered acceptable for inter-rater reliability, and our resulting Cohen's Kappa coefficient is $\kappa = 0.83$.

Method: To understand how well a paper describes its methodology, we recreate the experimental procedure to replicate the results. We aim to measure the presence of model descriptions as it is essential to both reproducibility and replicability studies. This constitutes the first part of our indirect reproducibility study. When we evaluate a paper on its Method, we consider three factors. First, we look at what model is used. If the paper does not outline its analysis model, we fundamentally cannot recreate its results. Second, we consider if all of the hyperparameters are described. Many ML models consist of numerous parameters (e.g., number of nodes, activation functions, loss functions) required for similar performance. We consider this on a numeric scale of no description (0), partial description (1), and complete description(2) of hyperparameters. Finally, we want to understand the paper's training procedure. A lack of a training procedure further inhibits any reproduction of its results. The training procedure is also measured on a scale similar to the hyperparameters.

Data: For the second part of our indirect reproducibility study, we evaluate the data on whether it is publically available and if it is split into train, validation, and test sets in an explicit way. Studies often use multiple datasets which can be either publically available or private. We consider that there are reasons why the data is not released and note that in the Reason variable. The data being available does not immediately mean the results are reproducible. Thus, we want to identify when a paper ensures that the data is used correctly.

Experiment: In machine learning papers, the experiment is often a computational procedure that is run through code. While running the code, a model is trained on processed data, evaluated on test data, and outputs a metric for performance. The previous two factors we study are important and affect computational reproducibility, but the majority of our analysis comes from evaluating the experimental artifacts. For our indirect study, we assess how often code is made available for a paper. We consider not only if the authors link a repository in the paper, but also if we can locate the artifacts online. We do this by searching the paper's title in a search engine and crawling the authors' websites.

4.3 Direct Study

The indirect study evaluates the availability of analytical methodology, data, and experimental artifacts. In contrast, the direct study aims to evaluate the efficacy of the available artifacts for recreating the results. Not only do we evaluate if the artifact is available, but we also evaluate the instructions to run the artifact, whether there is a trained model or training code available, if the artifact runs, if the output of the artifact is the metric in the paper, and that the results are consistent with the paper's claims. When artifacts are available via an online repository, we download the repository to our servers. Following the ReadMe we try to run the code, then in cases where we are required to request access, we do so.

If we are unable to immediately run the code based on the ReadMe, we spend at most one-hour debugging or setting up the project. This is similar in a time scale to Collberg et al.'s [127] methodology, which only evaluated whether artifacts compiled. If we are unable to run the artifact code after an hour of setup and debugging, we mark it as not working. Some projects provide the code to train their model but not the model they previously trained. Due to the scope of our study, we disregard the recommended training time as some require months of computational training time. We train the model for 10 hours and then perform its evaluation procedure, if available. We recognize this as a limitation and further discuss it in Section 8.2. After evaluation, we consider a result reproduced if we get within 5% of the claimed metric, similar to Raff [495]. Raff's study replicated the experiments by the paper's described methodology and create their own code to do so. They do not rely on the artifacts of the paper. As such, 5% is more generous in our reproducibility study, since we are running the authors' code.

Some artifacts require special architectures to run (e.g., GPUs). We run every model on a CPU unless a special architecture is specified by the authors. The results of the direct study for each artifact are reviewed by another reviewer. We run the experimental artifacts once unless, when the artifacts did not run, the second reviewer identifies a possible workaround. We accept the most positive result for the artifact. Following this methodology, we

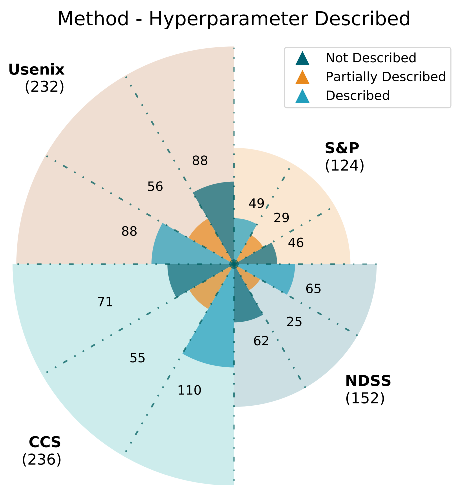

Figure 2: A modified Coxcomb plot that summarizes the indirect study of hyperparameters. A Coxcomb is a bar graph in polar coordinates. Thus, the radius depicts the count of papers in a category. For example, we find that of the 232 papers we consider at *USENIX* 88 describe their hyperparameters, 56 partially describe them, and 88 do not describe them.

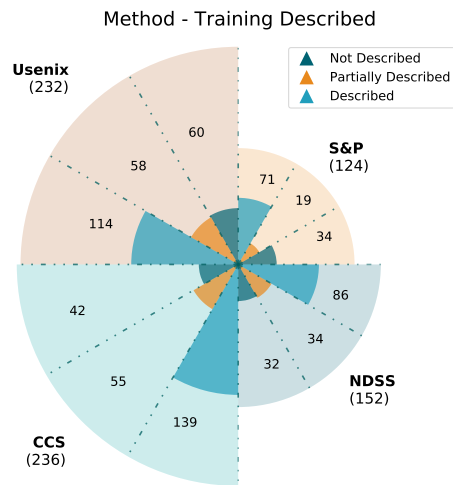

Figure 3: A modified Coxcomb to summarize the indirect study of training methodologies. We find that 77% of the papers we consider describe or partially describe their training procedures. 168 papers (23%) in our study did not describe their training procedure.

analyze over 298 code repositories constituting *over 8 person-years* worth of work and over 10,000 hours of computation time.

5 INDIRECT STUDY

We discuss the indirect study by analyzing the availability of the Method, Data, and Experiment. Each subsection outlines the importance of the factor, the results from our Indirect Study, a case study that includes examples of the presence of the factor, and lessons of the results.

5.1 Method

The method of a study is the fundamental process of evaluating a study. Accurate and complete descriptions of how the analytical methodology is conducted aid reviewers and readers in understanding the study. For recreating the machine learning security papers, we are primarily interested in two aspects, the hyperparameters of their model and the associated training mechanisms. The hyperparameters dictate the details of their model (e.g., the number of layers in a deep neural network). Changing any hyperparameter alters the underlying algorithm, and thus, inhibits reproducibility. The training mechanism outlines how to run the underlying algorithm (e.g., specifying the number of epochs to train for). Changes in how models are trained can result in different models. We measure both of these attributes on a scale of Not Described, Partially Described, and Described.

We find that of the 744 papers we look at, 312 papers fully described their hyperparameters, 165 partially described their hyperparameters, and 267 did not describe their hyperparameters. Figure 2 shows our results for the hyperparameters. We see that in *USENIX*, *S&P*, and *NDSS* there is an even split between not described and fully described at approximately 38% each, with the remaining going to the partially described. However, at *CCS* the number of Described is double that of the Not Described. Figure 3 shows the results of analyzing the training mechanisms. We find that of the 744 we look at, 410 papers described their training, 166 partially described their training, and 168 did not describe their training (**RQ1**).

Case Study: There are many ways to thoroughly explain hyperparameters and training procedures. Oesch et al.'s *That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers* [455] contains a detailed table in the appendix that lists all of the hyperparameters for their model. This is a succinct way to outline all associated hyperparameters.

Training procedures do not require extensive discussions, but it is important to list the associated parameters. Chen et al.'s *On Training Robust PDF Malware Classifiers* [114] outlines how they train their neural networks by listing the number of epochs, the batch size, the optimizer, and the learning rate. When using cross-validation, it is important to discuss the number of cross-validation folds and how they are chosen. Siby et al.'s *Encrypted DNS ⇒ Privacy? A Traffic Analysis Perspective* [566] uses 10-fold cross-validation describing exactly how they split the dataset to validate their model. We label papers as a partial description if they lack a stopping procedure, the number of epochs, or underdefined a procedure (e.g., "we performed cross-validation" instead of "we performed 10-fold cross-validation").

Lesson: While there is a considerable amount of work that provides an adequate discussion of their hyperparameters and training procedures, we find that there are improvements to be made. The papers with the best discussion of hyperparameters include a table with the full model hyperaparameters, as well as outlining how they

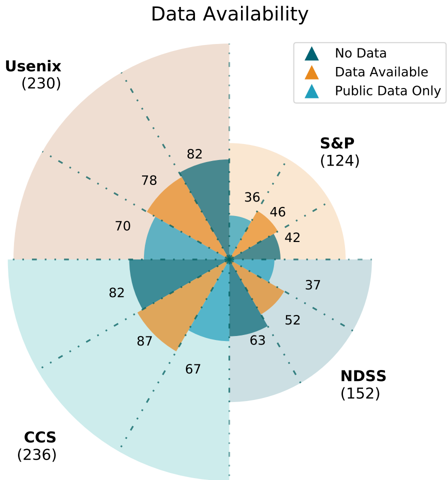

Figure 4: A summary of the data availability found in our indirect study. Approximately one out of every three studies collect data but do not make it available. 35% collect data and make it available. Every conference demonstrates the same trend where there are approximately an equal number of papers that make data available, do not make data available, and use publicly available data.

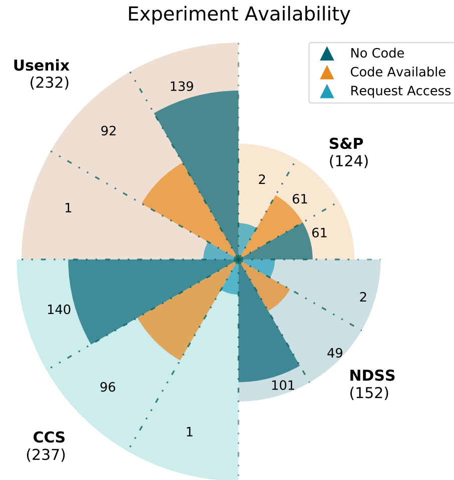

Figure 5: A representation of the papers within our study that made artifacts available. Only 298 of the 744 papers (40%) provide artifacts. About 1% of the total papers remain as request access. The remaining 59% (441 of 744) of papers did not provide any artifacts with their study.

choose the parameters (e.g., grid-search). Similarly, the best discussions on training procedures include an explicit training paragraph or section that explicitly details the training algorithm.

5.2 Data

Machine learning works by training on data, and thus data availability directly affects the reproducibility of a study. We categorize the data availability broadly into three categories: No Data Available is assigned when the paper collected data and did not make it available or when there is no avenue for accessing the data (e.g., a broken link); Data Available is used when a study collects their own data and makes it publically available; Public Data Only is when a study only uses previously published and accessible datasets.

Figure 4 summarizes the results of our Indirect Study. We find that approximately 36% of studies collected their own data and did not make this publically available, 36% collected data and made it available, and 28% of studies conducted their experiments on publically available data (**RQ2**). We see similar trends across *USENIX*, *CCS*, and *S&P* where the Data Available is larger than the No Data provided. However, *NDSS* has more No Data provided than Data Available. When the data was not available, only 10% of papers gave a reason with 7% working with sensitive data (e.g., personally identifiable information) and 3% working with proprietary data (e.g., collected malware intrusions). Approximately 90% of papers (243 of 269) that did not make their data available did not indicate as to why it was not available.

Case Study: Datasets require significant work and funding to create. Zheng et al.'s *Characterizing and Detecting Non-Consensual Photo Sharing on Social Networks* [768] created a dataset that depicts unaware people in a photo. This dataset contains 6,437 photos that are labeled by three users from a user study. The dataset is available online to promote future work in the area. Das et al.'s *The Web's Sixth Sense: A Study of Scripts Accessing Smartphone Sensors* [137] crawled 3,695 websites to detect when a website accessed device sensor data on mobile devices. They collect and provide data in both the United States and Europe. The online repository contains the javascripts that they found, as well as the features of each script, the assigned cluster, and aggregations across the various sensors. Further, in the repository's ReadMe, they connect each file to the methodology that created it in their paper (e.g., "using the methodology described in Section 5"). Although these two papers provide complete datasets, other studies only provide a small subset of their data. This helps to understand the data collected, but ultimately cannot lead to reproducibility. Finally, a different study cited a dataset that has multiple versions without specifying which one, while another study released its raw data but not its processing script or how it chose a train/test split.

We find that about 28% of the studies used publically available resources. The datasets range from static datasets such as CIFAR10 [314] and MNIST [326] to databases that are continuously updated like the OpenSky Database [526]. Using publically available datasets allows for benchmarking and direct comparison between systems.

Lesson: While we recognize that not every dataset can be made available, 36% of collected data remains inaccessible. This creates problems for future research such as a lack of benchmarks when trying to compare the same data, slowing the growth of future research on similar problems, and lack of validation of the dataset. Data collected to support one's argument should be made fully available, when possible. To the best of their ability, authors should make their data available including both processed and unprocessed versions, and when unable to, discuss why they cannot.

5.3 Experiment

The last part of the Indirect Study is assessing to what extent studies make their experiment (i.e., code) available. The results can be seen in Figure 5. We find that approximately 60% of papers (446 out of 744) did not provide code to run their experiments, 39% (298 out of 744) provided code, and approximately 1% still remain as request access.[3] *USENIX*, *CCS*, and *NDSS* contain approximately 1.5 times more papers without code than with code. There is an equal split between no code and code for *S&P* (**RQ3**). Interestingly, we find approximately 1% of papers state that their code will be available in the future or link an empty repository in their paper (even years after their publication).

Case Study: Of available artifacts, 95% are available via GitHub. 4% are websites hosted Google Sites or Google Drive links, and the remaining 1% are hosted at universities. Often the link is in a footnote in the paper. Some treated their repository as a citation, and the link is in the references section. We find that if the citation is ambiguous we often did not find it on the first pass (e.g., "We get 90% accuracy[0]." and the reference says "[0] - ToolName. *link*."). Pasquini et al.'s *Improving Password Guessing via Representation Learning* [477] cite the artifact, but it is explicitly in a section labeled "Availability". While generally if there are artifacts that exist but are not linked in the paper, we could find them via the authors' websites or searching sites like GitHub. However, for one outlier case, there was no code linked in the paper or the authors' websites, but we found a link to the artifact website in a Twitter thread.

Lesson: Not providing the experimental code limits the extent that future work can improve on new tasks and compare against the technique. Further, complex pre-processing tasks, novel analysis techniques, and complicated system designs are non-trivial to build from scratch. Providing implementations not only improves the state of reproducibility but allows further development of these techniques. We further discuss the state of the available experiments in Section 6.

6 DIRECT STUDY

A direct reproducibility study seeks to understand if the available experimental setup will allow us to recreate the results of the work. Building upon the indirect study, we take the papers that made its artifacts available and attempt to run them. In this section, we summarize the results of our direct study on the 298 papers that have available experimental artifacts.

6.1 Results

We find that we are unable to get 56% of the artifacts to run. Although the results show that 44% of repositories run, this does not depict the full story. With only 20% of the repositories running with the same results, the remaining 22% either recreate different results (4%) or execute but are missing arguments or outputs (18%). We can see this relationship in Figure 6 where the results are grouped by conferences. *USENIX* had 92 papers with code where 53 did not run, 21 recreated the claimed results, 3 did not create the same results, and 16 were missing parameters, files, or outputs. *CCS* contained

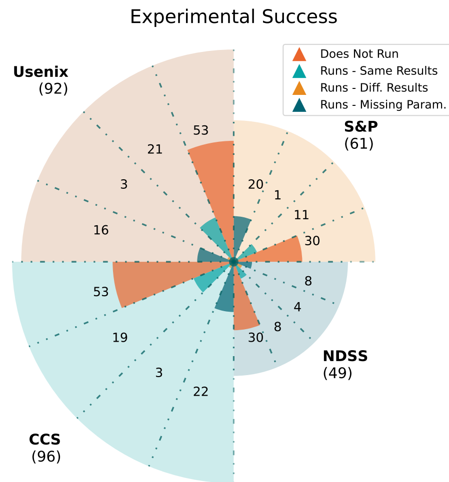

Figure 6: The number of experimental successes for the papers in experiment availability which their artifacts available. Only 20% of the available artifacts run and recreate the results claimed in the paper. 53 of the 92 papers (58%) at *USENIX*, 30 of the 61 papers (49%) at *S&P*, 53 of the 96 papers (55%) at *CCS*, and 30 of the 49 papers (61%) at *NDSS* did not run. Code not running is the dominant occurrence amongst papers that made artifacts available, which is a common trend across all conferences.

96 with code where 53 did not run, 19 recreated the results, 3 did not create the same results, and 22 were missing parameters. *S&P* had 61 papers with artifacts where 30 did not run, 11 recreated the same results, 1 did not create the same results, and 20 were missing parameters. *NDSS* had 49 papers where 30 did not run, 8 recreated the same results, 4 did not produce the same results, and 8 were missing parameters. A commonality across all conferences is that *over half* of all papers that make code available do not run. We observe artifacts missing parameters from a variety of areas such as output, arguments to run the commands, or data not being included(**RQ4**).

There are many factors that affect the running of an artifact including the clarity of the instructions, what is available in the repository, and what the code outputs. Figure 7 visualizes five factors that we noticed while running the repositories. Specifically, we look at the clarity of the ReadMe, whether the code works out-of-box, the output of the code, the train-test splitting, and if they included a trained model.

ReadMe: The ReadMe is the foundation for experimental artifacts. They inform about the purpose of the artifact, how to set up the repositories, what commands to run, and changes that can be made. 57% of the artifacts possess a ReadMe that offers instructions and necessary environments. Some artifacts provide directions for reproducing its results. For instance, the ReadMe for Mehnaz et al.'s *Are Your Sensitive Attributes Private? Novel Model Inversion Attribute Inference Attacks on Classification Models* [408] contains a section dedicated to reproducibility, and following the ReadMe reproduces their results. 30% of the repositories lack concise instructions or

[3]Papers remain as request access until we acquire access to their experiments. Then, we change the paper to a code available paper. Note some request access have been waiting for months.

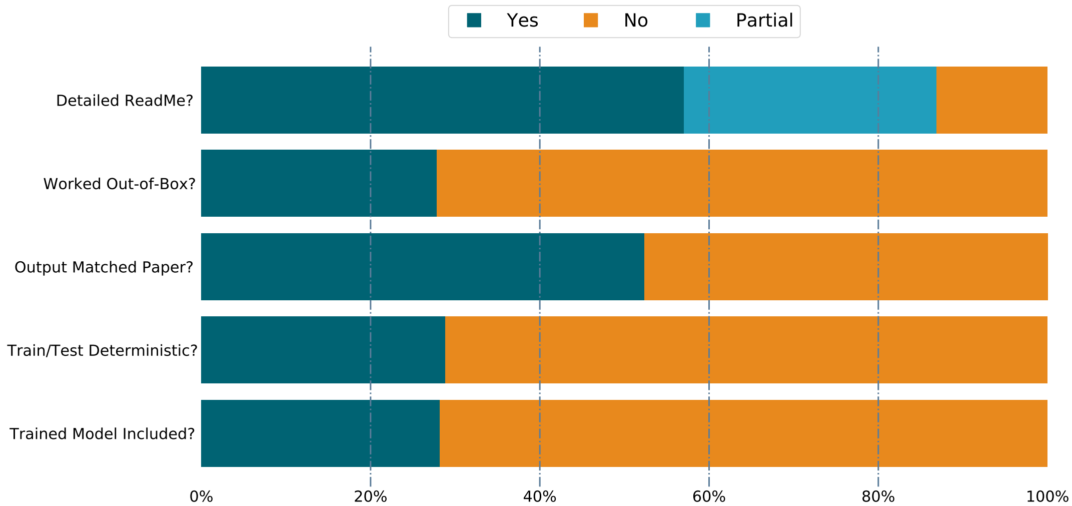

Figure 7: Visualization of the five factors we notices while running repositories: the quality of the ReadMe, whether the code works out-of-box, if the output matches the study's claims, whether the train/test splitting is deterministic, and if they included a trained model. We notice that outside the Readme, the other four factors resulted in the majority of papers failing to meet their requirements.

dependency descriptions. One problem we encounter is that some artifacts require specific package versions but do not specify which one. Others require a preprocessing step that is either not specified or made available. While 87% of the ReadMe's contain some information, we find that 13% contain nothing in the ReadMe besides a title. In these cases, we try to run any file that could lead to reproducibility (e.g., `eval_results.py`), but often we cannot run the artifact. In one example, the artifact contains 70+ files with no instructions on how to run. Further, when we opened `main.py`, the file is commented out.

Out-of-Box: While 42% of the artifacts run, most do not run immediately. Only 28% of the available code repositories work by either following the instructions or, in the absence of instructions, by running the main file. When it did not run immediately, sometimes we could get it to work by installing further packages, changing paths or directory structures, or fixing errors that appear. Repositories that immediately work often limit the number of commands required to run, provide a setup script, or provide a Docker image or virtual machine.

Output: When we gauge the reproducibility of a paper, we try to compare the output of an artifact with the study's claim, yet only 43% of the available repositories match its output to the claims made in the associated paper. In the instances where the output does not match, the code often analyzes small examples, demonstrates the system, or is a library. While the artifacts provide code, they often do not provide meaningful output. For example, one paper claims a performance boost in its system design, but the experimental code outputs "`DONE!`". It did not generate any files or any other output for further analysis. Further, correcting the output in most of these repositories is a non-trivial task requiring an expert understanding of the codebase, naming schema, and techniques applied.

Train/Test: As machine learning models learn from a training set, an artifact should use the same training set. We find that 22% of the artifacts determine their train, validation, and test sets in a deterministic way. Artifacts where the code specifically delineates the train and test sets usually place them in separate directories or the data contains a column that denotes which set the sample belongs to. Without clear separation of the train, validation, and test sets we are unable to accurately reproduce their results, though sometimes we can get close. Further, data availability affects our ability to reproduce results. If they do not include their complete data, scripts to process the data, or scripts to collect the data, we will not be to reproduce their results. For example, while one repository contains a detailed ReadMe with well-marked instructions for every file and how to reproduce their results, there is no collected data and the data collection scripts require access to a $1,000 oscilloscope.

Trained Model: Most machine learning algorithms are stochastic as they seek to find an optimal solution to a problem and thus add an element of randomization [205]. This directly affects an algorithm's reproducibility and can be simply alleviated by providing a trained model. While 17% of artifacts include a trained model, some papers provide both a trained model as well as code to re-train their model. For instance, Bollinger et al.'s *Automating Cookie Consent and GDPR Violation Detection* [61] provides extensive documentation on how to train their model or run the results with one of the trained models in the repository. We could reproduce their results in less than 10 minutes of work.

6.2 Statistical Analysis

In Section 3, we discussed our null hypothesis: There is no difference in whether code is available before and after the introduction of AECs to Tier 1 Security Conferences (2020). In this section, we conduct a statistical test to either accept or reject H_0.

Permutation Testing: We use the non-parametric statistical test, permutation testing because it does not require assumptions on the underlying distribution of the data. The test works by simulating multiple permutations of the data across the two groups, calculating the average distribution within the simulated permutations, taking the difference between the two groups' averages, and then calculating the proportion of samples with a higher difference than the true sample. We take the proportion of papers that make their code available against the papers that do not in a given year. We then separate the years into two groups, before 2020 ($year < 2020$) and after ($year \geq 2020$). We simulate 10,000 permutation distributions and calculate a $p = 0.068$, thus we accept the null hypothesis, H_0. While traditionally some scientists may have tried to argue that this is highly suggestive of significance, a modern interpretation requires a p-value significantly less than 0.05 to imply even a weak relationship. The p-value is close to indicating that there is a suggestive relationship between introducing the AECs and the availability of code, but it does not nearly meet the threshold to do so. In accepting the null hypothesis, we must therefore conclude there is no statistically significant difference between code artifacts produced before and after AECs were introduced to Tier 1 Security conferences.

While we do accept the null hypothesis, we recognize that the introduction of AECs was only three years ago. As such, we look at the quality of artifacts at *USENIX* from 2020 through 2022, as seen in Figure 8. There are not enough samples to test for statistical significance within the *USENIX* AEC; however, anecdotally, we notice that papers submitted to the AEC have a higher likelihood of working. Based on these observations, we believe that further inclusion of AECs in Tier 1 conferences may further increase not only the availability of artifacts but also artifacts that reproduce results claimed in the associated paper. So while it has yet to fully demonstrate its desired impact, gathering more data points on the AEC across multiple conferences will allow a better evaluation of the impact the experiment has had on computational reproducibility in the Security community.

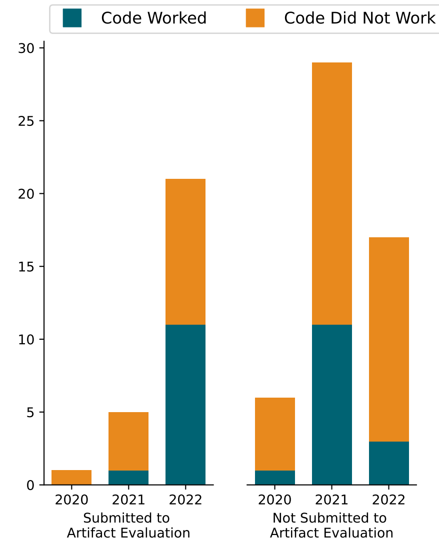

Figure 8: An overview of the impact the AEC has made on artifact availability and code running for papers submitted to *USENIX* over the past three years. We see an increase not only in the number of papers submitting artifacts but also a notable increase in the total number of papers that actually run when submitted to the AEC. Additionally, the number of papers that had artifacts but did not submit to the AEC has steadily decreased to the point of being less than the AEC submission.

6.3 USENIX Security 2022 Artifact Evaluation

USENIX started its artifact evaluation in 2020 with 40 artifacts submitted and 38 passing [79]. The badge awarded was the *Artifact Evaluated.* In 2021, *USENIX* awarded 34 of the 37 submitted artifacts with the same badge [209]. In 2022, *USENIX* changed its badge awarding process. For the 114 papers submitted, it awarded one or more of *Artifact Available* (107), *Artifact Functional* (98), and *Artifact Reproduced* (65) [70]. We aim to understand how AEC assign badges. Further, *USENIX* made the artifact appendices (i.e., instructions to recreate) available for all papers submitted to its AEC as well as the badges associated with each paper [406]. While this is a step towards transparency, they do not provide an analysis of how the AEC determined each badge. We follow the submitted artifact appendix for each paper that overlaps with our indirect study for a total of 21 papers. Two papers only made their data available to the AEC as it was sensitive user data, thus we do not include them in our analysis. Of the 19 remaining papers, we find that we can recreate 15 of the paper's badges (i.e., when following the artifact appendix, we agree with 15 of the paper's badges). There are 4 papers for which we cannot recreate the badges.

Most of the cases where our evaluation differed from the AEC consisted of two issues: permission issues and package problems. In the case of permission issues, either data was unavailable for privacy or access control purposes or the commands to grab given to the AEC required permissions are given to the committee but not to the general public. Package problems are twofold, either code written with imports that were not designated for installation or make files that were out of date and errored out. There was a single case where the AEC did not give a tag that we did, which was an availability tag.

Based on the difference between our tests and the AEC, the most common issue is that code, packages, and make files are sensitive to updates and changes and there is no real incentive for any code to be maintained once it is accepted into a conference. In most cases, the setups were not a complete failure and usually failed toward the end of the installation process. Many of these systems would benefit from a pre-made instance (e.g., Docker or VM) since the installation processes were quite complex and it would guarantee resiliency to the constantly updating versions of packages.

Our study consists of papers published in the past 10 years. The number of ML papers published at Tier 1 security conferences has been steadily rising since 2013. Figure 9 shows the number of papers considered by year and conference. As the number of ML papers

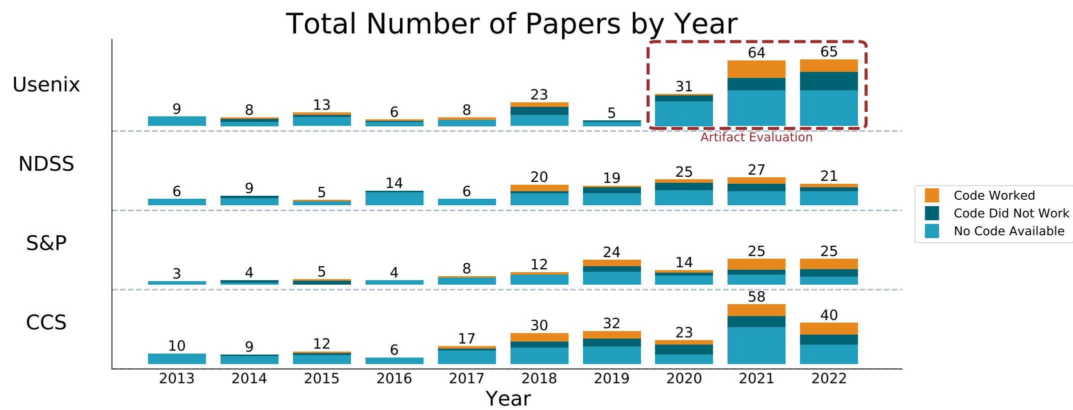

Figure 9: The change of experimental results across 10 years for each conference. As time progresses, we see that the number of papers considered in our study and the total number of papers with artifacts grows. The artifact evaluation commitee (AEC) that *USENIX* started in 2020 influenced both an increase in the number of artifacts and the proportion of artifacts that worked.

increases, there is an increase in the number of available artifacts. *USENIX*'s AEC was introduced in 2020, and it appears that this improves the code availability throughout the conference.

6.4 Academia and Industry

Past studies looking at systems research have observed low reproducibility among papers with authors from Industry [127]. To date, no one has performed a similar analysis in the Security community. We are unable to make any claims relating to this as Industry-related papers only account for 6% of the papers in this study. However, we understand that there are legitimate reasons companies or academics may not wish for every paper to be reproducible (e.g., private data, startups, or intellectual property). A larger discussion on the role of intentionally not publishing artifacts (e.g., intellectual property concerns, data sensitivity, etc) and how the community should collectively decide to evaluate the claims of those papers in relation to work with reproducible artifacts, should be had.

7 RECOMMENDATIONS

While there are numerous issues that we see throughout the direct study, we discuss five of the most common problems that we faced.

7.1 Packaging and Dependencies

The most common problem we face when running artifacts is maintaining consistent packages and dependencies as the artifact. Often, a specific version is not declared, and the current stable version does not work. However, this is not the only problem. Certain packages became deprecated or were no longer maintained resulting in numerous errors and hours of finding a workable version (e.g., Tensorflow 1 was deprecated in 2018). As such, tools like *pip* will not install it, yet we find that repositories from as late as 2022 still used a deprecated Tensorflow version. The install scripts or environments in the artifacts using this version did not properly set up the dependencies, forcing us to install it from the source. Other times the artifacts required unmaintained repositories and the current version is unstable with logged issues.

Lesson: When an artifact relies on complex dependencies or uses tools maintained by outside entities, there is an inherent risk that future work will not be able to use those artifacts. Special care is needed to identify what requirements an artifact needs and specify versions of any outside dependencies or packages.

Recommendation: It should be a standard practice to set a *requirements.txt* file with explicit version control with every repository and researchers need to be more consistent with providing self-contained environments (e.g., Docker images or Virtual Machines) to avoid issues with deprecation.

7.2 Incomplete Files or Data

A complete artifact contains every file required for running the code. This includes the data, preprocessing scripts, training scripts, and evaluation scripts. Often, the repositories were missing numerous files. As discussed in Section 6.1, the data availability adversely affected the reproducibility, but the missing data is not the only problem. We often saw repositories calling for functions that exist in a file that was not in the repository. We saw numerous spelling errors in the code, uninstantiated arrays, or call functions that were commented out. Some repositories provide the raw data but do not provide the labels or preprocessing scripts. Consequently, we are unable to reproduce their results. We saw numerous artifacts call on pre-trained models that were missing in the repository with no designation on how to acquire them. One repository even left coding the experiment for reproducing the results as something for the end-user to do. They provided the functions, data, and an outline to recreate the results but no script to do so.

Lesson: An artifact is only reproducible when all files required to do so are available. Scripts that contain bugs, call functions that do not exist, or rely on data not in the repository are difficult, if not impossible, to reproduce.

Recommendation: When creating a repository for a project, researchers should keep all files in a single location instead of requiring the end user to collect additional files from multiple locations. Additionally, if the community embraces artifact evaluation more, we can guarantee that research papers will have an instance where their work is not missing necessary files or code.

7.3 Incomplete Instructions

We outline in Section 6.1 the importance of ReadMes. While some ReadMe's contained no context or instructions, others are convoluted with excess information or instructions that are out-of-order. These instructions often called for confusing steps that should be unnecessary (e.g., changing every file path from a hardcoded, absolute path to our own absolute path). Another repository required root access, because it hardcoded an absolute path in their artifact. We find that some repositories never include the preprocessing of the data as an instruction. Yet further examples never mentioned running the script that labels the data for supervised learning.
Lesson: The steps involved to run a repository are often complex and cumbersome requiring non-trivial steps to get them to work. A lack of straightforward explicit instructions further complicates the effort required to run the artifact.
Recommendation: Each Readme should contain, at a minimum, a step-by-step set of instructions that explicitly give the exact commands necessary to run their system. If there are variable options to the command, an example should always accompany the command framework. Researchers should test that a non-expert can run their artifact solely based on the instructions that they provide.

7.4 Complex Hardware Setup

The designed systems often require a specific hardware setup both for collecting the data and running the system. In the case of collecting the data, we see that artifacts often do not include the selected data, requiring us to run their collection scripts. When the scripts use hardware that we do not have access to, we are unable to reproduce the results. When the artifact itself requires a complex hardware system to train, we struggle with setting up the system or adapting to the existing architecture that we have.
Lesson: While complex hardware improves performance or collects interesting data, by placing the burden on future work without consideration, it greatly increases the difficulty of reproducibility.
Recommendation: For specialized setups, the researchers should be aware that the reproducibility of their work is especially difficult if they fail to provide necessary factors such as their data. If data is collected with atypical hardware, when applicable, that data should be made available as an artifact of the paper.

7.5 Not Designed for Reproducibility

Finally, not all artifacts are designed for reproducibility. We often saw repositories that include examples, demonstrations, tools, or more supplementary information for the paper. As discussed in Section 6.1, the output did not match the claims in the paper 57% of the time. Artifacts often produced examples or online tools that allowed us to run a sample against their system (e.g., decompiling a binary), but we could not feasibly craft the output to reproduce their performance evaluation.
Lesson: Creating reproducible experiments is a conscious choice authors must make in computational sciences reflected in the state of their artifacts.
Recommendation: When approaching making projects reproducible, researchers need to ensure that every claim that is made in the paper can be reproduced in the artifact they release. Outputs to their code should be the table and data-driven figures that appear in the paper whenever possible. Researchers should be proactive and build their projects with reproducibility in mind for the design.

8 LIMITATIONS AND OPEN CHALLENGES

To encourage reproducibility in future work, we discuss the limitations of reproducing our study, further limitations within our study design, and the plethora of future work that exists in this area.

8.1 Reproducing Our Work

This body of work consists of over eight person-years of work from a large research team. We recognize that reproducing our work would take similarly significant time and resources. Furthermore, our work is inhibited by the fact that we cannot openly share a repository that has every paper's code and data. First, we do not own the code and are restricted by licenses for sharing their repositories. Second, most online repositories have a limit on the memory size of a repository. Including all 298 repositories of code exceeds this limit.

While we cannot avoid the above problems, we provide all of our processed data in a CSV in our repository. This CSV contains every paper, a URL link to the paper, a URL link to the code if one exists, and our coding of each paper. We also provide the scripts to create each figure in this paper, and we strongly encourage the reader to download and run our scripts to ensure that our figures are re-creatable from the data.

8.2 Limitations

To the best of our ability, we tried to limit biases and identify limitations. We acknowledge that there are still several biases and limitations and discuss them in this section.

First, selection bias could exist within our work. We do our best to systematically pick papers as noted in Section 4, and further confirm the selected papers by having a consensus with two reviewers. Further, our research questions are aimed at answering the state of reproducibility in machine learning at Tier 1 conferences in the Security community. There are possibly papers that we miss in our analysis due to our methodology. We also note that there are some code repositories that we may have missed. While we do our best to find online repositories that are not connected to the paper, our search is not exhaustive and some repositories may be left out.

Second, we recognize that parts of our analysis are subjective (e.g., what one reviewer considers as missing hyper-parameters may not be the same for another reviewer). We limit this by relying on objective measures as much as possible (e.g., paper X was missing the activation function) to inform a scale of the presence of a feature in our coding (e.g., instead of a binary class on whether paper X had hyper-parameters, we use varying degree). Further, there is a risk for any reviewer to favor negative results as that creates more interesting results [127, 443]. Thus, we accept the most positive result we can for a paper.

Finally, running, and in some cases training, multiple machine learning models is computationally and time intensive. We limit the amount of time a model was trained to 10 hours, and how much time we spend on debugging or attempting to set the repository up to one hour. We recognize that these limitations exist despite our best ability to limit them.

8.3 Open Challenges

One of the most significant but not discussed challenges to reproducibility comes in the form of funding for research. Often after papers are published, funding sources change and there are simply no means by which older projects can continue to be financially supported. Moreover, students and employees often move on to new positions, making it especially challenging to maintain complex research software in the long term. One potential means of improving outcomes in this space is making it easier for Funding Agencies to identify artifacts early and point their authors towards programs like the US National Science Foundation's "Transition to Practice" track. Additionally, authors should consider making their papers "reproducible by design", ensuring that artifacts are packaged into self-contained environments (e.g., Docker instances, VMs) whenever possible. Finally, we by no means recommend that significant effort be put into resurrecting the papers that we were unable to reproduce in this study; rather, effort and funding are likely better spent in making sure that future contributions improve their relative reproducibility.

While Section 2 clearly delineates the differences between reproducibility, replicability, and generalizability, our study focuses solely on the first of these goals. Studies into the latter two areas are extremely important and worth the attention of the community; however, due to their potential scale (e.g., collecting fundamentally new datasets), future studies may need to be even more narrowly scoped than our study of machine learning security. Lastly, as mentioned early in this work, we are unable to consider the correctness of implementations in this study - only the performance of available code to reported values. There is substantial research work to be done in this space that would result in significant improvement in the trust of research claims made by our community.

9 RELATED WORK

Our study is the first reproducibility study to *comprehensively* measure the reproducibility of machine learning in the Security community. We perform both an indirect and direct study of reproducibility that serves as the foundation for reproducibility studies in Security, complementing a vast swath of work in similar fields. We outline notable contributions in different areas and differentiate our work from previous work.

Computer Science: Prior reproducibility studies in computer science evaluated the availability of artifacts in venues such as *IEEE Transactions of Signal Processing* [634], *AIS International Conference of Information Systems* [496], and various *ACM* journals and conferences [127]. The indirect studies found that in 2004 only 9% of 134 considered papers had code and 33% provided a dataset [634], which improved to 28% of the 100 papers evaluated in 2019 providing code [496]. When directly studied, only 32% of considered papers in 2014 could be compiled within 30 minutes [127]. However, they did not evaluate whether the artifact recreated the results claimed in the paper.

Machine Learning: Machine learning is subject to the same concerns of reproducibility, if not more so[211]. Randomness greatly influences the performance of machine learning models [25], and only 8% of 45 evaluated papers between 2015 and 2018 discussed how randomness affected their model [363]. Even more worrisome, 11 of 12 reproduced recommender systems were outperformed by conceptually simpler models over multiple splits of the dataset [177]. Further compounding this issue, only 25% of papers published at *AAAI* and *IJCAI* in 2013 to 2016 described their method in a reproducible way [212]. Though, Raff [495] found that they could replicate results from described methodology 62% of the time looking at 255 papers from 1984 to 2017. While the extent of availability of artifacts has been observed in other conferences, our study is more comprehensive with a larger scope in Tier 1 conferences that have not been considered before.

Security: As noted in Section 2.2, calls for better reproducibility in the Security community have made slow changes to conferences. Yet, to the best of our knowledge, there are only two studies on reproducibility in the Security community that attempt to measure this problem. Van et al. [633] looked at 50 systems security papers from *USENIX*, *CCS*, *NDSS*, and *S&P* in the years 2010 and 2015. Their paper primarily focused on benchmarking flaws in the papers they looked at where 1 of the 5 pillars for errors was reproducibility. They found that approximately 1 out of 4 papers did not specify their platform or their software version. Hamm et al. [215] looked at 61 user studies from *Usenix*, *CCS*, and *S&P* from 2013 to 2018. They found that 51% of the papers offered up the questionnaire used in the user study and none provided the full response data. Both of these studies are indirect studies. Our paper expands beyond both of these papers in both scope and depth. We consider 744 papers and evaluated the papers in both an indirect and direct study.

10 CONCLUSION

Academic research often prioritizes novel, exploratory research. However, without reproducible science, the widespread adoption of new ideas and their transition to practice can be severely degraded. Measuring where our community stands in terms of reproducible science is crucial to making recommendations that meaningful help to normalize such contributions. We perform the first such longitudinal study of computational reproducibility in Computer Security research, investigating both indirect and direct reproducibility over a decade of publications. To our knowledge, our study is the most comprehensive reproducibility analysis of our community in size by at least an order of magnitude. Our results show both that making working artifacts available is not yet a priority of the community and that having artifact evaluation committees and badging may be leading to improvements.

Most critically, we show that common platitudes regarding reproducibility (i.e., "Just make code available") fail to meaningfully move our community forward. Instead, where making artifacts public is possible, researchers should focus on improving the five most-common issues preventing their work from being reproduced: packages and dependencies, incomplete or missing files/data, incomplete or confusing instructions, distribution of artifacts from complex hardware setups, and not designed for reproducibility. While other issues beyond the control of individual researchers still exist (i.e., funding for the continued maintenance of said artifacts, legal limitations on distributing code and/or datasets, etc), we believe that addressing these issues will help to make more papers "reproducible by design".

REFERENCES

[1] 2020. Artifact review and badging - current. https://www.acm.org/publications/policies/artifact-review-and-badging-current

[2] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. In *USENIX Security*.

[3] Sumayah A. Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, Xiaofeng Wang, Feng Qian, RaheemA. Beyah, and Damon McCoy. 2017. Under the Shadow of Sunshine: Understanding and Detecting Bulletproof Hosting on Legitimate Service Provider Networks. In *IEEE S&P*.

[4] Yousra Aafer, Wei You, Yi Sun, Yu Shi, Xiangyu Zhang, and Heng Yin. 2021. Android SmartTVs Vulnerability Discovery via Log-Guided Fuzzing. In *USENIX Security*.

[5] Martín Abadi, H. Brendan McMahan, Andy Chu, Ian Goodfellow, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *ACM CCS*.

[6] Sahar Abdelnabi and Mario Fritz. 2021. Adversarial Watermarking Transformer: Towards Tracing Text Provenance with Data Hiding. In *IEEE S&P*.

[7] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. VisualPhishNet: Zero-Day Phishing Website Detection by Visual Similarity. In *ACM CCS*.

[8] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin Butler, and Joseph Wilson. 2019. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In *NDSS*.

[9] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. 2021. Hear "No Evil", See "Kenansville"*: Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems. In *IEEE S&P*.

[10] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. 2021. SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *IEEE S&P*.

[11] Ismi Abidi, Ishan Nangia, Paarijaat Aditya, and Rijurekha Sen. 2022. Privacy in Urban Sensing with Instrumented Fleets, Using Air Pollution Monitoring As A Usecase. In *NDSS*.

[12] Hmed Mohammed Abuhamad, Tamer Abu, Aziz Mohaisen, and DaeHun Nyang. 2018. Large-Scale and Language-Oblivious Code Authorship Identification. In *ACM CCS*.

[13] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. In *USENIX Security*.

[14] Sadia Afroz, Aylin Caliskan-Islam, Ariel Stolerman, Rachel Greenstadt, and Damon McCoy. 2014. Doppelgänger Finder: Taking Stylometry To The Underground. In *IEEE S&P*.

[15] Hojjat Aghakhani, Fabio Gritti, Francesco Mecca, Martina Lindorfer, Stefano Ortolani, Davide Balzarotti, Giovanni Vigna, and Christopher Kruegel. 2020. When Malware is Packin' Heat; Limits of Machine Learning Classifiers Based on Static Analysis Features. In *NDSS*.

[16] Mayank Agrawal, Rebecca Han, Dinushka Herath, and David S Sholl. 2020. Does repeat synthesis in materials chemistry obey a power law? *Proceedings of the National Academy of Sciences* 117, 2 (2020), 877–882.

[17] Nitin Agrawal, Ali Shahin Shamsabadi, Matt J. Kusner, and Adrià Gascón. 2019. QUOTIENT: Two-Party Secure Neural Network Training and Prediction. In *ACM CCS*.

[18] Mansour Ahmadi, Reza Farkhani, Mirzazade, Ryan Williams, and Long Lu. 2021. Finding Bugs Using Your Own Code: Detecting Functionally-similar yet Inconsistent Code. In *USENIX Security*.

[19] Shimaa Ahmed, AmritaRoy Chowdhury, Kassem Fawaz, and Parmesh Ramanathan. 2020. Preech: A System for Privacy-Preserving Speech Transcription. In *USENIX Security*.

[20] Shimaa Ahmed, Ilia Shumailov, Nicolas Papernot, and Kassem Fawaz. 2022. Towards More Robust Keyword Spotting for Voice Assistants. In *USENIX Security*.

[21] Sumayah Alrwais, Xiaojing Liao, Xianghang Mi, Peng Wang, XiaoFeng Wang, Feng Qian, Raheem Beyah, and Damon McCoy. 2017. Detecting Bulletproof Hosting on Legitimate Service Provider Networks. In *IEEE S&P*.

[22] Abdulellah Alsaheel, Yuhong Nan, Shiqing Ma, Le Yu, Gregory Walkup, Z. Berkay Celik, Xiangyu Zhang, and Dongyan Xu. 2021. ATLAS: A Sequence-based Learning Approach for Attack Investigation. In *USENIX Security*.

[23] Shengwei An, Guanhong Tao, Qiuling Xu, Yingqi Liu, Guangyu Shen, Yuan Yao, Jingwei Xu, and Xiangyu Zhang. 2022. MIRROR: Model Inversion for Deep Learning Network with High Fidelity. In *NDSS*.

[24] Patricia Arias-Cabarcos, Thilo Habrich, Karen Becker, Christian Becker, and Thorsten Strufe. 2021. Inexpensive Brainwave Authentication: New Techniques and Insights on User Acceptance. In *USENIX Security*.

[25] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and Don'ts of Machine Learning in Computer Security. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 3971–3988.

[26] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and Don'ts of Machine Learning in Computer Security. In *USENIX Security*.

[27] Daniel Arp, Michael Spreitzenbarth, Malte Hubner, Hugo Gascon, and Konrad Rieck. 2014. Drebin: Effective and Explainable Detection of Android Malware in Your Pocket. In *NDSS*.

[28] Mozhgan Azimpourkivi and Umut Topkara. 2020. Human Distinguishable Visual Key Fingerprints. In *USENIX Security*.

[29] Ahmadreza Azizi, Ibrahim Asadullah Tahmid, Asim Waheed, Neal Mangaokar, Jiameng Pu, Mobin Javed, Chandan K. Reddy, and Bimal Viswanath. 2021. T-Miner: A Generative Approach to Defend Against Trojan Attacks on DNN-based Text Classification. In *USENIX Security*.

[30] Zhongjie Ba, Tianhang Zheng, Xinyu Zhang, Zhan Qin, Baochun Li, Xue Liu, and Kui Ren. 2020. Learning-based Practical Smartphone Eavesdropping with Built-in Accelerometer. In *NDSS*.

[31] Sangwook Bae, Mincheol Son, Dongkwan Kim, CheolJun Park, Jiho Lee, Sooel Son, and Yongdae Kim. 2022. Watching the Watchers: Practical Video Identification Attack in LTE Networks. In *USENIX Security*.

[32] Eugene Bagdasaryan and Vitaly Shmatikov. 2021. Blind Backdoors in Deep Learning Models. In *USENIX Security*.

[33] Alireza Bahramali, Milad Nasr, Amir Houmansadr, Dennis Goeckel, and Don Towsley. 2021. Robust Adversarial Attacks Against DNN-Based Wireless Communication Systems. In *ACM CCS*.

[34] Zhihao Bai, Ke Wang, Hang Zhu, Yinzhi Cao, and Xin Jin. 2021. Runtime Recovery of Web Applications under Zero-Day ReDoS Attacks. In *IEEE S&P*.

[35] Monya Baker. 2015. Over half of psychology studies fail reproducibility test. *Nature* 27 (2015), 1–3.

[36] Borja Balle, Giovanni Cherubin, and Jamie Hayes. 2022. Reconstructing Training Data with Informed Adversaries. In *IEEE S&P*.

[37] Teodora Baluta, Shiqi Shen, S. Hitarth, Shruti Tople, and Prateek Saxena. 2022. Membership Inference Attacks and Generalization: A Causal Perspective. In *ACM CCS*.

[38] Teodora Baluta, Shiqi Shen, Shweta Shinde, Kuldeep S. Meel, and Prateek Saxena. 2019. Quantitative Verification of Neural Networks and Its Security Applications. In *ACM CCS*.

[39] Sebastian Banescu, Christian Collberg, and Alexander Pretschner. 2017. Predicting the Resilience of Obfuscated Code Against Symbolic Execution Attacks via Machine Learning. In *USENIX Security*.

[40] Tiffany Bao, Jonathan Burket, Maverick Woo, Rafael Turner, and David Brumley. 2014. ByteWeight: Learning to Recognize Functions in Binary Code. In *USENIX Security*.

[41] Federico Barbero, Feargus Pendlebury, Fabio Pierazzi, and Lorenzo Cavallaro. 2022. Transcending TRANSCEND: Revisiting Malware Classification in the Presence of Concept Drift. In *IEEE S&P*.

[42] Diogo Barradas, Nuno Santos, and Luıs Rodrigues. 2018. Effective Detection of Multimedia Protocol Tunneling using Machine Learning. In *USENIX Security*.

[43] Diogo Barradas, Nuno Santos, Luís Rodrigues, and Vítor Nunes. 2020. Poking a Hole in the Wall: E!icient Censorship-Resistant Internet Communications by Parasitizing on WebRTC. In *ACM CCS*.

[44] Diogo Barradas, Nuno Santos, Lúis Rodrigues, Salvatore Signorello, Fernando M.V. Ramos, and André Madeira. 2021. FlowLens: Enabling Efficient Flow Classification for ML-based Network Security Applications. In *NDSS*.

[45] Gilles Barthem, Gian Pietro Farina, Marco Gaboardi, Emilio Jesús Gallego Arias, Andy Gordon, Justin Hsu, and Pierre-Yves Strub. 2016. Differentially Private Bayesian Programming. In *ACM CCS*.

[46] Armon Barton, Mohsen Imani, and Jiang Ming. 2018. Towards Predicting Efficient and Anonymous Tor Circuits. In *USENIX Security*.

[47] Adam Bates, Ryan Leonard, Hannah Pruse, Daniel Lowd, and Kevin Butler. 2014. Leveraging USB to Establish Host Identity Using Commodity Devices. In *NDSS*.

[48] M Beller. 2020. Why I will never join an Artifacts Evaluation Committee Again. *Inventitech. com. https://inventitech. com/blog/why-i-will-never-review-artifacts-again/(Accessed: Feb. 9, 2022)* (2020).

[49] Daniel J Benjamin and James O Berger. 2019. Three recommendations for improving the use of p-values. *The American Statistician* 73, sup1 (2019), 186–191.

[50] Terry Benzel. 2023. Security and Privacy Research Artifacts: Are We Making Progress? *IEEE Security & Privacy* 21, 01 (2023), 4–6.

[51] Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. 2020. Ligero++: A New Optimized Sublinear IOP. In *ACM CCS*.

[52] Shivam Bhasin, Anupam Chattopadhyay, Annelie Heuser, Dirmanto Jap, Stjepan Picek, and Ritu Ranjan Shrivastwa. 2020. Mind the Portability: A Warriors Guide through Realistic Profiled Side-channel Analysis. In *NDSS*.

[53] Rohit Bhatia, Vireshwar Kumar, Khaled Serag, Z. Berkay Celik, Mathias Payer, and Dongyan Xu. 2021. Evading Voltage-Based Intrusion Detection on Automotive CAN. In *NDSS*.

[54] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin Vechev. 2021. DP-Sniper: Black-Box Discovery of Differential Privacy Violations using Classifiers. In *IEEE S&P*.
[55] Leyla Bilge, Yufei Han, and Matteo DellAmico. 2017. RiskTeller: Predicting the Risk of Cyber Incidents. In *ACM CCS*.
[56] Igor Bilogrevic, Kévin Huguenin, Stefan Mihaila, Reza Shokri, and Jean-Pierre Hubaux. 2015. Predicting Users Motivations behind Location Check-Ins and Utility Implications of Privacy Protection Mechanisms. In *NDSS*.
[57] Simon Birnbach, Simon Eberz, and Ivan Martinovic. 2019. Peeves: Physical Event Verification in Smart Homes. In *ACM CCS*.
[58] Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. 2022. Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction. In *USENIX Security*.
[59] Eric Bodden, Siegfried Rasthofer, and Steven Arzt. 2014. A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks. In *NDSS*.
[60] Kenneth Bollen, John T Cacioppo, Robert M Kaplan, Jon A Krosnick, James L Olds, and Heather Dean. 2015. Social, behavioral, and economic sciences perspectives on robust and reliable science. *Report of the Subcommittee on Replicability in Science Advisory Committee to the National Science Foundation Directorate for Social, Behavioral, and Economic Sciences* 1 (2015).
[61] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. 2022. Automating Cookie Consent and GDPR Violation Detection. In *USENIX Security*.
[62] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM CCS*.
[63] Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna. 2013. Delta: Automatic Identification of Unknown Web-based Infection Campaigns. In *ACM CCS*.
[64] Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna. 2015. Meerkat: Detecting Website Defacements through Image-based Object Recognition. In *USENIX Security*.
[65] Yazan Boshmaf, Dionysios Logothetis, Georgos Siganos, Jorge Lería, José Lorenzo, Matei Ripeanu, and Konstantin Beznosov. 2015. Integro: Leveraging Victim Prediction for Robust Fake Account Detection in OSNs. In *NDSS*.
[66] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. 2015. Machine Learning Classification over Encrypted Data. In *NDSS*.
[67] Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. BadCharacters: Imperceptible NLP Attacks. In *IEEE S&P*.
[68] Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. 2021. Machine Unlearning. In *IEEE S&P*.
[69] Duc Bui, Yuan Yo, Kang G. Shin, Jong-Min Choi, and Junbum Shin. 2021. Consistency Analysis of Data-Usage Purposes in Mobile Apps. In *ACM CCS*.
[70] Kevin Butler and Kurt Thomas. 2022. Message from the USENIX Security'22 program co-chairs. In *31st USENIX Security Symposium, USENIX Security 2022.*
[71] Niklas Büscher, Daniel Demmler, Stefan Katzenbeisser, David Kretzmer, and Thomas Schneider. 2018. HyCC: Compilation of Hybrid Protocols for Practical Secure Computation. In *ACM CCS*.
[72] Aylin Caliskan, Fabian Yamaguchi, Edwin Dauber, Richard Harang, Konrad Rieck, Rachel Greenstadt, and Arvind Narayanan. 2018. When Coding Style Survives Compilation: De-anonymizing Programmers from Executable Binaries. In *NDSS*.
[73] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. 2015. De-anonymizing Programmers via Code Stylometry. In *USENIX Security*.
[74] Qiang Cao, Xiaowei Yang, Jieqi Yu, and Christopher Palow. 2014. Uncovering Large Groups of Active Malicious Accounts in Online Social Networks. In *ACM CCS*.
[75] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. 2021. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In *NDSS*.
[76] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. 2021. Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks. In *IEEE S&P*.
[77] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. 2019. Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving. In *ACM CCS*.
[78] Yinzhi Cao and Junfeng Yang. 2015. Towards Making Systems Forget with Machine Unlearning. In *IEEE S&P*.
[79] Srdjan Čapkun and Franziska Roesner. 2020. Message from the USENIX Security'20 program co-chairs. In *29th USENIX Security Symposium, USENIX Security 2020.*
[80] Matteo Cardaioli, Stefano Cecconello, Mauro Conti, Simone Milani, Stjepan Picek, and Eugen Saraci. 2022. Hand Me Your PIN! Inferring ATM PINs of Users Typing with a Covered Hand. In *USENIX Security*.
[81] Nicholas Carlini. 2021. Poisoning the Unlabeled Dataset of Semi-Supervised Learning. In *USENIX Security*.
[82] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tamér. 2022. Membership Inference Attacks From First Principles. In *IEEE S&P*.
[83] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Abhradeep Thakurta, and Florian Tramèr. 2021. Is Private Learning Possible with Instance Encoding?. In *IEEE S&P*.
[84] Nicholas Carlini, Pratyush Mishra, Yuankai Zhang, Micah Sherr, Clay Shields, and Wenchao Zhou. 2016. Hidden Voice Commands. In *USENIX Security*.
[85] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. In *USENIX Security*.
[86] Nicholas Carlini and David Wagner. 2017. Towards Evaluating the Robustness of Neural Networks. In *IEEE S&P*.
[87] Curtis Carmony, Xunchao Hu, Heng Yin, AbhishekVasisht Bhaskar, and Mu Zhang. 2016. Extract Me If You Can: Abusing PDF Parsers in Malware Detectors. In *NDSS*.
[88] Nishanth Chandran, Divya Gupta, Sai Lakshmi Bhavana Obbattu, and Akash Shah. 2022. Simc: ML Inference Secure Against Malicious Clients at Semi-Honest Cost. In *USENIX Security*.
[89] Varun Chandrasekaran, Kamalika Chaudhuri, Somesh Jha, and Songbai Yan. 2020. Exploring Connections Between Active Learning and Model Extraction. In *USENIX Security*.
[90] Thee Chanyaswad, Alex Dytso, H. Vincent Poor, and Prateek Mittal. 2018. MVG Mechanism: Differential Privacy under Matrix-Valued Query. In *ACM CCS*.
[91] Rahul Chatterjee, Joseph Bonneau, Ari Juels, and Thomas Ristenpart. 2015. Cracking-Resistant Password Vaults Using Natural Language Encoders. In *IEEE S&P*.
[92] Rahul Chatterjee, Periwinkle Doerfler, Hadas Orgad, Sam Havron, Jackeline Palmer, Diana Freed, Karen Levy, Nicola Dell, Damon McCoy, and Thomas Ristenpart. 2018. The Spyware Used in Intimate Partner Violence. In *IEEE S&P*.
[93] Rahul Chatterjee, M. Sadegh Riazi, Tanmoy Chowdhury, Emanuela Marasco, Farinaz Koushanfar, and Ari Juels. 2019. Multisketches: Practical Secure Sketches Using Off-the-Shelf Biometric Matching Algorithms. In *ACM CCS*.
[94] Rahul Chatterjee, Joanne Woodage, Yuval Pnueli, Anusha Chowdhury, and Thomas Ristenpart. 2017. The TypTop System: Personalized Typo-Tolerant Password Checking. In *ACM CCS*.
[95] Harsh Chaudhari, Rahul Rachuri, and Ajith Suresh. 2020. Trident: Efficient 4PC Framework for Privacy Preserving Machine Learning. In *NDSS*.
[96] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *ACM CCS*.
[97] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. In *IEEE S&P*.
[98] Hao Chen, Ilaria Chillotti, Yihe Dong, Oxana Poburinnaya, Ilya Razenshteyn, and M. Sadegh Riaza. 2020. SANNS: Scaling Up Secure Approximate k-Nearest Neighbors Search. In *USENIX Security*.
[99] Hao Chen, Wei Dai, Miran Kim, and Yongsoo Song. 2019. Efficient Multi-Key Homomorphic Encryption with Packed Ciphertexts with Application to Oblivious Neural Network Inference. In *ACM CCS*.
[100] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. 2020. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. In *IEEE S&P*.
[101] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2021. When Machine Unlearning Jeopardizes Privacy. In *ACM CCS*.
[102] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph Unlearning. In *ACM CCS*.
[103] Peng Chen and Hao Chen. 2018. Angora: Efficient Fuzzing by Principled Search. In *IEEE S&P*.
[104] Qibin Chen, Jeremy Lacomis, Edward J. Schwartz, Claire Le Goues, Graham Neubig, and Bogdan Vasilescu. 2022. Augmenting Decompiler Output with Learned Variable Names and Types. In *USENIX Security*.
[105] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. 2020. Metamorph: Injecting Inaudible Commands into Over-the-air Voice Controlled Systems. In *NDSS*.
[106] Yanjiao Chen, Yijie Bai, Richard Mitev, Kaibo Wang, Ahmad-Reza Sadeghi, and Wenyuan Xu. 2021. FakeWake: Understanding and Mitigating Fake Wake-up Words of Voice Assistants. In *ACM CCS*.
[107] Yuan Chen, Jiaqi Li, Guorui Xu, Yajin Zhou, Zhi Wang, Cong Wang, and Kui Ren. 2022. SGXLock: Towards Efficiently Establishing Mutual Distrust Between Host Application and Enclave for SGX. In *USENIX Security*.
[108] Yizheng Chen, Yacin Nadji, Athanasios Kountouras, Fabian Monrose, Roberto Perdisci, Manos Antonakakis, and Nikolaos Vasiloglou. 2017. Practical Attacks Against Graph-based Clustering. In *ACM CCS*.

[109] Yuqi Chen, ChristopherM. Poskitt, and Jun Sun. 2018. Learning from Mutants: Using Code Mutation to Learn and Monitor Invariants of a Cyber-Physical System. In *IEEE S&P*.
[110] Yufei Chen, Chao Shen, Cong Wang, and Yang Zhang. 2022. Teacher Model Fingerprinting Attacks Against Transfer Learning. In *USENIX Security*.
[111] Yi Chen, Di Tang, Yepeng Yao, Mingming Zha, XiaoFeng Wang, Xiaozhong Liu, Haixu Tang, and Dongfang Zhao. 2022. Seeing the Forest for the Trees: Understanding Security Hazards in the 3GPP Ecosystem through Intelligent Analysis on Change Requests. In *USENIX Security*.
[112] Yizheng Chen, Shiqi Wang, Weifan Jiang, Asaf Cidon, and Suman Jana. 2021. Cost-Aware Robust Tree Ensembles for Security Applications. In *USENIX Security*.
[113] Yizheng Chen, Shiqi Wang, Yue Qin, Xiaojing Liao, Suman Hana, and David Wagner. 2021. Learning Security Classifiers with Verified Global Robustness Properties. In *ACM CCS*.
[114] Yizheng Chen, Shiqi Wang, Dongdong She, and Suman Jana. 2020. On Training Robust PDF Malware Classifiers. In *USENIX Security*.
[115] Yi Chen, Yepeng Yao, XiaoFeng Wang, Dandan Xu, Chang Yue, Xiaozhong Liu, Kai Chen, Haixu Tang, and Baoxu Liu. 2021. Bookworm Game: Automatic Discovery of LTE Vulnerabilities Through Documentation Analysis. In *IEEE S&P*.
[116] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. Devil's Whisper: A General Approach for Physical Adversarial Attacks against Commercial Black-box Speech Recognition Devices. In *USENIX Security*.
[117] Yi Chen, Mingming Zha, Nan Zhang, Dandan Xu, Qianqian Zhao, Xuan Feng, Kan Yuan, Fnu Suya, Yuan Tian, Kai Chen, XiaoFeng Wang, and Wei Zou. 2019. Demystifying Hidden Privacy Settings in Mobile Apps. In *IEEE S&P*.
[118] Haibo Cheng, Wenting Li, and Ping Wang. 2021. Incrementally Updateable Honey Password Vaults. In *USENIX Security*.
[119] Yushi Cheng, Xiaoyu Ji, Juchuan Zhang, Wenyuan Xu, and Yi-Chao Chen. 2019. DeMiCPU: Device Fingerprinting with Magnetic Signals Radiated by CPU. In *ACM CCS*.
[120] Giovanni Cherubin, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2019. F-BLEAU: Fast Black-box Leakage Estimation. In *IEEE S&P*.
[121] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. 2022. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In *USENIX Security*.
[122] Garret Christensen and Edward Miguel. 2018. Transparency, reproducibility, and the credibility of economics research. *Journal of Economic Literature* 56, 3 (2018), 920–980.
[123] ZhengLeong Chua, Shiqi Shen, Prateek Saxena, and Zhenkai Liang. 2017. Neural Nets Can Learn Function Type Signatures From Binaries. In *USENIX Security*.
[124] Kenneth T. Co, Luis Muñoz-González, Sixtede Maupeou, and Emil C. Lupu. 2019. Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Convolutional Networks. In *ACM CCS*.
[125] Benjamin Coleman and Anshumali Shrivastava. 2021. A One-Pass Distributed and Private Sketch for Kernel Sums with Applications to Machine Learning at Scale. In *ACM CCS*.
[126] Open Science Collaboration. 2012. An open, large-scale, collaborative effort to estimate the reproducibility of psychological science. *Perspectives on Psychological Science* 7, 6 (2012), 657–660.
[127] Christian Collberg, Todd Proebsting, and Alex M Warren. 2015. Repeatability and benefaction in computer systems research. *University of Arizona TR* 14, 4 (2015).
[128] Tianshuo Cong, Xinlei He, and Yang Zhang. 2022. SSLGuard: A Watermarking Scheme for Self-supervised Learning Pre-trained Encoders. In *ACM CCS*.
[129] Ana-Maria Crețu, Florimond Houssiau, Antoine Cully, and Yves-Alexandrede Montjoye. 2022. QuerySnout: Automating the Discovery of Attribute Inference Attacks against Query-Based Systems. In *ACM CCS*.
[130] Patrick Cronin, Xing Gao, Haining Wang, and Chase Cotton. 2022. Time-Print: Authenticating USB Flash Drives with Novel Timing Fingerprints. In *IEEE S&P*.
[131] Patrick Cronin, Xing Gao, and Chengmo Yang. 2021. Charger-Surfing: Exploiting a Power Line Side-Channel for Smartphone Information Leakage. In *USENIX Security*.
[132] Tianyu Cui, Gaopeng Gou, Gang Xiong, Zhen Li, Mingxin Cui, and Chang Liu. 2021. SiamHAN: IPv6 Address Correlation Attacks on TLS Encrypted Traffic via Siamese Heterogeneous Graph Attention Network. In *USENIX Security*.
[133] Jiarun Dai, Yuan Zhang, Zheyue Jiang, Yingtian Zhou, Junyan Chen, Xinyu Xing, Xiaohan Zhang, Xin Tan, Min Yang, and Zhemin Yang. 2020. BScout: Direct Whole Patch Presence Test for Java Executables. In *USENIX Security*.
[134] Anders Dalskov, Daniel Escudero, and Marcel Keller. 2021. Fantastic Four: Honest-Majority Four-Party Secure Computation With Malicious Security. In *USENIX Security*.
[135] Ivan Damgård, Daniel Escudero, Tore Frederiksen, Marcel Keller, Peter Scholl, and Nikolaj Volgushev. 2019. New Primitives for Actively-Secure MPC over Rings with Applications to Private Machine Learning. In *IEEE S&P*.
[136] Hung Dang, Yue Huang, and Ee-Chien Chang. 2017. Evading Classifiers by Morphing in the Dark. In *ACM CCS*.
[137] Anupam Das, Gunes Acar, Nikita Borisov, and Amogh Pradeep. 2018. The Web's Sixth Sense: A Study of Scripts Accessing Smartphone Sensors. In *ACM CCS*.
[138] Anupam Das, Nikita Borisov, and Matthew Caesar. 2016. Tracking Mobile Web Users Through Motion Sensors: Attacks and Defenses. In *NDSS*.
[139] Sanjeev Das, Jan Werner, Manos Antonakakis, Michalis Polychronakis, and Fabian Monrose. 2019. SoK: The Challenges, Pitfalls, and Perils of Using Hardware Performance Counters for Security. In *IEEE S&P*.
[140] Anupam Datta, Matthew Fredrikson, Gihyuk Ko, Piotr Mardziel, and Shayak Sen. 2017. Use Privacy in Data-Driven Systems: Theory and Experiments with Machine Learnt Programs. In *ACM CCS*.
[141] Anupam Datta, Shayak Sen, and Yair Zick. 2016. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. In *IEEE S&P*.
[142] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Klaus Wehrle, Thomas Engel, and Andriy Panchenko. 2020. TrafficSliver: Fighting Website Fingerprinting Attacks with Traffic Splitting. In *ACM CCS*.
[143] Gonzalo De La Torre Parra, Luis Selvera, Joseph Khoury, Hector Irizarry, Elias Bou-Harb, and Paul Rad. 2022. Interpretable Federated Transformer Log Learning for Cloud Threat Forensics. In *NDSS*.
[144] Soteris Demetriou, Whitney Merrill, Wei Yang, Aston Zhang, and CarlA. Gunter. 2016. Free for All! Assessing User Data Exposure to Advertising Libraries on Android. In *NDSS*.
[145] Zizhuang Deng, Kai Chen, Guozhu Meng, Xiaodong Zhang, Ke Xu, and Yao Cheng. 2022. Understanding Real-world Threats to Deep Learning Models in Android Apps. In *ACM CCS*.
[146] Sanorita Dey, Nirupam Roy, Wenyuan Xu, RomitRoy Choudhury, and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable. In *NDSS*.
[147] Michalis Diamantaris, Serafeim Moustakas, Lichao Sun, Sotiris Ioannidis, and Jason Polakis. 2021. This Sneaky Piggy Went to the Android Ad Market: Misusing Mobile Sensors for Stealthy Data Exfiltration. In *ACM CCS*.
[148] Zainul Abi Din, Hari Venugopalan, Henry Lin, Adam Wushensky, Steven Liu, and Samuel T. King. 2021. Doing good by fighting fraud: Ethical anti-fraud systems for mobile payments. In *IEEE S&P*.
[149] Hailun Ding, Shenao Yan, Juan Zhai, and Shiqing Ma. 2021. ELISE: A Storage Efficient Logging System Powered by Redundancy Reduction and Representation Learning. In *USENIX Security*.
[150] Steven H.H. Ding, BenjaminC.M. Fung, and Philippe Charland. 2019. Asm2Vec: Boosting Static Representation Robustness for Binary Clone Search against Code Obfuscation and Compiler Optimization. In *IEEE S&P*.
[151] Priyanka Dodia, Mashael AlSabah, Omar Alrawi, and Tao Wang. 2022. Exposing the Rat in the Tunnel: Using Traffic Analysis for Tor-based Malware Detection. In *ACM CCS*.
[152] Brendan Dolan-Gavitt, Tim Leek, Josh Hodosh, and Wenke Lee. 2013. Tappan Zee (North) Bridge: Mining Memory Accesses for Introspection. In *ACM CCS*.
[153] Ying Dong, Wenbo Guo, Yueqi Chen, Xinyu Xing, Yuqing Zhang, and Gang Wang. 2019. Towards the Detection of Inconsistencies in Public Security Vulnerability Reports. In *USENIX Security*.
[154] Evan Downing, Yisroel Mirsky, Kyuhong Park, and Wenke Lee. 2021. DeepReflect: Discovering Malicious Functionality through Binary Reconstruction. In *USENIX Security*.
[155] Min Du, Zhi Chen, Chang Liu, Rajvardhan Oak, and Dawn Song. 2019. Lifelong Anomaly Detection Through Unlearning. In *ACM CCS*.
[156] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. 2017. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In *ACM CCS*.
[157] Tianyu Du, Shouling Ji, Lujia Shen, Yao Zhang, Jinfeng Li, Jie Shi, Chengfang Fang, Jianwei Yin, Raheem Beeyah, and Ting Wang. 2021. Cert-RNN: Towards Certifying the Robustness of Recurrent Neural Networks. In *ACM CCS*.
[158] Rui Duan, Zhe Qu, Shangqing Zhao, Leah Ding, Yao Liu, and Zhuo Lu. 2022. Perception-Aware Attack: Creating Adversarial Music via Reverse-Engineering Human Perception. In *ACM CCS*.
[159] Yue Duan, Xuezixiang Li, Jinghan Wang, and Heng Yin. 2020. DEEPBINDIFF: Learning Program-Wide Code Representations for Binary Diffing. In *NDSS*.
[160] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. 2022. FLAME: Taming Backdoors in Federated Learning. In *USENIX Security*.
[161] Simon Eberz, Giulio Lovisotto, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. 2019. 28 Blinks Later: Tackling Practical Challenges of Eye Movement Biometrics. In *ACM CCS*.
[162] Simon Eberz, Kasper B. Rasmussen, Vincent Lenders, and Ivan Martinovic. 2015. Preventing Lunchtime Attacks: Fighting Insider Threats With Eye Movement Biometrics. In *NDSS*.

[163] Thijsvan Ede, Hojjat Aghakhani, Noah Spahn, Riccardo Bortolameotti, Marco Cova, Andrea Continella, Maartenvan Steen, Andreas Peter, Christopher Kruegel, and Giovanni Vigna. 2022. DEEPCASE: Semi-Supervised Contextual Analysis of Security Events. In *IEEE S&P*.

[164] Thijsvan Ede, Riccardo Bortolameotti, Andrea Continella, Jingjing Ren, Daniel J. Dubois, Martina Lindorfer, David Choffnes, Maartenvan Steen, and Andreas Peter. 2020. FLOWPRINT: Semi-Supervised Mobile-App Fingerprinting on Encrypted Network Traffic. In *NDSS*.

[165] Manuel Egele, Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. COMPA: Detecting Compromised Accounts on Social Networks. In *NDSS*.

[166] Thorsten Eisenhofer, Lea Schönherr, Joel Frank, Lars Speckemeier, Dorothea Kolossa, and Thorsten Holz. 2021. Dompteur: Taming Audio Adversarial Examples. In *USENIX Security*.

[167] Muhammad Ejaz Ahmed, Il-Youp Kwak, JunHo Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In *USENIX Security*.

[168] Sebastian Eschweiler, Khaled Yakdan, and Elmar Gerhards-Padilla. 2016. discovRE: Efficient Cross-Architecture Identification of Bugs in Binary Code. In *NDSS*.

[169] Sina Faezi, Sujit Rokka Chhetri, Arnav Vaibhav Malawade, John Charles Chaput, William Grover, Philip Brisk, and Mohammad Abdullah Al Faruque. 2019. Oligo-Snoop: A Non-Invasive Side Channel Attack Against DNA Synthesis Machines. In *NDSS*.

[170] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *USENIX Security*.

[171] Aurore Fass, Michael Backes, and Ben Stock. 2019. HideNoSeek: Camouflaging Malicious JavaScript in Benign ASTs. In *ACM CCS*.

[172] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. 2019. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In *NDSS*.

[173] Qian Feng, Rundong Zhou, Chengcheng Xu, Yao Cheng, Brian Testa, and Heng Yin. 2016. Scalable Graph-based Bug Search for Firmware Images. In *ACM CCS*.

[174] Xuan Feng, Qiang Li, Haining Wang, and Limin Sun. 2018. Acquisitional Rule-based Engine for Discovering Internet-of-Thing Devices. In *USENIX Security*.

[175] Yu Feng, Osbert Bastani, Ruben Martins, Isil Dillig, and Saswat Anand. 2017. Automated Synthesis of Semantic Malware Signatures using Maximum Satisfiability. In *NDSS*.

[176] Hossein Fereidooni, Alexandra Dmitrienko, Phillip Rieger, Markus Miettinen, Ahmad-Reza Sadeghi, and Felix Madlener. 2022. FedCRI: Federated Mobile Cyber-Risk Intelligence. In *NDSS*.

[177] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A troubling analysis of reproducibility and progress in recommender systems research. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2021), 1–49.

[178] Felix Fischer, Konstantin Böttinger, Huang Xiao, Christian Stransky, Yasemin Acar, Michael Backes, and Sascha Fahl. 2017. Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security. In *IEEE S&P*.

[179] Felix Fischer, Yannick Stachelscheid, and Jens Grossklags. 2021. The Effect of Google Search on Software Security: Unobtrusive Security Interventions via Content Re-ranking. In *ACM CCS*.

[180] Felix Fischer, Huang Xiao, Ching-Yu Kao, Yannick Stachelscheid, Benjamin Johnson, Danial Raza, Paul Fawkesley, Nat Buckley, Konstantin Bottinger, Paul Muntean, and Jens Grossklags. 2019. Stack Overflow Considered Helpful! Deep Learning Security Nudges Towards Stronger Cryptography. In *USENIX Security*.

[181] David Formby, Preethi Srinivasan, AndrewM. Leonard, JonathanD. Rogers, and RaheemA. Beyah. 2016. Whos in Control of Your Control System? Device Fingerprinting for Cyber-Physical Systems. In *NDSS*.

[182] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In *ACM CCS*.

[183] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. 2014. Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing. In *USENIX Security*.

[184] Leonard P Freedman, Iain M Cockburn, and Timothy S Simcoe. 2015. The economics of reproducibility in preclinical research. *PLoS biology* 13, 6 (2015), e1002165.

[185] Sergey Frolov, Jack Wampler, and Eric Wustrow. 2020. Detecting Probe-resistant Proxies. In *NDSS*.

[186] Chuanpu Fu, Qi Li, Meng Shen, and Ke Xu. 2021. Realtime Robust Malicious Traffic Detection via Frequency Domain Analysis. In *ACM CCS*.

[187] Chenglong Fu, Qiang Zeng, and Xiaojiang Du. 2021. HAWatcher: Semantics-Aware Anomaly Detection for Appified Smart Homes. In *USENIX Security*.

[188] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, AlexX. Liu, and Ting Wang. 2022. Label Inference Attacks Against Vertical Federated Learning. In *USENIX Security*.

[189] Qi-An Fu, Yinpeng Dong, Hang Su, Jun Zhu, and Chao Zhang. 2022. AutoDA: Automated Decision-based Iterative Adversarial Attacks. In *USENIX Security*.

[190] Yuyou Gan, Yuhao Mao, Xuhong Zhang, Shouling Ji, Yuwen Pu, Meng Han, Jianwei Yin, and Ting Wang. 2022. Is your explanation stable?: A Robustness Evaluation Framework for Feature Attribution. In *ACM CCS*.

[191] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. 2018. Property Inference Attacks on Fully Connected Neural Networks using Permutation Invariant Representations. In *ACM CCS*.

[192] Haichang Gao, Wei Wang, Jiao Qi, Xuqin Wang, Xiyang Liu, and Jeff Yan. 2013. The Robustness of Hollow CAPTCHAs. In *ACM CCS*.

[193] Haichang Gao, Jeff Yan, Fang Cao, Zhengya Zhang, Lei Lei, Mengyun Tang, Ping Zhang, Xin Zhou, Xuqin Wang, and Jiawei Li. 2016. A Simple Generic Attack on Text Captchas. In *NDSS*.

[194] Peng Gao, Xusheng Xiao, Ding Li, Zhichun Li, Kangkook Jee, Zhenyu Wu, and Chung Hwan Kim. 2018. Saql: A Stream-based Query System for Real-Time Abnormal System Behavior Detection. In *USENIX Security*.

[195] Xianyi Gao, Yulong Yang, Can Liu, Christos Mitropoulos, and Janne Lindqvist. 2018. Forgetting of Passwords: Ecological Theory and Data. In *USENIX Security*.

[196] Yipeng Gao, Haichang Gao, Sainan Luo, Yang Zi, Shudong Zhang, Wenjie Mao, Ping Wang, Yulong Shen, and Jeff Yan. 2021. Research on the Security of Visual Reasoning CAPTCHA. In *USENIX Security*.

[197] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. 2018. AI2 : Safety and Robustness Certification of Neural Networks with Abstract Interpretation. In *IEEE S&P*.

[198] Daniel Genkin, Noam Nissan, Roei Schuster, and Eran Tromer. 2022. Lend Me Your Ear: Passive Remote Physical Side Channels on PCs. In *USENIX Security*.

[199] Daniel Genkin, Mihir Pattani, Roei Schuster, and Eran Tromer. 2019. Synesthesia: Detecting Screen Content via Remote Acoustic Side Channels. In *IEEE S&P*.

[200] Arthur Gervais, Reza Shokri, Adish Singla, Srdjan Capkun, and Vincent Lenders. 2014. Quantifying Web-Search Privacy. In *ACM CCS*.

[201] Jian Gong, Xinyu Zhang, Ju Ren, and Yaoxue Zhang. 2021. The Invisible Shadow: How Security Cameras Leak Private Activities. In *ACM CCS*.

[202] Xueluan Gong, Yanjiao Chen, Jianshuo Dong, and Qian Wang. 2022. ATTEQ-NN: Attention-based QoE-aware Evasive Backdoor Attacks. In *NDSS*.

[203] MariaPacheco Gonzales, Maxvon Hippel, Ben Weintraub, Dan Goldwasser, and Cristina Nita-Rotaru. 2022. Automated Attack Synthesis by Extracting Finite State Machines from Protocol Specification Documents. In *IEEE S&P*.

[204] José González Cabañas, Ángel Cuevas, and Rubén Cuevas. 2018. Unveiling and Quantifying Facebook Exploitation of Sensitive Personal Data for Advertising Purposes. In *USENIX Security*.

[205] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press, Cambridge, MA, USA.

[206] Ben Gras, Cristiano Giuffrida, Michael Kurth, Herbert Bos, and Kaveh Razavi. 2020. ABSynthe: Automatic Blackbox Side-channel Synthesis on Commodity Microarchitectures. In *NDSS*.

[207] Ben Gras, Kaveh Razavi, Herbert Bos, and Cristiano Giuffrid. 2018. Translation Leak-aside Buffer: Defeating Cache Side-channel Protections with TLB Attacks. In *USENIX Security*.

[208] Mariano Graziano, Davide Canali, Leyla Bilge, Andrea Lanzi, and Davide Balzarotti. 2015. Needles in a Haystack: Mining Information from Public Dynamic Analysis Sandboxes for Malware Intelligence. In *USENIX Security*.

[209] Rachel Greenstadt and Michael Bailey. 2021. Message from the USENIX Security'21 program co-chairs. In *30th USENIX Security Symposium, USENIX Security 2021*.

[210] Daniel Gruss, Julian Lettner, FelixSchusterOlyaOhrimenkoIstvan Haller, and Manuel Costa. 2017. Strong and Efficient Cache Side-Channel Protection using Hardware Transactional Memory. In *USENIX Security*.

[211] Odd Erik Gundersen, Yolanda Gil, and David W Aha. 2018. On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI magazine* 39, 3 (2018), 56–68.

[212] Odd Erik Gundersen and Sigbjørn Kjensmo. 2018. State of the art: Reproducibility in artificial intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[213] Hanqing Guo, Yuanda Wang, Nikolay Ivanov, Li Xiao, and Qiben Yan. 2022. SPECPATCH: Human-In-The-Loop Adversarial Audio Spectrogram Patch Attack on Speech Recognition. In *ACM CCS*.

[214] Wenbo Guo, Dongliang Mu, Jun Xu, Purui Su, Gang Wang, and Xinyu Xing. 2018. LEMNA: Explaining Deep Learning based Security Applications. In *ACM CCS*.

[215] Peter Hamm, David Harborth, and Sebastian Pape. 2019. A Systematic Analysis of User Evaluations in Security Research. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 1–7.

[216] Dianqi Han, Ang Li, Jiawei Li, Yan Zhang, Tao Li, and Yanchao Zhang. 2021. DroneKey: A Drone-Aided Group-Key Generation Scheme for Large-Scale IoT Networks. In *ACM CCS*.

[217] Dongqi Han, Zhiliang Wang, Wenqi Chen, Ying Zhong, Su Wang, Han Zhang, Jiahai Yang, Xingang Shi, and Xia Yin. 2021. DeepAID: Interpreting and Improving Deep Learning-based Anomaly Detection in Security Applications. In *ACM CCS*.

[218] Jun Han, AlbertJin Chung, Manal Kumar Sinha, Madhumitha Harishankar, Shijia Pan, HaeYoung Noh, Pei Zhang, and Patrick Tague. 2018. Do You Feel What I Hear? Enabling Autonomous IoT Device Pairing using Different Sensor Types. In *IEEE S&P*.

[219] Rebecca Han, Krista S Walton, and David S Sholl. 2019. Does chemical engineering research have a reproducibility problem? *Annual review of chemical and biomolecular engineering* 10 (2019), 43–57.

[220] Xueyuan Han, Xiao Yu, Thomas Pasquier, Ding Li, Junghwan Rhee, James Mickens, Margo Seltzer, and Haifeng Chen. 2021. SIGL: Securing Software Installations Through Deep Graph Learning. In *USENIX Security*.

[221] Yi Han, Matthew Chan, Zahra Aref, NilsOle Tippenhauer, and Saman Zonouz. 2022. Hiding in Plain Sight? On the Efficacy of Power Side Channel-Based Control Flow Monitoring. In *USENIX Security*.

[222] Yi Han, Sriharsha Etigowni, Hua Liu, SamanA. Zonouz, and AthinaP. Petropulu. 2017. Watch Me, but Dont Touch Me! Contactless Control Flow Monitoring via Electromagnetic Emanations. In *ACM CCS*.

[223] Qingying Hao, Licheng Luo, Steve T. K. Jan, and Gang Wang. 2021. It's Not What It Looks Like: Manipulating Perceptual Hashing based Applications. In *ACM CCS*.

[224] Hamza Harkous, Kassem Fawaz, Remi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. 2018. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. In *USENIX Security*.

[225] Hamza Harkous, Sai Teja Peddinti, Rishabh Khandelwal, Animesh Srivastava, and Nina Taft. 2022. Hark: A Deep Learning System for Navigating Privacy Feedback at Scale. In *IEEE S&P*.

[226] Rakibul Hasan, David Crandall, Mario Fritz, and Apu Kapadia. 2020. Automatically Detecting Bystanders in Photos to Reduce Privacy Risks. In *IEEE S&P*.

[227] Chaoxiang He, Bin Benjamin Zhu, Xiaojing Ma, Hai Jin, and Shengshan Hu. 2021. Feature-Indistinguishable Attack to Circumvent Trapdoor-Enabled Defense. In *ACM CCS*.

[228] Jingxuan He, Mislav Balunovi*f*á, Nodar Ambroladze, Petar Tsankov, and Martin Vechev. 2019. Learning to Fuzz from Symbolic Execution with Application to Smart Contracts. In *ACM CCS*.

[229] Jingxuan He, Persho Ivanov, Petar Tsankov, Veselin Raychev, and Martin Vechev. 2018. Debin: Predicting Debug Information in Stripped Binaries. In *ACM CCS*.

[230] Jingxuan He, Gishor Sivanrupan, Petar Tsankov, and Martin Vechev. 2021. Learning to Explore Paths for Symbolic Execution. In *ACM CCS*.

[231] Ruiwen He, Xiaoyu Ji, Xinfeng Li, Yushi Cheng, and Wenyuan Xu. 2022. "OK, Siri" or "Hey, Google": Evaluating Voiceprint Distinctiveness via Content-based PROLE Score. In *USENIX Security*.

[232] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing Links from Graph Neural Networks. In *USENIX Security*.

[233] Xinlei He and Yang Zhang. 2021. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In *ACM CCS*.

[234] Yingzhe He, Guozhu Meng, Kai Chen, Xingbo Hu, and Jinwen He. 2021. DrMi: A Dataset Reduction Technology based on Mutual Information for Black-box Attacks. In *USENIX Security*.

[235] Yuyu He, Lei Zhang, Zhemin Yang, Yinzhi Cao, Keke Lian, Shuai Li, Wei Yang, Zhibo Zhang, Min Yang, Yuan Zhang, and Haixin Duan. 2020. TextExerciser: Feedback-driven Text Input Exercising for Android Applications. In *IEEE S&P*.

[236] Kihong Heo, Woosuk Lee, Pardis Pashakhanloo, and Mayur Naik. 2018. Effective Program Debloating via Reinforcement Learning. In *ACM CCS*.

[237] Cormac Herley. 2022. Automated Detection of Automated Traffic. In *USENIX Security*.

[238] Nestor Hernandez, Mizanur Rahman, Ruben Recabarren, and Bogdan Carbunar. 2018. Fraud De-Anonymization For Fun and Profit. In *ACM CCS*.

[239] Thomas Herndon, Michael Ash, and Robert Pollin. 2014. Does high public debt consistently stifle economic growth? A critique of Reinhart and Rogoff. *Cambridge journal of economics* 38, 2 (2014), 257–279.

[240] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. 2017. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. In *ACM CCS*.

[241] Jordan Holland, Paul Schmitt, Nick Feamster, and Prateek Mittal. 2021. New Directions in Automated Traffic Analysis. In *ACM CCS*.

[242] Charlie Hou, Mingxun Zhou, Yan Ji, Phil Daian, Florian Tramer, Giulia Fanti, and Ari Juels. 2021. SquirRL: Automating Attack Analysis on Blockchain Incentive Mechanisms with Deep Reinforcement Learning. In *NDSS*.

[243] Aoting Hu, Renjie Xie, Zhigang Lu, Aiqun Hu, and Minhui Xue. 2021. TableGAN-MCA: Evaluating Membership Collisions of GAN-Synthesized Tabular Data Releasing. In *ACM CCS*.

[244] Hang Hu, Peng Peng, and Gang Wang. 2019. Characterizing Pixel Tracking through the Lens of Disposable Email Services. In *IEEE S&P*.

[245] Yiqing Hua, Armin Namavari, Kaishuo Cheng, Mor Naaman, and Thomas Ristenpart. 2022. Increasing Adversarial Uncertainty to Scale Private Similarity Testing. In *USENIX Security*.

[246] Hai Huang, Jiaming Mu, Neil Zhenqiang Gong, Qi Li, Bin Liu, and Mingwei Xu. 2021. Data Poisoning Attacks to Deep Learning Based Recommender Systems. In *NDSS*.

[247] Hai Huang, Zhikun Zhang, Yun Shen, Michael Backes, Qi Li, and Yang Zhang. 2022. On the Privacy Risks of Cell-Based NAS Architectures. In *ACM CCS*.

[248] Long Huang and Chen Wang. 2022. PCR-Auth: Solving Authentication Puzzle Challenge with Encoded Palm Contact Response. In *IEEE S&P*.

[249] Zhicong Huang, Erman Ayday, Jacques Fellay, Jean-Pierre Hubaux, and Ari Juels. 2015. GenoGuard: Protecting Genomic Data against Brute-Force Attacks. In *IEEE S&P*.

[250] Zhicong Huang, Wen-jie Lu, Cheng Hong, and Jiansheng Ding. 2022. Cheetah: Lean and Fast Secure Two-Party Deep Neural Network Inference. In *USENIX Security*.

[251] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. 2021. Practical Blind Membership Inference Attack via Differential Comparisons. In *NDSS*.

[252] SiamUmar Hussain, Mojan Javaheripi, Mohammad Samragh, and Farinaz Koushanfar. 2021. COINN: Crypto/ML Codesign for Oblivious Inference via Neural Networks. In *ACM CCS*.

[253] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2021. WaveGuard: Understanding and Mitigating Audio Adversarial Examples. In *USENIX Security*.

[254] Luca Invernizzi, Stanislav Miskovic, Ruben Torres, Christopher Kruegel, Sabyasachi Saha, Giovanni Vigna, Sung-Ju Lee, and Marco Mellia. 2014. Nazca: Detecting Malware Distribution in Large-Scale Networks. In *NDSS*.

[255] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. 2016. Cloak of Visibility: Detecting When Machines Browse a Different Web. In *IEEE S&P*.

[256] John P.A. Ioannidis. 2016. Why most clinical research is not useful. *PLoS Medicine* 13, 6 (2016), e1002049.

[257] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *IEEE S&P*.

[258] Umar Iqbal, Peter Snyder, Shitong Zhu, Benjamin Livshits, Zhiyun Qian, and Zubair Shafiq. 2020. ADGRAPH: A Graph-Based Approach to Ad and Tracker Blocking. In *IEEE S&P*.

[259] Umar Iqbal, Charlie Wolfe, Charles Nguyen, Steven Englehardt, and Zubair Shafiq. 2022. Khaleesi: Breaker of Advertising and Tracking Request Chains. In *USENIX Security*.

[260] Akira Ito, Rei Ueno, and Naofumi Homma. 2022. On the Success Rate of Side-Channel Attacks on Masked Implementations: Information-Theoretical Bounds and Their Practical Usage. In *ACM CCS*.

[261] Roger Iyengar, Om Thakkar, Joseph P. Near, Abhradeep Thakurta, Dawn Song, and Lun Wang. 2019. Towards Practical Differentially Private Convex Optimization. In *IEEE S&P*.

[262] Isaiah J. King and H. Howie Huang. 2022. EULER: Detecting Network Lateral Movement via Scalable Temporal Link Prediction. In *NDSS*.

[263] Arthur S. Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and LisandroZ. Granville. 2022. AI/ML for Network Security: The Emperor has no Clothes. In *ACM CCS*.

[264] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High Accuracy and High Fidelity Extraction of Neural Networks. In *USENIX Security*.

[265] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. In *IEEE S&P*.

[266] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. 2021. Subpopulation Data Poisoning Attacks. In *ACM CCS*.

[267] Nav Jagpal, Eric Dingle, Jean-Philippe Gravel, Panayiotis Mavrommatis, Niels Provos, Moheeb Abu Rajab, and Kurt Thomas. 2015. Trends and Lessons from Three Years Fighting Malicious Extensions. In *USENIX Security*.

[268] Shubham Jain, Ana-Maria Crețu, and Yves-Alexandrede Montjoye. 2022. Adversarial Detection Avoidance Attacks: Evaluating the robustness of perceptual hashing-based client-side scanning. In *USENIX Security*.

[269] Suman Jana, David Molnar, Alexander Moshchuk, Alan Dunn, Benjamin Livshits, Helen J. Wang, and Eyal Ofek. 2013. Enabling Fine-Grained Permissions for Augmented Reality Applications with Recognizers. In *USENIX Security*.

[270] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. 2013. A Scanner Darkly: Protecting User Privacy From Perceptual Applications. In *IEEE S&P*.

[271] Kai Jansen, Liang Niu, Nian Xue, Ivan Martinovic, and Christina Pöpper. 2021. Trust the Crowd: Wireless Witnessing to Detect Attacks on ADS-B-Based Air-Traffic Surveillance. In *NDSS*.

[272] Rob Jansen, Marc Juarez, Rafa Galvez, Tariq Elahi, and Claudia Diaz. 2018. Inside Job: Applying Traffic Analysis to Measure Tor from Within. In *NDSS*.

[273] Rob Jansen, Matthew Traudt, and Nicholas Hopper. 2018. Privacy-Preserving Dynamic Learning of Tor Network Traffic. In *ACM CCS*.

[274] Bargav Jayaraman and David Evans. 2022. Are Attribute Inference Attacks Just Imputation. In *ACM CCS*.

[275] Xiaoyu Ji, Yushi Cheng, Yuepeng Zhang, Kai Wang, Chen Yan, Wenyuan Xu, and Kevin Fu. 2021. Poltergeist: Acoustic Adversarial Machine Learning against

Cameras and Computer Vision. In *IEEE S&P*.
[276] Yujie Ji, Xinyang Zhang, Shouling Ji, Xiapu Luo, and Ting Wang. 2018. Model-Reuse Attacks on Deep Learning Systems. In *ACM CCS*.
[277] Hengrui Jia, ChristopherA. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. 2021. Entangled Watermarks as a Defense against Model Extraction. In *USENIX Security*.
[278] Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. 2021. Proof-of-Learning: Definitions and Practice. In *IEEE S&P*.
[279] Jinyuan Jia, Yupei Liu, and NeilZhenqiang Gong. 2022. BadEncoder: Backdoor Attacks to Pre-trained Encoders in Self-Supervised Learning. In *IEEE S&P*.
[280] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *ACM CCS*.
[281] Jinyuan Jia and Neil Zhenqiang Gong. 2018. AttriGuard: A Practical Defense Against Attribute Inference Attacks via Adversarial Machine Learning. In *USENIX Security*.
[282] Wei Jia, Zhaojun Lu, Haichun Zhang, Zhenglin Liu, Jie Wang, GangQuWei Jia, Zhaojun Lu, Haichun Zhang, Zhenglin Liu, Jie Wang, and Gang Qu. 2022. Fooling the Eyes of Autonomous Vehicles: Robust Physical Adversarial Examples Against Traffic Sign Recognition Systems. In *NDSS*.
[283] Xiaoqian Jiang, Miran Kim, Kristin Lauter, and Yongsoo Song. 2018. Secure Outsourced Matrix Computation and Application to Neural Networks. In *ACM CCS*.
[284] Jiankai Jin, Eleanor McMurtry, BenjaminI.P. Rubinstein, and Olga Ohrimenko. 2022. Are We There Yet? Timing and Floating-Point Attacks on Differential Privacy Systems. In *IEEE S&P*.
[285] Wenqiang Jin, Srinivasan Murali, Huadi Zhu, and Ming Li. 2021. Periscope: A Keystroke Inference Attack Using Human Coupled Electromagnetic Emanations. In *ACM CCS*.
[286] Xin Jin, Kexin Pei, JunYeon Won, and Zhiqiang Lin. 2022. SymLM: Predicting Function Names in Stripped Binaries via Context-Sensitive Execution-Aware Code Embeddings. In *ACM CCS*.
[287] Roberto Jordaney, Kumar Sharad, Santanu K. Dash, Zhi Wang, Davide Papini, Ilia Nouretdinov, and Lorenzo Cavallaro. 2017. Transcend: Detecting Concept Drift in Malware Classification Models. In *USENIX Security*.
[288] Matthew Joslin, Neng Li, Shuang Hao, Minhui Xue, and Haojin Zhu. 2019. Measuring and Analyzing Search Engine Poisoning of Linguistic Collisions. In *IEEE S&P*.
[289] Zeinab Joudaki, Julie Thorpe, and Miguel Vargas Martin. 2018. Reinforcing System-Assigned Passphrases Through Implicit Learning. In *ACM CCS*.
[290] Nikola Jovanović, Marc Fischer, Samuel Steffen, and Martin Vechev. 2022. Private and Reliable Neural Network Inference. In *ACM CCS*.
[291] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *ACM CCS*.
[292] Chiraag Juveka, Vinod Vaikuntanathan, and Anantha Chandrakasan. 2018. GAZELLE: A Low Latency Framework for Secure Neural Network Inference. In *USENIX Security*.
[293] Beliz Kaleli, Brian Kondracki, Manuel Egele, Nick Nikiforakis, and Gianluca Stringhini. 2021. To Err.Is Human: Characterizing the Threat of Unintended URLs in Social Media. In *NDSS*.
[294] George Kappos, Haaroon Yousaf, Rainer Stütz, Sofia Rollet, Bernhard Haslhofer, and Sarah Meiklejohn. 2022. How to Peel a Million: Validating and Expanding Bitcoin Clusters. In *USENIX Security*.
[295] Alexandros Kapravelos, Yan Shoshitaishvili, Marco Cova, Christopher Kruegel, and Giovanni Vigna. 2013. Revolver: An Automated Approach to the Detection of Evasive Web-based Malware. In *USENIX Security*.
[296] Gabriel Kaptchuk, Tushar M. Jois, Matthew Green, and Aviel D. Rubin. 2021. Meteor: Cryptographically Secure Steganography for Realistic Distributions. In *ACM CCS*.
[297] Mahimna Kelkar, PhiHung Le, Mariana Raykova, and Karn Seth. 2022. Secure Poisson Regression. In *USENIX Security*.
[298] MohammadTaha Khan, Christopher Tran, Shubham Singh, Dimitri Vasilkov, Chris Kanich, Blase Ur, and Elena Zheleva. 2021. Helping Users Automatically Find and Manage Sensitive, Expendable Files in Cloud Storage. In *USENIX Security*.
[299] Rishabh Khandelwal, Thomas Linden, and Hamza Harkous. 2021. PriSEC: A Privacy Settings Enforcement Controller. In *USENIX Security*.
[300] Amin Kharraz, William Robertson, and Engin Kirda. 2018. Surveylance: Automatically Detecting Online Survey Scams. In *IEEE S&P*.
[301] Taeri Kim, Noseong Park, Jiwon Hong, and Sang-Wook Kim. 2022. Phishing URL Detection: A Network-based Approach Robust to Evasion. In *ACM CCS*.
[302] Lucien K.L. Ng and Sherman S.M. Chow. 2021. GForce: GPU-Friendly Oblivious and Rapid Neural Network Inference. In *USENIX Security*.
[303] Marcel Kneib and Christopher Huth. 2018. Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Network. In *ACM CCS*.
[304] Marcel Kneib, Oleg Schell, and Christopher Huth. 2020. EASI: Edge-Based Sender Identification on Resource-Constrained Platforms for Automotive Networks. In *NDSS*.
[305] Aashish Kolluri, Teodora Baluta, Bryan Hooi, and Prateek Saxena. 2022. LPGNet: Link Private Graph Networks for Node Classification. In *ACM CCS*.
[306] Brian Kondracki, Babak Amin Azad, Najmeh Miramirkhani, and Nick Nikiforakis. 2022. The Droid is in the Details: Environment-aware Evasion of Android Sandboxes. In *NDSS*.
[307] Brian Kondracki, Babak Amin Azad, Oleksii Staro, and Nick Nikiforakis. 2021. Catching Transparent Phish: Analyzing and Detecting MITM Phishing Toolkits. In *ACM CCS*.
[308] Deguang Kong, Lei Cen, and Hongxia Jin. 2015. AUTOREB: Automatically Understanding the Review-to-Behavior Fidelity in Android Applications. In *ACM CCS*.
[309] Evgenios Kornaropoulos, Charalampos Papamanthou, and Roberto Tamassia. 2019. Data Recovery on Encrypted Databases With k-Nearest Neighbor Query Leakage. In *IEEE S&P*.
[310] Nishat Koti, Varsha Bhat Kukkala, Arpita Patra, and Bhavish Raj Gopal. 2022. PentaGOD: Stepping beyond Traditional GOD with Five Parties. In *ACM CCS*.
[311] Nishat Koti, Mahak Pancholi, Arpita Patra, and Ajith Suresh. 2021. SWIFT: Super-fast and Robust Privacy-Preserving Machine Learning. In *USENIX Security*.
[312] Thilo Krachenfels, Tuba Kiyan, Shahin Tajik, and Jean-Pierre Seifert. 2021. Automatic Extraction of Secrets from the Transistor Jungle using Laser-Assisted Side-Channel Attacks. In *USENIX Security*.
[313] Shriram Krishnamurthi. [n. d.]. About Artifact Evaluation. https://artifact-eval.org/about.html
[314] Alex Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images.
[315] Nishant Kumar, Mayank Rathee, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. 2020. CRYPTFLOW: Secure TensorFlow Inference. In *IEEE S&P*.
[316] Amit Kumar Sikder, Hidayet Aksu, and A. Selcuk Uluagac. 2017. 6thSense: A Context-aware Sensor-based Attack Detector for Smart Devices. In *USENIX Security*.
[317] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. In *USENIX Security*.
[318] Bum Jun Kwon, Jayanta Mondal, Jiyong Jang, Leyla Bilge, and Tudor Dumitraș. 2015. The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics. In *ACM CCS*.
[319] Yongin Kwon, Sangmin Lee, Hayoon Yi, Donghyun Kwon, Seungjun Yang, Byung-Gon Chun, Ling Huang, Petros Maniatis, Mayur Naik, and Yunheung Paek. 2013. Mantis: Automatic Performance Prediction for Smartphone Applications. In *USENIX Security*.
[320] Alexander Küchler, Alessandro Mantovani, Yufei Han, Leyla Bilge, and Davide Balzarotti. 2021. Does Every Second Count? Time-based Evolution of Malware Behavior in Sandboxes. In *NDSS*.
[321] Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. 2013. Measuring Password Guessability for an Entire University. In *ACM CCS*.
[322] Alexander S. La Cour, Khurram K. Afridi, and G. Edward Suh. 2021. Wireless Charging Power Side-Channel Attacks. In *ACM CCS*.
[323] Tomer Laor, Naif Mehanna, Antonin Durey, Vitaly Dyadyuk, Pierre Laperdrix, Clémentine Maurice, Yossi Oren, Romain Rouvoy, Walter Rudametkin, and Yuval Yarom. 2022. DRAWN APART: A Device Identification Technique based on Remote GPU Fingerprinting. In *NDSS*.
[324] Hieu Le, Athina Markopoulou, and Zubair Shafiq. 2021. CV-INSPECTOR: Towards Automating Detection of Adblock Circumvention. In *NDSS*.
[325] Stevens Le Blond, Cedric Gilbert, Utkarsh Upadhyay, Manuel Gomez Rodriguez, and David Choffnes. 2017. A Broad View of the Ecosystem of Socially Engineered Exploit Documents. In *NDSS*.
[326] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
[327] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. 2019. Certified Robustness to Adversarial Examples with Differential Privacy. In *IEEE S&P*.
[328] Sunwoo Lee, Wonsuk Choi, and Dong Hoon Lee. 2021. Usable User Authentication on a Smartwatch using Vibration. In *ACM CCS*.
[329] Suyoung Lee, HyungSeok Han, SangKil Cha, and Sooel Son. 2020. Montage: A Neural Network Language Model-Guided JavaScript Engine Fuzzer. In *USENIX Security*.
[330] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. 2017. Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing. In *USENIX Security*.
[331] Ryan Lehmkuhl, Pratyush Mishra, Akshayaram Srinivasan, and Raluca Ada Popa. 2021. Muse: Secure Inference Resilient to Malicious Clients. In *USENIX Security*.
[332] Klas Leino and Matt Fredrikson. 2020. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *USENIX Security*.

[333] Christopher Lentzsch, Sheel Jayesh Shah, Benjamin Andow, Martin Degeling, Anupam Das, and William Enck. 2021. Hey Alexa, is this Skill Safe?: Taking a Closer Look at the Alexa Skill Ecosystem. In *NDSS*.

[334] Changjiang Li, Li Wang, Shouling Ji, Xuhong Zhang, Zhaohan Xi, Shanqing Guo, and Ting Wang. 2022. Seeing is Living? Rethinking the Security of Facial Liveness Verification in the Deepfake Era. In *USENIX Security*.

[335] Huiying Li, Shawn Shan, Emily Wenger, Jiayun Zhang, Haitao Zheng, and Ben Y. Zhao. 2022. Blacklight: Scalable Defense for Neural Networks against Query-Based Black-Box Attacks. In *USENIX Security*.

[336] Huichen Li, Xiaojun Xu, Chang Liu, Teng Ren, Kun Wu, Xuezhi Cao, Weinan Zhang, Yong Yu, and Dawn Song. 2018. A Machine Learning Approach To Prevent Malicious Calls Over Telephony Networks. In *IEEE S&P*.

[337] Jinfeng Li, Tianyu Du, Shouling Ji, Rong Zhang, Quan Lu, Min Yang, and Ting Wang. 2020. TEXTSHIELD: Robust Text Classification Based on Multimodal Embedding and Neural Machine Translation. In *USENIX Security*.

[338] Jingjie Li, Kassem Fawaz, and Younghyun Kim. 2019. Velody: Nonlinear Vibration Challenge-Response for Resilient User Authentication. In *ACM CCS*.

[339] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. TEXTBUGGER: Generating Adversarial Text Against Real-world Applications. In *NDSS*.

[340] Jianfeng Li, Hao Zhou, Shuohan Wu, Xiapu Luo, Ting Wang, Xian Zhan, and Xiaobo Ma. 2022. FOAP: Fine-Grained Open-World Android App Fingerprinting. In *USENIX Security*.

[341] Linyi Li, Maurice Weber, Xiajun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. 2021. TSS: Transformation-Specific Smoothing for Robustness Certification. In *ACM CCS*.

[342] Lingjun Li, Xinxin Zhao, and Guoliang Xue. 2013. Unobservable Re-authentication for Smartphones. In *NDSS*.

[343] Shuai Li, Huajun Guo, and Nicholas Hopper. 2018. Measuring Information Leakage in Website Fingerprinting Attacks and Defenses. In *ACM CCS*.

[344] Shaofeng Li, Hui Liu, Tian Dong, Benjamin ZiHao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021. Hidden Backdoors in Human-Centric Language Models. In *ACM CCS*.

[345] Shasha Li, Ajaya Neupane, Sujoy Paul, Chengyu Song, Srikanth V. Krishnamurthy, Amit K. Roy Chowdhury, and Ananthram Swami. 2019. Stealthy Adversarial Perturbations Against Real-Time Video Classification Systems. In *NDSS*.

[346] Xuezixiang Li, Yu Qu, and Heng Yin. 2021. PalmTree: Learning an Assembly Language Model for Instruction Embedding. In *ACM CCS*.

[347] Yu Li, Min Li, Bo Luo, Ye Tian, and Qiang Xu. 2020. DeepDyve: Dynamic Verification for Deep Neural Networks. In *ACM CCS*.

[348] Yan Li, Yingjiu Li, Qiang Yan, Hancong Kong, and Robert H. Deng. 2015. Seeing Your Face Is Not Enough: An Inertial Sensor-Based Liveness Detection for Face Authentication. In *ACM CCS*.

[349] Yun Li, Cun Ye, Yuguang Hu, Ivring Morpheus, Yu Guo, Chao Zhang, Yupeng Zhang, Zhipeng Sun, Yiwen Lu, and Haodi Wang. 2021. ZKCPlus: Optimized Fair-exchange Protocol Supporting Practical and Flexible Data Exchange. In *ACM CCS*.

[350] Zhenyuan Li, Qi Alfred Chen, Chunlin Xiong, Yan Chen, Tiantian Zhu, and Hai Yang. 2019. Effective and Light-Weight Deobfuscation and Semantic-Aware Attack Detection for PowerShell Scripts. In *ACM CCS*.

[351] Zheng Li, Yiyong Liu, Xinlei He, Ning Yu, Michael Backes, and Yang Zhang. 2022. Auditing Membership Leakages of Multi-Exit Networks. In *ACM CCS*.

[352] Zhengxiong Li, Fenglong Ma, Aditya Singh Rathore, Zhuolin Yang, Baicheng Chen, Lu Su, and Wenyao Xu. 2020. WaveSpy: Remote and Through-wall Screen Attack via mmWave Sensing. In *IEEE S&P*.

[353] Ziyang Li, Aravind Machiry, Binghong Chen, Mayur Naik, Ke Wang, and Le Song. 2021. ARBITRAR: User-Guided API Misuse Detection. In *IEEE S&P*.

[354] Zhengxiong Li, Aditya Singh Rathore, Chen Song, Sheng Wei, Yanzhi Wang, and Wenyao Xu. 2018. PrinTracker: Fingerprinting 3D Printers using Commodity Scanners. In *ACM CCS*.

[355] Zhuohang Li, Cong Shi, Tianfang Zhang, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2021. Robust Detection of Machine-induced Audio Attacks in Intelligent Audio Systems with Microphone Array. In *ACM CCS*.

[356] Zhenhua Li, Weiwei Wang, Christo Wilson, Jian Chen, Chen Qian, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu. 2017. FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild. In *NDSS*.

[357] Zhuohang Li, Yi Wu, Jian Liu, and Yingying Chen. 2020. AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations. In *ACM CCS*.

[358] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *ACM CCS*.

[359] Zhen Li, Deqing Zou, Shouhuai Xu, Xinyu Ou, Hai Jin, Sujuan Wang, Zhijun Deng, and Yuyi Zhong. 2018. VulDeePecker: A Deep Learning-Based System for Vulnerability Detection. In *NDSS*.

[360] Junjie Liang, Wenbo Guo, Tongbo Luo, Vasant Honavar, Gang Wang, and Xinyu Xing. 2021. FARE: Enabling Fine-grained Attack Categorization under Low-quality Labeled Data. In *NDSS*.

[361] Sizhuang Liang, Saman A. Zonouz, and Raheem Beyah. 2022. Hiding My Real Self! Protecting Intellectual Property in Additive Manufacturing Systems Against Optical Side-Channel Attacks. In *NDSS*.

[362] Xiaojing Liao, Kan Yuan, XiaoFeng Wang, Zhongyu Pei, Hao Yang, Jianjun Chen, Haixin Duan, Kun Du3, Eihal Alowaisheq, Sumayah Alrwais, Luyi Xing, and Raheem Beyah. 2016. Seeking Nonsense, Looking for Trouble: Efficient Promotional-Infection Detection through Semantic Inconsistency Search. In *IEEE S&P*.

[363] Cynthia Liem and Annibale Panichella. 2020. Run, forest, run? on randomization and reproducibility in predictive software engineering. *arXiv preprint arXiv:2012.08387* (2020).

[364] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. 2020. Composite Backdoor Attack for Deep Neural Network by Mixing Existing Benign Features. In *ACM CCS*.

[365] Yun Lin, Ruofan Liu, DinilMon Divakaran, JunYang Ng, QingZhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and JinSong Dong. 2021. Phishpedia: A Hybrid Deep Learning Based Approach to Visually Identify Phishing Webpages. In *USENIX Security*.

[366] Xiang Ling, Shouling Ji, Jiaxu Zou, Jiannan Wang, Chunming Wu, Bo Li, and Ting Wang. 2019. DeepSec: A Uniform Platform for Security Analysis of Deep Learning Models. In *IEEE S&P*.

[367] Changchang Liu, Supriyo Chakraborty, and Prateek Mittal. 2016. Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples. In *NDSS*.

[368] Daiping Liu, Zhou Li, Kun Du, Haining Wang, Baojun Liu, and Haixin Duan. 2017. Don't Let One Rotten Apple Spoil the Whole Barrel: Towards Automated Detection of Shadowed Domains. In *ACM CCS*.

[369] Fucheng Liu, Yu Wen, Dongxue Zhang, Xihe Jiang, Xinyu Xing, and Dan Meng. 2019. Log2vec: A Heterogeneous Graph Embedding Based Approach for Detecting Cyber Threats within Enterprise. In *ACM CCS*.

[370] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. 2021. EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. In *ACM CCS*.

[371] Hongbin Liu, Jinyuan Jia, and Neil Zhenqiang Gong. 2022. PoisonedEncoder: Poisoning the Unlabeled Pre-training Data in Contrastive Learning. In *USENIX Security*.

[372] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. 2017. Oblivious Neural Network Predictions via MiniONN Transformations. In *ACM CCS*.

[373] Jiawei Liu, Kaisong Song, Yangyang Kang, Changlong Sun, Wei Lu, Xiaozhong Liu, Di Tang, and Xiaofeng Wang. 2022. Order-Disorder: Imitation Adversarial Attacks for Black-box Neural Ranking Models. In *ACM CCS*.

[374] Ruofan Liu, Yun Lin, Xianglin Yang, SiangHwee Ng, DinilMon Divakaran, and JinSong Dong. 2022. Inferring Phishing Intention via Webpage Appearance and Dynamics: A Deep Vision Based Approach. In *USENIX Security*.

[375] Shunchang Liu, Jiakai Wang, Aishan Liu, Yingwei Li, Yijie Gao, Xianglong Liu, and Dacheng Tao. 2022. Harnessing Perceptual Adversarial Patches for Crowd Counting. In *ACM CCS*.

[376] Tianyi Liu, Xiang Xie, and Yupeng Zhang. 2021. zkCNN: Zero Knowledge Proofs for Convolutional Neural Network Predictions and Accuracy. In *ACM CCS*.

[377] Xiangyu Liu, Zhe Zhou, Wenrui Diao, Zhou Li, and Kehuan Zhang. 2015. When Good Becomes Evil: Keystroke Inference with Smartwatch. In *ACM CCS*.

[378] Yushan Liu, Shouling Ji, and Prateek Mittal. 2016. SmartWalk: Enhancing Social Network Security via Adaptive Random Walks. In *ACM CCS*.

[379] Yupei Liu, Jinyuan Jia, Hongbin Liu, and Neil Zhenqiang Gong. 2022. StolenEncoder: Stealing Pre-trained Encoders in Self-supervised Learning. In *ACM CCS*.

[380] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. ABS: Scanning Neural Networks for Back-doors by Artificial Brain Stimulation. In *ACM CCS*.

[381] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. 2018. Trojaning Attack on Neural Networks. In *NDSS*.

[382] Yang Liu, Armin Sarabi, Jing Zhang, Manish Karir, Naghizadeh Parinaz, Michael Bailey, and Mingyan Liu. 2015. Cloudy with a Chance of Breach: Forecasting Cyber Security Incidents. In *USENIX Security*.

[383] Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. PICCOLO : Exposing Complex Backdoors in NLP Transformer Models. In *IEEE S&P*.

[384] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. 2022. ML-Doctor: Holistic Risk Assessment of Inference Attacks Against Machine Learning Models. In *USENIX Security*.

[385] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. 2022. Membership Inference Attacks by Exploiting Loss Trajectory. In *ACM CCS*.

[386] Zeyan Liu, Fengjun Li, Zhu Li, and Bo Luo. 2022. LoneNeuron: A Highly-Effective Feature-Domain Neural Trojan Using Invisible and Polymorphic Watermarks. In *ACM CCS*.

[387] Zhuoran Liu, Niels Samwel, Leo Weissbart, Zhengyu Zhao, Dirk Lauret, Lejla Batina, and Martha Larson. 2021. Screen Gleaning: A Screen Reading TEMPEST Attack on Mobile Devices Exploiting an Electromagnetic Side Channel. In *NDSS*.

[388] Giulio Lovisotto, Henry Turner, Ivo Sluganovic, Martin Strohmeier, and Ivan Martinovic. 2021. SLAP: Improving Physical Adversarial Examples with Short-Lived Adversarial Perturbations. In *USENIX Security*.

[389] Chaoyi Lu, Baojun Liu, Yiming Zhang, Zhou Li, Fenglu Zhang, Haixin Duan, Ying Liu, Joann Qiongna Chen, Jinjin Liang, Zaifeng Zhang, Shuang Hao, and Min Yang. 2021. From WHOIS to WHOWAS: A Large-Scale Measurement Study of Domain Registration Privacy under the GDPR. In *NDSS*.

[390] Wen-jie Lu, Zhicong Huang, Cheng Hong, Yiping Ma, and Hunter Qu. 2021. PEGASUS: Bridging Polynomial and Non-polynomial Evaluations in Homomorphic Encryption. In *IEEE S&P*.

[391] Matthew Luckie, Robert Beverly, Ryan Koga, Ken Keys, Joshua A. Kroll, and k claffy. 2019. Network Hygiene, Incentives, and Regulation: Deployment of Source Address Validation in the Internet. In *ACM CCS*.

[392] Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. 2022. SoK: How Robust is Image Classification Deep Neural Network Watermarking?. In *IEEE S&P*.

[393] Shiqing Luo, Anh Nguyen, Chen Song, Feng Lin, Wenyao Xu, and Zhisheng Yan. 2020. OcuLock: Exploring Human Visual System for Authentication in Virtual Reality Head-mounted Display. In *NDSS*.

[394] Tao Lv, Ruishi Li, Yi Yang, Kai Chen, Xiaojing Liao, XiaoFeng Wang, Peiwei Hu, and Luyi Xing. 2020. RTFM! Automatic Assumption Discovery and Verification Derivation from Library Document for API Misuse Detection. In *ACM CCS*.

[395] Yunlong Lyu, Yi Fang, Yiwei Zhang, Qibin Sun, Siqi Ma, Elisa Bertino, Kangjie Lu, and Juanru Li. 2022. Goshawk: Hunting Memory Corruptions via Structure-Aware and Object-Centric Memory Operation Synopsis. In *IEEE S&P*.

[396] Mathias Lécuyer, Riley Spahn, Yannis Spiliopolous, Augustin Chaintreau, Roxana Geambasu, and Daniel J. Hsu. 2015. Sunlight: Fine-grained Targeting Detection at Scale with Statistical Confidence. In *ACM CCS*.

[397] Jared M. Smith, Kyle Birkeland, Tyler McDaniel, and Max Schuchard. 2020. Withdrawing the BGP Re-Routing Curtain: Understanding the Security Impact of BGP Poisoning via Real-World Measurements. In *NDSS*.

[398] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. 2014. A Study of Probabilistic Password Models. In *IEEE S&P*.

[399] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. 2019. NIC: Detecting Adversarial Samples with Neural Network Invariant Checking. In *NDSS*.

[400] Xinyu Tang Saeed Mahloujifar, Liwei Song, Milad Nasr, Virat Shejwalkar, Amir Houmansadr, and Prateek Mittal. 2022. Mitigating Membership Inference Attacks by Self-Distillation Through a Novel Ensemble Architecture. In *USENIX Security*.

[401] Mohammad Malekzadeh, Anastasia Borovykh, and Deniz Gündüz. 2021. Honest-but-Curious Nets: Sensitive Attributes of Private Inputs Can Be Secretly Coded into the Classifiers' Outputs. In *ACM CCS*.

[402] David Mandell Freeman, Markus Durmuth, Battista Biggio, and Giorgio Giacinto. 2016. Who Are You? A Statistical Approach to Measuring User Authenticity. In *NDSS*.

[403] Alessandro Mantovani, Simone Aonzo, Xabier Ugarte-Pedrero, Alessio Merlo, and Davide Balzarotti. 2020. Prevalence and Impact of Low-Entropy Packing Schemes in the Malware Ecosystem. In *NDSS*.

[404] Andrea Marcelli, Mariano Graziano, Xabier Ugarte-Pedrero, Yanick Fratantonio, Mohamad Mansouri, and Davide Balzarotti. 2022. How Machine Learning Is Solving the Binary Function Similarity Problem. In *USENIX Security*.

[405] Enrico Mariconti, Lucky Onwuzurike, Panagiotis Andriotis, Emiliano De Cristofaro, Gordon Ross, and Gianluca Stringhini. 2017. MaMaDroid: Detecting Android Malware by Building Markov Chains of Behavioral Models. In *ACM CCS*.

[406] Clémentine Maurice and Cristiano Giuffrida. 2022. Artifact Appendices to the Proceedings of the 31st USENIX Security Symposium. In *31st USENIX Security Symposium, USENIX Security 2022*.

[407] Chris McMahon Stone, Sam L. Thomas, Mathy Vanhoef, James Henderson, Nicolas Bailluet, and Tom Chothia. 2022. The Closer You Look, The More You Learn: A Grey-box Approach to Protocol State Machine Learning. In *ACM CCS*.

[408] Shagufta Mehnaz, Sayanton V. Dibbo, Ehsanul Kabir, Ninghui Li, and Elisa Bertino. 2022. Are Your Sensitive Attributes Private? Novel Model Inversion Attribute Inference Attacks on Classification Models. In *USENIX Security*.

[409] William Melicher, Blase Ur, SeanM. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In *USENIX Security*.

[410] Luca Melis, George Danezis, and Emiliano De Cristofaro. 2016. Efficient Private Statistics with Succinct Sketches. In *NDSS*.

[411] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE S&P*.

[412] Dongyu Meng and Hao Chen. 2017. MagNet: a Two-Pronged Defense against Adversarial Examples. In *ACM CCS*.

[413] Tey Chee Meng, Payas Gupta, and Debin Gao. 2013. I can be You: Questioning the use of Keystroke Dynamics as Biometrics. In *NDSS*.

[414] Wei Meng, Ren Ding, SimonP. Chung, Steven Han, and Wenke Lee. 2016. The Price of Free: Privacy Leakage in Personalized Mobile In-App Adds. In *NDSS*.

[415] Yan Meng, Jiachun Li, Matthew Pillari, Arjun Deopujari, Liam Brennan, Hafsah Shamsie, Haojin Zhu, and Yuan Tian. 2022. Your Microphone Array Retains Your Identity: A Robust Voice Liveness Detection System for Smart Speakers. In *USENIX Security*.

[416] Xianghang Mi, Xuan Feng, Xiaojing Liao, Baojun Liu, XiaoFeng Wang, Feng Qian, Zhou Li, Sumayah Alrwais, Limin Sun, and Ying Liu. 2019. Resident Evil: Understanding Residential IP Proxy as a Dark Service. In *IEEE S&P*.

[417] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing Speech From Gyroscope Signals. In *USENIX Security*.

[418] Yan Michalevsky, Aaron Schulman, Gunaa Arumugam Veerapian, Dan Bonehy, and Gabi Nakibly. 2015. PowerSpy: Location Tracking Using Mobile Device Power Analysis. In *USENIX Security*.

[419] Mohsen Minaei, S Chandra Mouli, Mainack Mondal, Bruno Ribeiro, and Aniket Kate. 2021. Deceptive Deletions for Protecting Withdrawn Posts on Social Media Platforms. In *NDSS*.

[420] Jaron Mink, Licheng Luo, NatãM. Barbosa, Olivia Figueira, Yang Wang, and Gang Wang. 2022. DeepPhish: Understanding User Trust Towards Artificially Generated Profiles in Online Social Networks. In *USENIX Security*.

[421] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. In *NDSS*.

[422] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. 2020. DELPHI: A Cryptographic Inference Service for Neural Networks. In *USENIX Security*.

[423] Meisam Mohammady, Shangyu Xie, Yuan Hong, Mengyuan Zhang, Lingyu Wang, Makan Pourzandi, and Mourad Debbabi. 2020. R2DP: A Universal and Automated Approach to Optimizing the Randomization Mechanisms of Differential Privacy for Utility Metrics with No Known Optimal Distributions. In *ACM CCS*.

[424] Payman Mohassel and Peter Rindal. 2018. ABY3 : A Mixed Protocol Framework for Machine Learning. In *ACM CCS*.

[425] Payman Mohassel and Yupeng Zhang. 2017. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *IEEE S&P*.

[426] John V. Monaco. 2022. Device Fingerprinting with Peripheral Timestamps. In *IEEE S&P*.

[427] Mainack Mondal, Günce Su Yılmaz, Noah Hirsch, Mohammad Taha Khan, Michael Tang, Christopher Tran, Chris Kanich, Blase Ur, and Elena Zheleva. 2019. Moving Beyond Set-It-And-Forget-It Privacy Settings on Social Media. In *ACM CCS*.

[428] Jiaming Mu, Binghui Wang, Qi Li, Kun Sun, Mingwei Xu, and Zhuotao Liu. 2021. A Hard Label Black-box Adversarial Attack Against Graph Neural Networks. In *ACM CCS*.

[429] Raymond Muller, Yanmao Man, Z. Berkay Celik, Ming Li, and Ryan M. Gerdes. 2022. Physical Hijacking Attacks against Object Trackers. In *ACM CCS*.

[430] MulongLuoAndrewC. Myers and G.Edward Suh. 2020. Stealthy Tracking of Autonomous Vehicles with Cache Side Channels. In *USENIX Security*.

[431] Hoda Naghibijouybari, Ajaya Neupane, Zhiyun Qian, and Nael Abu-Ghazaleh. 2018. Rendered Insecure: GPU Side Channel A!acks are Practical. In *ACM CCS*.

[432] Yuhong Nan, Min Yang, Zhemin Yang, Shunfan Zhou, Guofei Gu, and XiaoFeng Wang. 2015. UIPicker: User-Input Privacy Identification in Mobile Applications. In *USENIX Security*.

[433] Yuhong Nan, Zhemin Yang, Xiaofeng Wang, Yuan Zhang, Donglai Zhu, and Min Yang. 2018. Finding Clues for Your Secrets: Semantics-Driven, Learning-Based Privacy Discovery in Mobile Apps. In *NDSS*.

[434] Faraz Naseem, Ahmet Aris, Leonardo Babun, Ege Tekiner, and A. Selcuk Uluagac. 2021. MINOS*: A Lightweight Real-Time Cryptojacking Detection System. In *NDSS*.

[435] Mohammad Naseri, Yufei Han, Enrico Mariconti, Yun Shen, Gianluca Stringhini, and Emiliano De Cristofaro. 2022. CERBERUS: Exploring Federated Prediction of Security Events. In *ACM CCS*.

[436] Mohammad Naseri, Jamie Hayes, and EmilianoDe Cristofaro. 2022. Local and Central Differential Privacy for Robustness and Privacy in Federated Learning. In *NDSS*.

[437] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. 2018. DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In *ACM CCS*.

[438] Milad Nasr, Alireza Bahramali, and Amir Houmansadr. 2021. Defeating DNN-Based Traffic Analysis Systems in Real-Time With Blind Adversarial Perturbations. In *USENIX Security*.

[439] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *ACM CCS*.

[440] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning. In *IEEE S&P*.

[441] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. 2021. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. In *IEEE S&P*.

[442] Ben Nassi, Yisroel Mirsky, Dudi Nassi, Raz Ben-Netanel, Oleg Drokin, and Yuval Elovici. 2020. Phantom of the ADAS: Securing Advanced Driver-Assistance Systems from Split-Second Phantom Attacks. In *ACM CCS*.

[443] National Academies of Sciences Engineering and Medicine and others. 2019. *Reproducibility and replicability in science.* National Academies Press.

[444] Kartik Nayak, XiaoShaun Wang, Stratis Ioannidis, Udi Weinsberg, Nina Taft, and Elaine Shi. 2015. GraphSC: Parallel Secure Computation Made Easy. In *IEEE S&P*.

[445] Parimarjan Negi, Prafull Sharma, Vivek Sanjay Jain, and Bahman Bahmani. 2018. K-means++ vs. Behavioral Biometrics: One Loop to Rule Them All. In *NDSS*.

[446] Terry Nelms, Roberto Perdisci, and Mustaque Ahamad. 2013. ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates. In *USENIX Security*.

[447] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad. 2015. WebWitness: Investigating, Categorizing, and Mitigating Malware Download Paths. In *USENIX Security*.

[448] Ajaya Neupane, Nitesh Saxena, Leanne Hirshfield, and Sarah Elaine Bratt. 2019. The Crux of Voice (In)Security: A Brain Study of Speaker Legitimacy Detection. In *NDSS*.

[449] Ajaya Neupane, Nitesh Saxena, Keya Kuruvilla, Michael Georgescu, and RajeshK. Kana. 2014. Neural Signatures of User-Centered Security: An fMRI Study of Phishing, and Malware Warnings. In *NDSS*.

[450] Duc Cuong Nguyen, Erik Derr, Michael Backes, and Sven Bugiel. 2019. Short Text, Large Effect: Measuring the Impact of User Reviews on Android App Security & Privacy. In *IEEE S&P*.

[451] TrungTin Nguyen, Michael Backes, and Ben Stock. 2022. Freely Given Consent?: Studying Consent Notice of Third-Party Tracking and Its Violations of GDPR in Android Apps. In *ACM CCS*.

[452] Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. 2013. Privacy-Preserving Ridge Regression on Hundreds of Millions of Records. In *IEEE S&P*.

[453] Shirin Nilizadeh, Francois Labreche, Alireza Sedighian, Ali Zand, JoséM. Fernandez, Christopher Kruegel, Gianluca Stringhini, and Giovanni Vigna. 2017. POISED: Spotting Twitter Spam Off the Beaten Paths. In *ACM CCS*.

[454] Guevara Noubir. 2016. Reproducibility in wireless experimentation: need, challenges, and approaches. In *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization.* 1–1.

[455] Sean Oesch and Scott Ruoti. 2020. That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers. In *USENIX Security*.

[456] SeEun Oh, Taiji Yang, Nate Mathews, James K. Holland, Mohammad Saidur Rahman, Nicholas Hopper, and Matthew Wright. 2022. DeepCoFFEA: Improved Flow Correlation Attacks on Tor via Metric Learning and Amplification. In *IEEE S&P*.

[457] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. 2016. Oblivious Multi-Party Machine Learning on Trusted Processors. In *USENIX Security*.

[458] Katarzyna Olejnik, Italo Dacosta, Joana Soares Machado, Kevin Huguenin, Mohammad Emtiyaz Khan, and Jean-Pierre Hubaux. 2017. SmarPer: Context-Aware and Automatic Runtime-Permissions for Mobile Devices. In *IEEE S&P*.

[459] Tripp Omer, Salvatore Guarnieri, Marco Pistoia, and Aleksandr Aravkin. 2014. ALETHEIA: Improving the Usability of Static Security Analysis. In *ACM CCS*.

[460] Jonathan P. Chapman. 2018. SAD THUG: Structural Anomaly Detection for Transmissions of High-value Information Using Graphics. In *USENIX Security*.

[461] Riccardo Paccagnella, Licheng Luo, and Christopher W. Fletcher. 2021. Lord of the Ring(s): Side Channel Attacks on the CPU On-Chip Ring Interconnect Are Practical. In *USENIX Security*.

[462] Bijeeta Pal, Tal Daniel, Rahul Chatterjee, and Thomas Ristenpart. 2019. Beyond Credential Stuffing: Password Similarity Models using Neural Networks. In *IEEE S&P*.

[463] Xiang Pan, Yinzhi Cao, Xuechao Du, Boyuan He, Gan Fang, and Yan Chen. 2018. FlowCog: Context-aware Semantics Extraction and Analysis of Information Flow Leaks in Android Apps. In *USENIX Security*.

[464] Xiaorui Pan, Xueqiang Wang, Yue Duan, XiaoFeng Wang, and Heng Yin. 2017. Dark Hazard: Learning-based, Large-Scale Discovery of Hidden Sensitive Operations in Android Apps. In *NDSS*.

[465] Xudong Pan, Mi Zhang, Shouling Ji, and Min Yang. 2020. Privacy Risks of General-Purpose Language Models. In *IEEE S&P*.

[466] Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. Hidden Trigger Backdoor Attack on NLP Models via Linguistic Style Manipulation. In *USENIX Security*.

[467] Xudong Pan, Mi Zhang, Duocai Wu, Qifan Xiao, Shouling Ji, and Min Yang. 2020. Justinian's GAAvernor: Robust Distributed Learning with Gradient Aggregation Agent. In *USENIX Security*.

[468] Xudong Pan, Mi Zhang, Yifan Yan, Jiaming Zhu, and Min Yang. 2022. Exploring the Security Boundary of Data Reconstruction via Neuron Exclusivity Analysis. In *USENIX Security*.

[469] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. 2016. Website Fingerprinting at Internet Scale. In *NDSS*.

[470] Sharbani Pandit, Roberto Perdisci, Mustaque Ahamad, and Payas Gupta. 2018. Towards Measuring the Effectiveness of Telephony Blacklists. In *NDSS*.

[471] Rahul Pandita, Xusheng Xiao, Wei Yang, William Enck, and Tao Xie. 2013. WHYPER: Towards Automating Risk Assessment of Mobile Applications. In *USENIX Security*.

[472] Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex X. Liu, and Ting Wang. 2020. A Tale of Evil Twins: Adversarial Inputs versus Poisoned Models. In *ACM CCS*.

[473] Ren Pang, Zhaohan Xi, Shouling Ji, Xiapu Luo, and Ting Wang. 2022. On the Security Risks of AutoML. In *USENIX Security*.

[474] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2016. Distillation as a Defense to Adversarial Pertubations against Deep Neural Networks. In *IEEE S&P*.

[475] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. 2021. Unleashing the Tiger: Inference Attacks on Split Learning. In *ACM CCS*.

[476] Dario Pasquini, Marco Cianfriglia, Giuseppe Ateniese, and Massimo Bernaschi. 2021. Reducing Bias in Modeling Real-world Password Strength via Deep Learning and Dynamic Dictionaries. In *USENIX Security*.

[477] Dario Pasquini, Ankit Gangwal, Giuseppe Ateniese, Massimo Bernaschi, and Mauro Conti. 2021. Improving Password Guessing via Representation Learning. In *IEEE S&P*.

[478] Arpita Patra, Thomas Schneider, Ajith Suresh, and Hossein Yalame. 2021. ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation. In *USENIX Security*.

[479] Arpita Patra and Ajith Suresh. 2020. BLAZE: Blazing Fast Privacy-Preserving Machine Learning. In *NDSS*.

[480] Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. 2018. Sonar: Detecting SS7 Redirection Attacks With Audio-Based Distance Bounding. In *IEEE S&P*.

[481] Kexin Pei, Jonas Guan, David Williams-King, Junfeng Yang, and Suman Jana. 2021. XDA: Accurate, Robust Disassembly with Transfer Learning. In *NDSS*.

[482] Feargus Pendlebury, Fabio Pierazzi, Roberto Jordaney, Johannes Kinder, and Lorenzo Cavallaro. 2019. TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time. In *USENIX Security*.

[483] Henning Perl, Sergej Dechand, Matthew Smith, Daniel Arp, Fabian Yamaguchi, Konrad Rieck, Sascha Fahl, and Yasemin Acar. 2015. VCCFinder: Finding Potential Vulnerabilities in Open-Source Projects to Assist Code Audits. In *ACM CCS*.

[484] Fabio Pierazzi, Feargus Pendlebury, Jacopo Cortellazzi, and Lorenzo Cavallaro. 2020. Intriguing Properties of Adversarial ML Attacks in the Problem Space. In *IEEE S&P*.

[485] Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. 2021. Improving reproducibility in machine learning research (a report from the neurips 2019 reproducibility program). *The Journal of Machine Learning Research* 22, 1 (2021), 7459–7478.

[486] Jack P.K. Ma, Yongjun Zhao, Tai Raymond K.H., Ahmet Aris, and Sherman S.M. Chow. 2021. Let's Stride Blindfolded in a Forest: Sublinear Multi-Client Decision Trees Evaluation. In *NDSS*.

[487] Victo rLe Pochat, Tim Van hamme, Sourena Maroofi, Tom Van Goethem, Davy Preuveneers, Andrzej Duda, Wouter Joosen, and Maciej Korczyński. 2020. A Practical Approach for Taking Down Avalanche Botnets Under Real-World Constraints. In *NDSS*.

[488] Rishabh Poddar, Ganesh Ananthanarayanan, Srinath Setty, Stavros Volos, and Raluca Ada Popa. 2020. Visor: Privacy-Preserving Video Analytics as a Cloud Service. In *USENIX Security*.

[489] Sathvik Prasad, Elijah Bouma-Sims, Athishay Kiran Mylappan, and Bradley Reaves. 2020. Who's Calling? Characterizing Robocalls through Audio and Metadata Analysis. In *USENIX Security*.

[490] Muhammad Qasim Ali and Ehab Al-Shaer. 2013. Configuration-based IDS for Advanced Metering Infrastructure. In *ACM CCS*.

[491] Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. 2014. AutoCog: Measuring the Description-to-permission Fidelity in Android Applications. In *ACM CCS*.

[492] Raul Quinonez, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. 2020. SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants. In *USENIX Security*.

[493] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. 2020. Adversarial Preprocessing: Understanding and Preventing Image-Scaling Attacks in Machine Learning. In *USENIX Security*.

[494] Viet Quoc Vo, Ehsan Abbasnejad, and Damith C. Ranasinghe. 2022. RamBoAttack: A Robust Query Efficient Deep Neural Network Decision Exploit. In *NDSS*.

[495] Edward Raff. 2019. A step toward quantifying independently reproducible machine learning research. *Advances in Neural Information Processing Systems* 32 (2019).

[496] Wullianallur Raghupathi, Viju Raghupathi, and Jie Ren. 2022. Reproducibility in computing research: An empirical study. *IEEE Access* 10 (2022), 29207–29223.

[497] Moheeb Abu Rajab, Lucas Ballard, Noe Lutz, Panayiotis Mavrommatis, and Niels Provos. 2013. CAMP: Content-Agnostic Malware Protection. In *NDSS*.

[498] Adnan Siraj Rakin, Md Hafizul Islam Chowdhuryy, Fan Yao, and Deliang Fan. 2022. DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories. In *IEEE S&P*.

[499] RamSundara Raman, Adrian Stoll, Jakub Dalek, Reethika Ramesh, Will Scott, and Roya Ensafi. 2020. Measuring the Deployment of Network Censorship Filters at Global Scale. In *NDSS*.

[500] Sivaramakrishnan Ramanathan, Jelena Mirkovic, and Minlan Yu. 2020. BLAG: Improving the Accuracy of Blacklists. In *NDSS*.

[501] KasperBonne Rasmussen, Marc Roeschlin, Ivan Martinovic, and Gene Tsudik. 2014. Authentication Using Pulse-Response Biometrics. In *NDSS*.

[502] Deevashwer Rathe, Mayank Rathee, Nishant Kumar, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. 2020. CrypTFlow2: Practical 2-Party Secure Inference. In *ACM CCS*.

[503] Deevashwer Rathee, Mayank Rathee, Rahul Kranti Kiran Goli, Divya Gupta, Rahul Sharma, Nishanth Chandran, and Aseem Rastogi. 2021. SIRNN: A Math Library for Secure RNN Inference. In *IEEE S&P*.

[504] ElissaM. Redmiles, Ziyun Zhu, Sean Kross, Dhruv Kuchhal, Tudor Dumitras, and Michelle L. Mazurek. 2018. Asking for a Friend: Evaluating Response Biases in Security User Studies. In *ACM CCS*.

[505] M. Sadegh Riazi, Mohammad Samragh, Hao Chen, Kim Laine, Kristin Lauter, and Farinaz Koushanfar. 2019. Xonn: XNOR-based Oblivious Deep Neural Network Inference. In *USENIX Security*.

[506] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. 2022. DeepSight: Mitigating Backdoor Attacks in Federated Learning Through Deep Model Inspection. In *NDSS*.

[507] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. 2018. Automated Website Fingerprinting through Deep Learning. In *NDSS*.

[508] Thomas Roche, Victor Lomné, Camille Mutschler, and Laurent Imbert. 2021. A Side Journey To Titan Revealing and Breaking NXP's P5x ECDSA Implementation on the Way. In *USENIX Security*.

[509] Jose Rodrigo Sanchez Vicarte, Gang Wang, and Christopher W. Fletcher. 2021. Double-Cross Attacks: Subverting Active Learning Systems. In *USENIX Security*.

[510] Marc Roeschlin, Ivan Martinovic, and Kasper Bonne Rasmussen. 2018. Device Pairing at the Touch of an Electrode. In *NDSS*.

[511] Marco Romanelli, Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Pablo Piantanida. 2020. Estimating g-Leakage via Machine Learning. In *ACM CCS*.

[512] Marc B. Rosen, James Parker, and AlexJ. Malozemoff. 2021. Balboa: Bobbing and Weaving around Network Censorship. In *USENIX Security*.

[513] Nicolás Rosner, Ismet Burak Kadron, Lucas Bang, and Tevfik Bultan. 2019. Profit: Detecting and Quantifying Side Channels in Networked Applications. In *NDSS*.

[514] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on Layer Two. In *IEEE S&P*.

[515] Carl Sabottke, Octavian Suciu, and Tudor Dumitras. 2015. Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. In *USENIX Security*.

[516] Mohd Sabra, Anindya Maiti, and Murtuza Jadliwala. 2021. Zoom on the Keystrokes: Exploiting Video Calls for Keystroke Inference Attacks. In *NDSS*.

[517] Merve Sahin and Aurélien Francillon. 2021. Understanding and Detecting International Revenue Share Fraud. In *NDSS*.

[518] Sena Sahin and Frank Li. 2021. Don't Forget the Stuffing! Revisiting the Security Impact of Typo-Tolerant Password Authentication. In *ACM CCS*.

[519] Sina Sajadmanesh and Daniel Gatica-Perez. 2021. Locally Private Graph Neural Networks. In *ACM CCS*.

[520] Nazir Saleheen, Md Azim Ullah, Supriyo Chakraborty, Deniz S. Ones, Mani Srivastava, and Santosh Kumar. 2021. WristPrint: Characterizing User Re-identification Risks from Wrist-worn Accelerometry Data. In *ACM CCS*.

[521] Ahmed Salem, Michael Backes, and Yang Zhang. 2022. Get a Model! Model Hijacking Attack Against Machine Learning Models. In *NDSS*.

[522] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *USENIX Security*.

[523] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *NDSS*.

[524] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jia, Xue Lin, and Qi Alfred Chen. 2021. Dirty Road Can Attack: Security of Deep Learning based Automated Lane Centering under Physical-World Attack. In *USENIX Security*.

[525] Sinem Sav, Apostolos Pyrgelis, Juan Ramón Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, JoaoSa Sousa, and Jean-Pierre Hubaux. 2021. POSEIDON: Privacy-Preserving Federated Neural Network Learning. In *NDSS*.

[526] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. 2014. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*. IEEE, 83–94.

[527] Stephanvan Schaik, Marina Minkin, Andrew Kwong, Daniel Genkin, and Yuval Yarom. 2021. CacheOut: Leaking Data on Intel CPUs via Cache Evictions. In *IEEE S&P*.

[528] Lea Schonherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial Attacks Against Automatic Speech Recognition Systems via Psychoacoustic Hiding. In *NDSS*.

[529] Phillipp Schoppmann, Adrià Gascón, Mariana Raykova, and Benny Pinkas. 2019. Make Some ROOM for the Zeros: Data Sparsity in Secure Distributed Machine Learning. In *ACM CCS*.

[530] Samuel Schuppen, Dominik Teubert, and Patrick Herrmann. 2018. FANCI : Feature-based Automated NXDomain Classification and Intelligence. In *USENIX Security*.

[531] Roei Schuster, Vitaly Shmatikov, and Eran Tromer. 2017. Beauty and the Burst: Remote Identification of Encrypted Video Streams. In *USENIX Security*.

[532] Roei Schuster, Congzheng Song, Eran Tromer, and Vitaly Shmatikov. 2021. You Autocomplete Me: Poisoning Vulnerabilities in Neural Code Completion. In *USENIX Security*.

[533] Asuman Senol, Gunes Acar, Mathias Humbert, and Frederik Zuiderveen Borgesius. 2022. Leaky Forms: A Study of Email and Password Exfiltration Before Form Submission. In *USENIX Security*.

[534] Abdul Serwadda and Vir V. Phoha. 2013. When Kids' Toys Breach Mobile Phone Security. In *ACM CCS*.

[535] Giorgio Severi, Jim Meyer, Scott Coull, and Alina Oprea. 2021. Explanation-Guided Backdoor Poisoning Attacks Against Malware Classifiers. In *USENIX Security*.

[536] Shawn Shan, Wenxin Ding, Emily Wenger, Haitao Zheng, and Ben Y. Zhao. 2022. Post-breach Recovery: Protection against White-box Adversarial Examples for Leaked DNN Models. In *ACM CCS*.

[537] Shawn Shan, Arjun Nitin Bhagoji, Haitao Zheng, and Ben Y. Zhao. 2022. Poison Forensics: Traceback of Data Poisoning Attacks in Neural Networks. In *USENIX Security*.

[538] Shawn Shan, Emily Wenger, Bolun Wang, Bo Li, Haitao Zheng, and Ben Y. Zhao. 2020. Gotta Catch'Em All: Using Honeypots to Catch Adversarial Attacks on Neural Networks. In *ACM CCS*.

[539] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2020. Fawkes: Protecting Privacy against Unauthorized Deep Learning Models. In *USENIX Security*.

[540] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and MichaelK. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *ACM CCS*.

[541] Mahmood Sharif, Jumpei Urakawa, Nicolas Christin, Ayumu Kubota, and Akira Yamada. 2018. Predicting Impending Exposure to Malicious Content from User Behavior. In *ACM CCS*.

[542] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. 2022. Lumos: Identifying and Localizing Diverse Hidden IoT Devices in an Unfamiliar Environment. In *USENIX Security*.

[543] Vandit Sharma and Mainack Mondal. 2022. Understanding and Improving Usability of Data Dashboards for Simplified Privacy Control of Voice Assistant Data. In *USENIX Security*.

[544] Dongdong She, Yizheng Chen, Abhishek Shah, Baishakhi Ray, and Suman Jana. 2020. Neutaint: Efficient Dynamic Taint Analysis with Neural Networks. In *IEEE S&P*.

[545] Dongdong She, Kexin Pei, Dave Epstein, Junfeng Yang, Baishakhi Ray, and Suman Jana. 2019. NEUZZ: Efficient Fuzzing with Neural Program Smoothing. In *IEEE S&P*.

[546] Ryan Sheastsley, Blaine Hoak, Eric Pauley, Yohan Beugin, Michael J. Weishman, and Patrick McDaniel. 2021. On the Robustness of Domain Constraints. In *ACM CCS*.

[547] Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *NDSS*.

[548] Virat Shejwalkar, Amir Houmansadr, Peter Kairouz, and Daniel Ramage. 2022. Back to the drawing board: A critical evaluation of poisoning attacks on production federated learning. In *IEEE S&P*.

[549] Cheng Shen, Tian Liu, Jun Huang, and Rui Tan. 2021. When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient. In *IEEE S&P*.

[550] Lugia Shen, Shouling Ji, Xuhong Zhang, Jinfeng Li, Jing Chen, Jie Shi, Chengfang Fang, Jianwie Yin, and Ting Wang. 2021. Backdoor Pre-trained Models Can Transfer to All. In *ACM CCS*.

[551] Yun Shen, Yufei Han, Zhikun Zhang, Min Chen, Ting Yu, Michael Backes, Yang Zhang, and Gianluca Stringhini. 2022. Finding MNEMON: Reviving Memories of Node Embeddings. In *ACM CCS*.

[552] Yun Shen, Xinlei He, Yufei Han, and Yang Zhang. 2022. Model Stealing Attacks Against Inductive Graph Neural Networks. In *IEEE S&P*.

[553] Yun Shenq, Enrico Mariconti, Pierre-Antoine Vervierq, and Gianluca Stringhini. 2018. Tiresias: Predicting Security Events Through Deep Learning. In *ACM CCS*.

[554] Rakshith Shetty, Bernt Schiele, and Mario Fritz. 2018. A4NT: Author Attribute Anonymity by Adversarial Training of Neural Machine Translation. In *USENIX Security*.

[555] Faysal Hossain Shezan, Kaiming Cheng, Zhen Zhang, Yinzhi Cao, and Yuan Tian. 2020. TKPERM: Cross-platform Permission Knowledge Transfer to Detect Overprivileged Third-party Applications. In *NDSS*.

[556] Chenghui Shi, Shouling Ji, Qianjun Liu, Changchang Liu, Yuefeng Chen, Yuan He, Zhe Liu, Raheem Beyah, and Ting Wang. 2020. Text Captcha Is Dead? A Large Scale Deployment and Empirical Study. In *ACM CCS*.

[557] Eui Chul Richard Shin, Dawn Song, and Reza Moazzezi. 2015. Recognizing Functions in Binaries with Neural Networks. In *USENIX Security*.

[558] Shen Shiqi, Shweta Shinde, Soundarya Ramesh, Abhik Roychoudhury, and Prateek Saxena. 2019. Neuro-Symbolic Execution: Augmenting Symbolic Execution with Neural Constraints. In *NDSS*.

[559] Maliheh Shirvanian and Nitesh Saxena. 2014. Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones. In *ACM CCS*.

[560] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-Preserving Deep Learning. In *ACM CCS*.

[561] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks against Machine Learning Models. In *IEEE S&P*.

[562] Xiaokui Shu, Danfeng (Daphne) Yao, and Naren Ramakrishnan. 2015. Unearthing Stealthy Program Attacks Buried in Extremely Long Execution Paths. In *ACM CCS*.

[563] Anatoly Shusterman, Ayush Agarwal, Sioli O'Connell, Daniel Genkin, Yossi Oren, and Yuval Yarom. 2021. Prime+Probe 1, JavaScript 0: Overcoming Browser-based Side-Channel Defenses. In *USENIX Security*.

[564] Anatoly Shusterman, Lachlan Kang, Yarden Haskal, Yosef Meltser, Prateek Mittal, Yossi Oren, and Yuval Yarom. 2019. Robust Website Fingerprinting Through the Cache Occupancy Channel. In *USENIX Security*.

[565] Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. 2022. WebGraph: Capturing Advertising and Tracking Information Flows for Robust Blocking. In *USENIX Security*.

[566] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2020. Encrypted DNS ⇒ Privacy? A Traffic Analysis Perspective. In *NDSS*.

[567] RavinduDe Silva, Mohamed Nabeel, Charith Elvitigala, Issa Khalil, Ting Yu, and Chamath Keppitiyagama. 2021. Compromised or Attacker-Owned: A Large Scale Classification and Study of Hosting Domains of Malicious URLs. In *USENIX Security*.

[568] Aditya Singh Rathore, Yijie Shen, Chenhan Xu, Jacob Snyderman, Jinsong Han, Fan Zhang, Zhengxiong Li, Feng Lin, Wenyao Xu, and Kui Ren. 2022. FakeGuard: Exploring Haptic Response to Mitigate the Vulnerability in Commercial Fingerprint Anti-Spoofing. In *NDSS*.

[569] Adnan Siraj Rakin, Yukui Luo, and Xiaolin Xu. 2021. Deep-Dup: An Adversarial Weight Duplication Attack Framework to Crush Deep Neural Network in Multi-Tenant FPGA. In *USENIX Security*.

[570] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM CCS*.

[571] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. 2019. Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning. In *ACM CCS*.

[572] Suphannee Sivakorn, Kangkook Jee, Yixin Sun, Lauri Korts-Parn, Zhichun Li, Cristian Lumezanu, Zhenyu Wu, Lu-An Tang, and Ding Li. 2019. Countering Malicious Processes with Process-DNS Association. In *NDSS*.

[573] Jared M. Smith and Max Schuchard. 2018. Routing Around Congestion Defeating DDoS Attacks and Adverse Network Conditions via Reactive BGP Routing. In *IEEE S&P*.

[574] Charles Smutz and Angelos Stavrou. 2016. When a Tree Falls: Using Diversity in Ensemble Classifiers to Identify Evasion in Malware Detectors. In *NDSS*.

[575] Sunbeom So, Seongjoon Hong, and Hakjoo Oh. 2021. SmarTest: Effectively Hunting Vulnerable Transaction Sequences in Smart Contracts through Language Model-Guided Symbolic Execution. In *USENIX Security*.

[576] Congzheng Song and Ananth Raghunathan. 2020. Information Leakage in Embedding Models. In *ACM CCS*.

[577] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine Learning Models that Remember Too Much. In *ACM CCS*.

[578] Jonghyuk Song, Sangho Lee, and Jong Kim. 2015. CrowdTarget: Target-based Detection of Crowdturfing in Online Social Networks. In *ACM CCS*.

[579] Liwei Song and Prateek Mittal. 2021. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security*.

[580] Liwei Song, Reza Shokri, and Prateek Mittal. 2019. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. In *ACM CCS*.

[581] Lushan Song, Jiaxuan Wang, Zhexuan Wang, Xinyu Tu, Guopeng Lin, Wenqiang Ruan, Haoqi Wu, and Weili Han. 2022. pMPL: A Robust Multi-Party Learning Framework with a Privileged Party. In *ACM CCS*.

[582] Wei Song, Heng Yin, Chang Liu, and Dawn Song. 2018. DeepMem: Learning Graph Neural Network Models for Fast and Robust Memory Forensic Analysis. In *ACM CCS*.

[583] Kyle Soska and Nicolas Christin. 2014. Automatically Detecting Vulnerable Websites Before They Turn Malicious. In *USENIX Security*.

[584] Kyle Soska and Nicolas Christin. 2015. Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem. In *USENIX Security*.

[585] R. Spencer Hallyburton, Yupei Liu, Yulong Cao, Z. Morley Mao, and Miroslav Pajic. 2022. Security Analysis of Camera-LiDAR Fusion Against Black-Box Attacks on Autonomous Vehicles. In *USENIX Security*.

[586] Nedim Srndic and Pavel Laskov. 2013. Detection of Malicious PDF Files Based on Hierarchical Document Structure. In *NDSS*.

[587] Nedim Srndic and Pavel Laskov. 2014. Practical Evasion of a Learning-Based Classifier: A Case Study. In *IEEE S&P*.

[588] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic Data – Anonymisation Groundhog Day. In *USENIX Security*.

[589] Timothy Stevens, Christian Skalka, Christelle Vincent, John Ring, Samuel Clark, and Joseph Near. 2022. Efficient Differentially Private Secure Aggregation for Federated Learning via Hardness of Learning with Errors. In *USENIX Security*.

[590] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. 2013. Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages. In *ACM CCS*.

[591] Aaron Stupple, David Singerman, and Leo Anthony Celi. 2019. The reproducibility crisis in the age of digital medicine. *NPJ digital medicine* 2, 1 (2019), 2.

[592] Liya Su, Xinyue Shen, Xiangyu Du, Xiaojing Liao, XiaoFeng Wang, Luyi Xing, and Baoxu Liu. 2021. Evil Under the Sun: Understanding and Discovering Attacks on Ethereum Decentralized Applications. In *USENIX Security*.

[593] Octavian Suciu, Radu Marginean, Yigitcan Kaya, Hal Daume III, and Tudor Dumitras. 2018. When Does Machine Learning FAIL? Generalized Transferability for Evasion and Poisoning Attacks. In *USENIX Security*.

[594] Octavian Suciu, Connor Nelson, Zhuoer Lyu, Tiffany Bao, and Tudor Dumitraș. 2022. Expected Exploitability: Predicting the Development of Functional Vulnerability Exploits. In *USENIX Security*.

[595] Shridatt Sugrim, Can Liu, Meghan McLean, and Janne Lindqvist. 2019. Robust Performance Metrics for Authentication Systems. In *NDSS*.

[596] Jiachen Sun, Yulong Cao, and Z. Morley Mao. 2020. Towards Robust LiDAR-based Perception in Autonomous Driving: General Black-box Adversarial Sensor Attack and Countermeasures. In *USENIX Security*.

[597] Jingchao Sun, Xiaocong Jin, Yimin Chen, Jinxue Zhang, Yanchao Zhang, and Rui Zhang. 2016. VISIBLE: Video-Assisted Keystroke Inference from Tablet Backside Motion. In *NDSS*.

[598] Suibin Sun, Le Yu, Xiaokuan Zhang, Minhui Xue, Ren Zhou, Haojin Zhu, Shuang Hao, and Xiaodong Lin. 2021. Understanding and Detecting Mobile Ad Fraud Through the Lens of Invalid Traffic. In *ACM CCS*.

[599] Zhibo Sun, Adam Oest, Penghui Zhang, Carlos Rubio-Medrano, Tiffany Bao, Ruoyu Wang, Ziming Zhao, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2021. Having Your Cake and Eating It: An Analysis of Concession-Abuse-as-a-Service. In *USENIX Security*.

[600] Zhichuang Sun, Ruimin Sun, Long Lu, and Alan Mislove. 2021. Mind Your Weight(s): A Large-scale Study on Insufficient Machine Learning Model Protection in Mobile Apps. In *USENIX Security*.

[601] Kimia Tajik, Akshith Gunasekharan, Rhea Dutta, Brandon Ellis, Rakesh B. Bobba, Mike Rosulek, Charles V. Wright, and Wu-Chi Feng. 2019. Balancing Image Privacy and Usability with Thumbnail-Preserving Encryption. In *NDSS*.

[602] Joshua Tan, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2020. Practical Recommendations for Stronger, More Usable Passwords Combining Minimum-strength, Minimum-length, and Blocklist Requirements. In *ACM CCS*.

[603] Mingtian Tan, Junpeng Wan, Zhe Zhou, and Zhou Li. 2021. Invisible Probe: Timing Attacks with PCIe Congestion Side-channel. In *IEEE S&P*.

[604] Sijun Tan, Brian Knott, Yuan Tian, and David J. Wu. 2021. CRYPTGPU: Fast Privacy-Preserving Machine Learning on the GPU. In *IEEE S&P*.

[605] Xin Tan, Yuan Zhang, Chenyuan Mi, Jiajun Cao, Kun Sun, Yifan Lin, and Min Yang. 2021. Locating the Security Patches for Disclosed OSS Vulnerabilities with Vulnerability-Commit Correlation Ranking. In *ACM CCS*.

[606] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. 2021. Demon in the Variant: Statistical Analysis of DNNs for Robust Backdoor Contamination Detection. In *USENIX Security*.

[607] Di Tang, Zhe Zhou, Yinqian Zhang, and Kehuan Zhang. 2018. Face Flashing: a Secure Liveness Detection Protocol based on Light Reflections. In *NDSS*.

[608] Siyuan Tang, Xianghang Mi, Ying Li, XiaoFeng Wang, and Kai Chen. 2022. Clues in Tweets: Twitter-Guided Discovery and Analysis of SMS Spam. In *ACM CCS*.

[609] Guanhong Tao, Yingqi Liu, Guangyu Shen, Qiuling Xu, Shengwei An, Zhuo Zhang, and Xiangyu Zhang. 2022. Model Orthogonalization: Class Distance Hardening in Neural Networks for Better Security. In *IEEE S&P*.

[610] Sai Teja Peddinti, Aleksandra Korolova, Elie Bursztein, and Geetanjali Sampemane. 2014. Cloak and Swagger: Understanding Data Sensitivity Through the Lens of User Anonymity. In *IEEE S&P*.

[611] Ege Tekiner, Abbas Acar, and A.Selcuk Uluagac. 2022. A Lightweight IoT Cryptojacking Detection Mechanism in Heterogeneous Smart Home Networks. In *NDSS*.

[612] Henrique Teles Maia, Chang Xiao, Dingzeyu Li, Eitan Grinspun, and Changxi Zheng. 2022. Can one hear the shape of a neural network?: Snooping the GPU via Magnetic Side Channel. In *USENIX Security*.

[613] Robert Templeman, Mohammed Korayem, DavidJ. Crandall, and Apu Kapadia. 2014. PlaceAvoider: Steering First-Person Cameras away from Sensitive Spaces. In *NDSS*.

[614] Saravanan Thirumuruganathan, Mohamed Nabeel, Euijin Choo, Issa Khalil, and Ting Yu. 2022. SIRAJ: A Unified Framework for Aggregation of Malicious Entity Detectors. In *IEEE S&P*.

[615] Kurt Thomas, Frank Li, Chris Grier, and Vern Paxson. 2014. Consequences of Connectivity: Characterizing Account Hijacking on Twitter. In *ACM CCS*.

[616] Kurt Thomas, Damon McCoy, Chris Grier, Alek Kolcz, and Vern Paxson. 2013. Trafficking Fraudulent Accounts: The Role of the Underground Market in Twitter Spam and Abuse. In *USENIX Security*.

[617] Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. 2022. On the Necessity of Auditable Algorithmic Definitions for Machine Unlearning. In *USENIX Security*.

[618] Han Tian, Chaoliang Zeng, Zhenghang Ren, Di Chai, Junxue Zhang, Kai Chen, and Qiang Yang. 2022. Sphinx: Enabling Privacy-Preserving Online Learning over the Cloud. In *IEEE S&P*.

[619] Jing Tian, Chengzhang Qu, Wenyuan Xu, and Song Wang. 2013. KinWrite: Handwriting-Based Authentication Using Kinect. In *NDSS*.

[620] Yuan Tian, Nan Zhang, Yueh-Hsun Lin, Xiaofeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. 2017. SmartAuth: User-Centered Authorization for the Internet of Things. In *USENIX Security*.

[621] Saeid Tizpaz-Niari, Pavol Černý, and Ashutosh Trivedi. 2020. Data-Driven Debugging for Functional Side Channels. In *NDSS*.

[622] Steve T.K. Jan, Qingying Hao, Tianrui Hu, Jiameng Pu, Sonal Oswal, Gang Wang, and Bimal Viswanath. 2020. Throwing Darts in the Dark? Detecting Bots with Limited Data using Neural Data Augmentation. In *IEEE S&P*.

[623] Florian Tramer, Reza Shokri, Ayrton San Joaquin, Hoang Le, Matthew Jagielski, Sanghyun Hong, and Nicholas Carlini. 2022. Truth Serum: Poisoning Machine Learning Models to Reveal Their Secrets. In *ACM CCS*.

[624] Florian Tramèr, Pascal Dupré, Gili Rusak, Giancarlo Pellegrino, and Dan Boneh. 2019. AdVersarial: Perceptual Ad Blocking meets Adversarial Machine Learning. In *ACM CCS*.

[625] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *USENIX Security*.

[626] Rahmadi Trimananda, Hieu Le, Hao Cui, Janice Tran Ho, Anastasia Shuba, and Athina Markopoulou. 2022. OVRseen: Auditing Network Traffic and Privacy Policies in Oculus VR. In *USENIX Security*.

[627] Carmela Troncoso and Emiliano De Cristofaro. 2018. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. In *NDSS*.

[628] Sebastian Uellenbeck, Markus Dürmuth, Christopher Wolf, and Thorsten Holz. 2013. Quantifying the Security of Graphical Passwords: The Case of Android Unlock Patterns. In *ACM CCS*.

[629] Wajih Ul Hassan, Mark Lemay, Nuraini Aguse, Adam Bates, and Thomas Moyer. 2018. Towards Scalable Cluster Auditing through Grammatical Inference over Provenance Graphs. In *NDSS*.

[630] Erkam Uzun, Simon Pak Ho Chung, Irfan Essa, and Wenke Lee. 2018. rtCaptcha: A Real-Time CAPTCHA Based Liveness Detection System. In *NDSS*.

[631] Erkam Uzun, Simon P. Chung, Vladimir Kolesnikov, Alexandra Boldyreva, and Wenke Lee. 2021. Fuzzy Labeled Private Set Intersection with Applications to Private Real-Time Biometric Search. In *USENIX Security*.

[632] Pratik Vaishnavi, Kevin Eykholt, and Amir Rahmati. 2022. Transferring Adversarial Robustness Through Robust Representation Matching. In *USENIX Security*.

[633] Erik van der Kouwe, Gernot Heiser, Dennis Andriesse, Herbert Bos, and Cristiano Giuffrida. 2019. SoK: Benchmarking flaws in systems security. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 310–325.

[634] Patrick Vandewalle, Jelena Kovacevic, and Martin Vetterli. 2009. Reproducible research in signal processing. *IEEE Signal Processing Magazine* 26, 3 (2009), 37–47.

[635] Antoine Vastel, Pierre Laperdrix, Walter Rudametkin, and Romain Rouvoy. 2018. FP-STALKER: Tracking Browser Fingerprint Evolutions. In *IEEE S&P*.

[636] Shivaram Venkataraman, Erik Bodzsar, Indrajit Roy, Alvin AuYoung, and Robert S. Schreiber. 2013. Presto: Distributed Machine Learning and Graph Processing with Sparse Matrices. In *ACM CCS*.

[637] Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On Semantic Patterns of Passwords and their Security Impact. In *NDSS*.

[638] Alexander Viand, Patrick Jattke, and Anwar Hithnawi. 2021. SoK: Fully Homomorphic Encryption Compilers. In *IEEE S&P*.

[639] Nishant Vishwamitra, Hongxin Hu, Feng Luo, and Long Cheng. 2021. Towards Understanding and Detecting Cyberbullying in Real-world Images. In *NDSS*.

[640] Thomas Vissers, Wouter Joosen, and Nick Nikiforakis. 2015. Parking Sensors: Analyzing and Detecting Parked Domains. In *NDSS*.

[641] Bimal Viswanath, M. Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards Detecting Anomalous User Behavior in Online Social Networks. In *USENIX Security*.

[642] Binghui Wang and Neil Zhenqiang Gong. 2019. Attacking Graph-based Classification via Manipulating the Graph Structure. In *ACM CCS*.

[643] Binghui Wang, Jinyuan Jia, and Neil Zhenqiang Gong. 2019. Graph-based Security and Privacy Analytics via Collective Classification with Joint Weight Learning and Propagation. In *NDSS*.

[644] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. 2021. DataLens: Scalable Privacy Preserving Training via Gradient Compression and Aggregation. In *ACM CCS*.

[645] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and BenY. Zhao. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. In *IEEE S&P*.

[646] Bolun Wang, Yuanshun Yao, Bimal Viswanath, and Haitao Zheng. 2018. With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning. In *USENIX Security*.

[647] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing Hyperparameters in Machine Learning. In *IEEE S&P*.

[648] Ding Wang, Haibo Cheng, Ping Wang, Jeff Yan, and Xinyi Huang. 2018. A Security Analysis of Honeywords. In *NDSS*.

[649] Daimeng Wang, Ajaya Neupane, Zhiyun Qian, Nael Abu-Ghazaleh, Srikanth V. Krishnamurthy Edward J. M. Colbert, and Paul Yu. 2019. Unveiling your keystrokes: A Cache-based Side-channel Attack on Graphics Libraries. In *NDSS*.

[650] Daimeng Wang, Zheng Zhang, Hang Zhang, Zhiyun Qian, Srikanth V. Krishnamurthy, and Nael Abu-Ghazaleh. 2021. SyzVegas: Beating Kernel Fuzzing Odds with Reinforcement Learning. In *USENIX Security*.

[651] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. 2013. You Are How You Click: Clickstream Analysis for Sybil Detection. In *USENIX Security*.

[652] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. 2014. Man vs. Machine: Practical Adversarial Detection of Malicious Crowdsourcing Workers. In *USENIX Security*.

[653] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, and Jingbo Sun. 2018. De-anonymization of Mobility Trajectories: Dissecting the Gaps between Theory and Practice. In *NDSS*.

[654] Jinghan Wang, Chengyu Song, and Heng Yin. 2021. Reinforcement Learning-based Hierarchical Seed Scheduling for Greybox Fuzzing. In *NDSS*.

[655] Liang Wang, Kevin P. Dyer, Aditya Akella, Thomas Ristenpart, and Thomas Shrimpton. 2015. Seeing through Network-Protocol Obfuscation. In *ACM CCS*.

[656] Lun Wang, Usmann Khan, Joseph Near, Qi Pang, Jithendaraa Subramanian, Neel Somani, Peng Gao, Andrew Low, and Dawn Song. 2022. PrivGuard: Privacy Regulation Compliance Made Easier. In *USENIX Security*.

[657] Peng Wang, Xiaojing Liao, Yue Qin, and XiaoFeng Wang. 2020. Into the Deep Web: Understanding E-commerce Fraud from Autonomous Chat with Cybercriminals. In *NDSS*.

[658] Peng Wang, Xianghang Mi, Xiaojing Liao, XiaoFeng Wang, Kan Yuan, Feng Qian, and Raheem Beyah. 2018. Game of Missuggestions: Semantic Analysis of Search-Autocomplete Manipulations. In *NDSS*.

[659] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A. Gunter. 2019. Charting the Atack Surface of Trigger-Action IoT Platforms. In *ACM CCS*.

[660] Qi Wang, Wajih Ul Hassan, Ding Li, Kangkook Jee, Xiao Yu, Kexuan Zou, Junghwan Rhee, Zhengzhang Chen, Wei Cheng, Carl A. Gunter, and Haifeng Chen. 2020. You Are What You Do: Hunting Stealthy Malware via Data Provenance Analysis. In *NDSS*.

[661] Ruowen Wang, William Enck, Douglas Reeves, Xinwen Zhang, Peng Ning, Dingbang Xu, Wu Zhou, and Ahmed M. Azab. 2015. EASEAndroid: Automatic Policy Analysis and Refinement for Security Enhanced Android via Large-Scale Semi-Supervised Learning. In *USENIX Security*.

[662] Shu Wang, Jiahao Cao, Xu He, Kun Sun, and Qi Li. 2020. When the Differences in Frequency Domain are Compensated: Understanding and Defeating Modulated Replay Attacks on Automatic Speech Recognition. In *ACM CCS*.

[663] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. 2018. Formal Security Analysis of Neural Networks using Symbolic Intervals. In *USENIX Security*.

[664] Tao Wang. 2020. High Precision Open-World Website Fingerprinting. In *IEEE S&P*.

[665] Tao Wang. 2021. The One-Page Setting: A Higher Standard for Evaluating Website Fingerprinting Defenses. In *ACM CCS*.

[666] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. 2014. Effective Attacks and Provable Defenses for Website Fingerprinting. In *USENIX Security*.

[667] Wei Wang, Yao Yao, Xin Liu, Xiang Li, Pei Hao, and Ting Zhu. 2021. I Can See the Light: Attacks on Autonomous Vehicles Using Invisible Lights. In *ACM CCS*.

[668] Xiuling Wang and WendyHui Wang. 2022. Group Property Inference Attacks Against Graph Neural Networks. In *ACM CCS*.

[669] Yuanda Wang, Hanqing Guo, and Qiben Yan. 2022. GhostTalk: Interactive Attack on Smartphone Voice System Through Power Line. In *NDSS*.

[670] YanXiongChengSuWenchaoHuangFuyouMiaoWansen Wang and Hengyi Ouyang. 2020. SmartVerif: Push the Limit of Automation Capability of Verifying

Security Protocols by Dynamic Strategies. In *USENIX Security*.

[671] Yinglei Wang, Wing-kei Yu, Sara Q. Xu, Edwin Kan, and G. Edward Suh. 2013. Hiding Information in Flash Memory. In *IEEE S&P*.

[672] Jean-Luc Watson, Sameer Wagh, and Raluca Ada Popa. 2022. Piranha: A GPU Platform for Secure Computation. In *USENIX Security*.

[673] Rolfvan Wegberg, Samaneh Tajalizadehkhoob, Kyle Soska, Ugur Akyazi, CarlosHernandez Ganan, Bram Klievink, Nicolas Christin, and Michelvan Eeten. 2018. Plug and Prey? Measuring the Commoditization of Cybercrime via Online Anonymous Markets. In *USENIX Security*.

[674] Jianxin Wei, Ergute Bao, Xiaokui Xiao, and Yin Yang. 2022. DPIS: An Enhanced Mechanism for Differentially Private SGD with Importance Sampling. In *ACM CCS*.

[675] Chenkai Weng, Kang Yang, Xiang Xie, Jonathan Katz, and Xiao Wang. 2021. Mystique: Efficient Conversions for Zero-Knowledge Proofs with Applications to Machine Learning. In *USENIX Security*.

[676] Emily Wenger, Max Bronkers, Christian Cianfarani, Jenna Cryan, Angela Sha, Haitao Zheng, and Ben Y. Zhao. 2021. "Hello, It's Me": Deep Learning-based Speech Synthesis Attacks in the Real World. In *ACM CCS*.

[677] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, DavidA. Wagner, and Konstantin Beznosov. 2017. The Feasibility of Dynamically Granted Permissions: Aligning Mobile Privacy with User Preferences. In *IEEE S&P*.

[678] Stefan Winter, Christopher S Timperley, Ben Hermann, Jürgen Cito, Jonathan Bell, Michael Hilton, and Dirk Beyer. 2022. A retrospective study of one decade of artifact evaluations. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 145–156.

[679] Cong Wu, Jing Chen, Kun He, Ziming Zhao, Ruiying Du, and Chen Zhang. 2022. EchoHand: High Accuracy and Presentation Attack Resistant Hand Authentication on Commodity Mobile Devices. In *ACM CCS*.

[680] Cong Wu, Kun He, Jing Chen, Ziming Zhao, and Ruiying Du. 2020. Liveness is Not Enough: Enhancing Fingerprint Authentication with Behavioral Biometrics to Defeat Puppet Attacks. In *USENIX Security*.

[681] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. 2022. LINKTELLER: Recovering Private Edges from Graph Neural Networks via Influence Analysis. In *IEEE S&P*.

[682] Howard Wu, Wenting Zheng, Alessandro Chiesa, RalucaAda Popa, and Ion Stoica. 2018. DIZK: A Distributed Zero Knowledge Proof System. In *USENIX Security*.

[683] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. 2020. The Value of Collaboration in Convex Machine Learning with Differential Privacy. In *IEEE S&P*.

[684] Ruoyu Wu, Taegyu Kim, Dave (Jing) Tian, Antonio Bianchi, and Dongyan Xu. 2022. DnD: A Cross-Architecture Deep Neural Network Decompiler. In *USENIX Security*.

[685] Shujiang Wu, Jianjia Yu, Min Yang, and Yinzhi Cao. 2022. Rendering Contention Channel Made Practical in Web Browsers. In *USENIX Security*.

[686] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. 2021. Adversarial Policy Training against Deep Reinforcement Learning. In *USENIX Security*.

[687] Shengqu Xi, Shao Yang, Xusheng Xiao, Yuan Yao, Yayuan Xiong, Fengyuan Xu, Haoyu Wang, Peng Gao, Zhuotao Liu, Feng Xu, and Jian Lu. 2019. DeepIntent: Deep Icon-Behavior Learning for Detecting Intention-Behavior Discrepancy in Mobile Apps. In *ACM CCS*.

[688] Zhaohan Xi, Ren Pang, Shouling Ji, and Ting Wang. 2021. Graph Backdoor. In *USENIX Security*.

[689] Chong Xiang, ArjunNitin Bhagoji, Vikash Sehwag, and Prateek Mittal. 2021. PatchGuard: A Provably Robust Defense against Adversarial Patches via Small Receptive Fields and Masking. In *USENIX Security*.

[690] Chong Xiang, Saeed Mahloujifar, and Prateek Mittal. 2022. PatchCleanser: Certifiably Robust Defense against Adversarial Patches for Any Image Classifier. In *USENIX Security*.

[691] Chong Xiang and Prateek Mittal. 2021. DetectorGuard: Provably Securing Object Detectors against Localized Patch Hiding Attacks. In *ACM CCS*.

[692] Chengcheng Xiang, Yudong Wu, Bingyu Shen, Mingyao Shen, Haochen Huang, Tianyin Xu, Yuanyuan Zhou, Cindy Moore, Xinxin Jin, and Tianwei Sheng. 2019. Towards Continuous Access Control Validation and Forensics. In *ACM CCS*.

[693] Chaowei Xiao, Armin Sarabi, Yang Liu, Bo Li, Mingyan Liu, and Tudor Dumitras. 2018. From Patching Delays to Infection Symptoms: Using Risk Profiles for an Early Discovery of Vulnerabilities Exploited in the Wild. In *USENIX Security*.

[694] Qiuyu Xiao, MichaelK. Reiter, and Yinqian Zhang. 2015. Mitigating Storage Side Channels Using Statistical Privacy Mechanisms. In *ACM CCS*.

[695] Jiayun Xu, Yingjiu Li, and Robert H. Deng. 2021. Differential Training: A Generic Framework to Reduce Label Noises for Android Malware Detection. In *NDSS*.

[696] Ming Xu, Chuanwang Wang, Jitao Yu, Junjie Zhang, Kai Zhang, and Weili Han. 2021. Chunk-Level Password Guessing: Towards Modeling Refined Password Composition Representations. In *ACM CCS*.

[697] Teng Xu, Gerard Goossen, HuseyinKerem Cevahir, Sara Khodeir, Yingyezhe Jin, Frank Li, Shawn Shan, Sagar Patel, David Freeman, and Paul Pearce. 2021. Deep Entity Classification: Abusive Account Detection for Online Social Networks. In *USENIX Security*.

[698] Weilin Xu, David Evans, and Yanjun Qi. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *NDSS*.

[699] Weitao Xu, Guohao Lan, Qi Lin, Sara Khalifa, Neil Bergmann, Mahbub Hassan, and Wen Hu. 2017. KEH-Gait: Towards a Mobile Healthcare User Authentication System by Kinetic Energy Harvesting. In *NDSS*.

[700] Weilin Xu, Yanjun Qi, and David Evans. 2016. Automatically Evading Classifiers: A Case Study on PDF Malware Classifiers. In *NDSS*.

[701] Xiaojun Xu, Chang Liu, Qian Feng, Heng Yin, Le Song, and Dawn Song. 2017. Neural Network-based Graph Embedding for Cross-Platform Binary Code Similarity Detection. In *ACM CCS*.

[702] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, CarlA. Gunter, and Bo Li. 2021. Detecting AI Trojans Using Meta Neural Analysis. In *IEEE S&P*.

[703] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos. In *USENIX Security*.

[704] Fabian Yamaguchi, Alwin Maier, Hugo Gascon, and Konrad Rieck. 2015. Automatic Inference of Search Patterns for Taint-Style Vulnerabilities. In *IEEE S&P*.

[705] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. 2019. The Catcher in the Field: A Fieldprint based Spoofing Detection for Text-Independent Speaker Verification. In *ACM CCS*.

[706] Mengjia Yan, Christopher W. Fletcher, and Josep Torrellas. 2020. Cache Telepathy: Leveraging Shared Resource Attacks to Learn DNN Architectures. In *USENIX Security*.

[707] Limin Yang, Wenbo Guo, Qingying Hao, Arridhana Ciptadi, Ali Ahmadzadeh, Xinyu Xing, and Gang Wang. 2021. CADE: Detecting and Explaining Concept Drift Samples for Security Applications. In *USENIX Security*.

[708] Ronghai Yang, Xianbo Wang, Cheng Chi, Dawei Wang, Jiawei He, Siming Pang, and WingCheong Lau. 2021. Scalable Detection of Promotional Website Defacements in Black Hat SEO Campaigns. In *USENIX Security*.

[709] Yuqing Yang, Mohamed Elsabagh, Chaoshun Zuo, Ryan Johnson, Angelos Stavrou, and Zhiqiang Lin. 2022. Detecting and Measuring Misconfigured Manifests in Android Apps. In *ACM CCS*.

[710] Yijun Yang, Ruiyuan Gao, Yu Li, Qiuxia Lai, and Qiang Xu. 2022. What You See is Not What the Network Infers: Detecting Adversarial Examples Based on Semantic Contradiction. In *NDSS*.

[711] Zhiju Yang, Weiping Pei, Monchu Chen, and Chuan Yue. 2022. WTAGRAPH: Web Tracking and Advertising Detection using Graph Neural Networks. In *IEEE S&P*.

[712] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural Network Inversion in Adversarial Setting via Background Knowledge Alignment. In *ACM CCS*.

[713] Fan Yao, Adnan Siraj Rakin, and Deliang Fan. 2020. DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips. In *USENIX Security*.

[714] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y. Zhao. 2019. Latent Backdoor Attacks on Deep Neural Networks. In *ACM CCS*.

[715] Yuanshun Yao, Bimal Viswanath, Jenna Cryan, Haitao Zheng, and Ben Y. Zhao. 2017. Automated Crowdturfing Attacks and Defenses in Online Review Systems. In *ACM CCS*.

[716] Guixin Ye, Zhanyong Tang, Dingyi Fang, Zhanxing Zhu, Yansong Feng, Pengfei Xu, Xiaojiang Chen, and Zheng Wang. 2018. Yet Another Text Captcha Solver: A Generative Adversarial Network Based Approach. In *ACM CCS*.

[717] Ting-Fang Yen, Victor Heorhiadi, Alina Oprea, MichaelK. Reiter, and Ari Juels. 2014. An Epidemiological Study of Malware Encounters in a Large Enterprise. In *ACM CCS*.

[718] Jeffrey Young, Song Liao, Long Cheng, Hongxin Hu, and Huixing Deng. 2022. SkillDetective: Automated Policy-Violation Detection of Voice Assistant Applications in the Wild. In *USENIX Security*.

[719] Hyunwoo Yu, Jaemin Lim, Kiyeon Kim, and Suk-Bok Lee. 2018. Pinto: Enabling Video Privacy for Commodity IoT Cameras. In *ACM CCS*.

[720] Honggang Yu, Kaichen Yang, Teng Zhang, Yun-Yun Tsai, Tsung-Yi Ho, and Yier Jin. 2020. CloudLeak: Large-Scale Deep Learning Models Stealing Through Adversarial Examples. In *NDSS*.

[721] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. 2019. Differentially Private Model Publishing for Deep Learning. In *IEEE S&P*.

[722] Lingjing Yu, Bo Luo, Zhaoyu Zhou, and Qingyun Liu. 2020. You Are What You Broadcast: Identification of Mobile and IoT Devices from (Public) WiFi. In *USENIX Security*.

[723] Sheng Yu, Yu Qu, Xunchao Hu, and Heng Yin. 2022. DeepDi: Learning a Relational Graph Convolutional Network Model on Instructions for Fast and Accurate Disassembly. In *USENIX Security*.

[724] Zhiyuan Yu, Zhuohang Li, Yuanhaur Chang, Skylar Fong, Jian Liu, and Ning Zhang. 2022. HeatDeCam: Detecting Hidden Spy Cameras via Thermal Emissions. In *ACM CCS*.

[725] Dong Yuan, Yuanli Miao, Neil Zhenqiang Gong, Zheng Yang, Qi Li, Dawn Song, Qian Wang, and Xiao Liang. 2019. Detecting Fake Accounts in Online Social

Networks at the Time of Registrations. In *ACM CCS*.
[726] Kan Yuan, Haoran Lu, Xiaojing Liao, and XiaoFeng Wang. 2018. Reading Thieves' Cant: Automatically Identifying and Understanding Dark Jargons from Cybercrime Marketplaces. In *USENIX Security*.
[727] Kan Yuan, Di Tang, Xiaojing Liao, XiaoFeng Wang, Xuan Feng, Yi Chen, Menghan Sun, Haoran Lu, and Kehuan Zhang. 2019. Stealthy Porn: Understanding Real-World Adversarial Images for Illicit Online Promotion. In *IEEE S&P*.
[728] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Huang Heqing, XiaoFeng Wang, and Carl A. Gunter. 2018. CommanderSong: A Systematic Approach for Practical Adversarial Voice Recognition. In *USENIX Security*.
[729] Yuanyuan Yuan, Qi Pang, and Shuai Wang. 2022. Automated Side Channel Analysis of Media Software with Manifold Learning. In *USENIX Security*.
[730] Mojtaba Zaheri, Yossi Oren, and Reza Curtmola. 2022. Targeted Deanonymization via the Cache Side Channel: Attacks and Defenses. In *USENIX Security*.
[731] Santiago Zanella-Béguelin, Lukas Wutschitz, Shruti Tople, Victor Rühle, Andrew Paverd, Olga Ohrimenko, Boris Köpf, and Marc Brockschmidt. 2020. Analyzing Information Leakage of Updates to Natural Language Models. In *ACM CCS*.
[732] Jun Zeng, Xiang Wang, Jiahao Liu, Yinfang Chen, Zhenkai Liang, Tat-Seng Chua, and ZhengLeong Chua. 2022. SHADEWATCHER: Recommendation-guided Cyber Threat Analysis using System Audit Records. In *IEEE S&P*.
[733] Mingming Zha, Jice Wang, Yuhong Nan, Xiaofeng Wang, Yuqing Zhang, and Zelin Yang. 2022. Hazard Integrated: Understanding Security Risks in App Extensions to Team Chat Systems. In *NDSS*.
[734] Guoming Zhang, Xiaoyu Ji, Xinfeng Li, Gang Qu, and Wenyuan Xu. 2021. EarArray: Defending against DolphinAttack via Acoustic Attenuation. In *NDSS*.
[735] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. DolphinAttack: Inaudible Voice Commands. In *ACM CCS*.
[736] Jiaheng Zhang, Zhiyong Fang, Yupeng Zhang, and Dawn Song. 2020. Zero Knowledge Proofs for Decision Tree Predictions and Accuracy. In *ACM CCS*.
[737] Jiaheng Zhang, Tianyi Liu, Weijie Wang, Yinuo Zhang, Dawn Song, Xiang Xie, and Yupeng Zhang. 2021. Doubly Efficient Interactive Proofs for General Arithmetic Circuits with Linear Prover Time. In *ACM CCS*.
[738] Jiang Zhang, Konstantinos Psounis, Muhammad Haroon, and Zubair Shafiq. 2022. HARPO: Learning to Subvert Online Behavioral Advertising. In *NDSS*.
[739] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. VoiceLive: A Phoneme Localization based Liveness Detection for Voice Authentication on Smartphones. In *ACM CCS*.
[740] Lei Zhang, Zhemin Yang, Yuyu He, Zhenyu Zhang, Zhiyun Qian, Geng Hong, Yuan Zhang, and Min Yang. 2018. Invetter: Locating Insecure Input Validations in Android Services. In *ACM CCS*.
[741] Minxing Zhang, Zhaochun Ren, Zihan Wang, Pengjie Ren, Zhumin Chen, Pengfei Hu, and Yang Zhang. 2021. Membership Inference Attacks Against Recommender Systems. In *ACM CCS*.
[742] Nan Zhang, Xianghang Mi, Xuan Feng, XiaoFeng Wang, Yuan Tian, and Feng Qian. 2019. Dangerous Skills: Understanding and Mitigating Security Risks of Voice-Controlled Third-Party Functions on Virtual Personal Assistant Systems. In *IEEE S&P*.
[743] Qiao Zhang, Chunsheng Xin, and Hongyi Wu. 2021. GALA: Greedy ComputAtion for Linear Algebra in Privacy-Preserved Neural Networks. In *NDSS*.
[744] Wen Zhang, You Chen, Thaddeus R. Cybulski, Daniel Fabbri, Carl A. Gunter, Patrick Lawlor, David Liebovitz, and Bradley Malin. 2014. Decide Now or Decide Later? Quantifying the Tradeoff between Prospective and Retrospective Access Decisions. In *ACM CCS*.
[745] Wei Zhang, Yan Meng, Yugeng Liu, Xiaokuan Zhang, Yinqian Zhang, and Haojin Zhu. 2018. HoMonit: Monitoring Smart Home Apps from Encrypted Traffic. In *ACM CCS*.
[746] Wanrong Zhang, Shruti Tople, and Olga Ohrimenko. 2021. Leakage of Dataset Properties in Multi-Party Machine Learning. In *USENIX Security*.
[747] Xing Zhang, Jiongyi Chen, Chao Feng, Ruilin Li, Yunfei Su, Bin Zhang, Jing Lei, and Chaojing Tang. 2021. Reducing Test Cases with Attention Mechanism of Neural Networks. In *USENIX Security*.
[748] Xiaokuan Zhang, Jihun Hamm, Michael K. Reiter, and Yinqian Zhang. 2019. Statistical Privacy for Streaming Traffic. In *NDSS*.
[749] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. 2020. Interpretable Deep Learning under Fire. In *USENIX Security*.
[750] Xiaokuan Zhang, Xueqiang Wang, Xialong Bai, Yianqian Zhang, and XiaoFeng Wang. 2018. OS-level Side Channels without Procfs: Exploring Cross-App Information Leakage on iOS. In *NDSS*.
[751] Xiaohan Zhang, Yuan Zhang, Ming Zhong, Daizong Ding, Yinzhi Cao, Yukun Zhang, Mi Zhang, and Min Yang. 2020. Enhancing State-of-the-art Classifiers with API Semantics to Detect Evolved Android Malware. In *ACM CCS*.
[752] Yang Zhang, Mathias Humbert, Bartlomiej Surma, Praveen Manoharan, Jilles Vreeken, and Michael Backes. 2020. Towards Plausible Graph Anonymization. In *NDSS*.
[753] Yiming Zhang, Baojun Liu, Chaoyi Lu, Zhou Li, Haixin Duan, Shuang Hao, Mingxuan Liu, Ying Liu, Dong Wang, and Qiang Li. 2020. Lies in the Air: Characterizing Fake-base-station Spam Ecosystem in China. In *ACM CCS*.
[754] Yinqian Zhang and Michael K. Reiter. 2013. Düppel: Retrofitting Commodity Operating Systems to Mitigate Cache Side Channels in the Cloud. In *ACM CCS*.
[755] Yangyong Zhang, Lei Xu, Abner Mendoza, Guangliang Yang, Phakpoom Chinprutthiwong, and Guofei Gu. 2019. Life after Speech Recognition: Fuzzing Semantic Misinterpretation for Voice Assistant Applications. In *NDSS*.
[756] Yige Zhang, Xuy Yuan, Jin Li, Jiadong Lou, Li Chen, and Nian-Feng Tzeng. 2021. Reverse Attack: Black-box Attacks on Collaborative Recommendation. In *ACM CCS*.
[757] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. 2022. Inference Attacks Against Graph Neural Networks. In *USENIX Security*.
[758] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. 2018. CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. In *ACM CCS*.
[759] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. 2021. PrivSyn: Differentially Private Data Synthesis. In *USENIX Security*.
[760] Zhenkai Zhang, Zihao Zhan, Daniel Balasubramanian, Bo Li, Péter Völgyesi, and XenofonD. Koutsoukos. 2020. Leveraging EM Side-Channel Information to Detect Rowhammer Attacks. In *IEEE S&P*.
[761] Benjamin ZiHao Zhao, Hassan Jameel Asghar, and Mohamed Ali Kaafar. 2020. On the Resilience of Biometric Authentication Systems against Random Inputs. In *NDSS*.
[762] Kaifa Zhao, Hao Zhou, Yulin Zhu, Xian Zhan, Kai Zhou, Jianfeng Li, Le Yu, Wei Yuan, and Xiapu Luo. 2021. Structural Attack against Graph Based Android Malware Detection. In *ACM CCS*.
[763] Yue Zhao, Hong Zhu, Kai Chen, and Shengzhi Zhang. 2021. AI-Lancet: Locating Error-inducing Neurons to Optimize Neural Networks. In *ACM CCS*.
[764] Yue Zhao, Hong Zhu, Ruigang Liang, Qintao Shen, Shengzhi Zhang, and Kai Chen. 2019. Seeing isn't Believing: Towards More Robust Adversarial Attack Against Real World Object Detectors. In *ACM CCS*.
[765] Ziming Zhao, Gail-Joon Ahn, Jeong-Jin Seo, and Hongxin Hu. 2013. On the Security of Picture Gesture Authentication. In *USENIX Security*.
[766] Baolin Zheng, Peipei Jiang, Qian Wang, Qi Li, Chao Shen, Cong Wang, Yunjie Ge, Qingyang Teng, and Shenyi Zhang. 2021. Black-box Adversarial Attacks on Commercial Speech Platforms with Minimal Information. In *ACM CCS*.
[767] Haizhong Zheng, Minhui Xue, Hao Lu, Shuang Hao, Haojin Zhu, Xiaohui Liang, and Keith Ross. 2018. Smoke Screener or Straight Shooter: Detecting Elite Sybil Attacks in User-Review Social Networks. In *NDSS*.
[768] Tengfei Zheng, Tongqing Zhou, Qiang Liu, Kui Wu, and Zhiping Cai. 2022. Characterizing and Detecting Non-Consensual Photo Sharing on Social Networks. In *ACM CCS*.
[769] Wenting Zheng, Ryan Deng, Weikeng Chen, Raluca Ada Popa, Aurojit Panda, and Ion Stoica. 2021. Cerebro: A Platform for Multi-Party Cryptographic Collaborative Learning. In *USENIX Security*.
[770] Wenting Zheng, RalucaAda Popa, Joseph E. Gonzalez, and Ion Stoica. 2019. Helen: Maliciously Secure Coopetitive Learning for Linear Models. In *IEEE S&P*.
[771] Neil Zhenqiang Gong and Bin Liu. 2016. You Are Who You Know and How You Behave: Attribute Inference Attacks via Users' Social Friends and Behaviors. In *USENIX Security*.
[772] Hao Zhou, Xiapu Luo, Haoyu Wang, and Haipeng Cai. 2022. Enhanced Membership Inference Attacks against Machine Learning Models. In *ACM CCS*.
[773] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. 2022. Property Inference Attacks Against GANs. In *NDSS*.
[774] Jianping Zhu, Rui Hou, XiaoFeng Wang, Wenhao Wang, Jiangfeng Cao, Boyan Zhao, Zhongpu Wang, Yuhui Zhang, Jiameng Ying, Lixin Zhang, and Dan Meng. 2020. Enabling Rack-scale Confidential Computing using Heterogeneous Trusted Execution Environment. In *IEEE S&P*.
[775] Tong Zhu, Yan Meng, Haotian Hu, Xiaokuan Zhang, Minhui Xue, and Haojin Zhu. 2021. Dissecting Click Fraud Autonomy in the Wild. In *ACM CCS*.
[776] Tao Zhu, David Phipps, Adam Pridgen, Jedidiah R. Crall, and Dan S.Wallach. 2013. The Velocity of Censorship: High-Fidelity Detection of Microblog Post Deletions. In *USENIX Security*.
[777] Wanzheng Zhu, Hongyu Gong, Rohan Bansal, Zachary Weinberg, Nicolas Christin, Giulia Fanti, and Suma Bhat. 2021. Self-Supervised Euphemism Detection and Identification for Content Moderation. In *IEEE S&P*.
[778] Yuankun Zhu, Yueqiang Cheng, Husheng Zhou, and Yantao Lu. 2021. Hermes Attack: Steal DNN Models with Lossless Inference Accuracy. In *USENIX Security*.
[779] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. 2021. Can We Use Arbitrary Objects to Attack LiDAR Perception in Autonomous Driving?. In *ACM CCS*.
[780] Sebastian Zimmeck and Steven M. Bellovin. 2014. Privee: An Architecture for Automatically Analyzing Web Privacy Policies. In *USENIX Security*.
[781] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, and Joel Reidenberg. 2017. Automated Analysis of Privacy Requirements for Mobile Apps. In *NDSS*.
[782] Adam Morrison Zirui Neil Zhao, Christopher W. Fletcher, and Josep Torrellas. 2022. Binoculars: Contention-Based Side-Channel Attacks Exploiting the Page Walker. In *USENIX Security*.

[783] Peiyuan Zong, Tao Lv, Dawei Wang, Zizhuang Deng, Ruigang Liang, and Kai Chen. 2020. FuzzGuard: Filtering out Unreachable Inputs in Directed Grey-box Fuzzing through Deep Learning. In *USENIX Security*.

[784] Fei Zuo, Xiaopeng Li, Patrick Young, Lannan Luo, Qiang Zeng, and Zhexin Zhang. 2019. Neural Machine Translation Inspired Binary Code Similarity Comparison beyond Function Pairs. In *NDSS*.

A APPENDIX

A.1 List of Papers We Analyze by Year

2013 - [490], [63], [152], [165], [192], [269], [270], [295], [319], [342], [321], [413], [446], [452], [471], [497], [534], [586], [590], [616], [619], [628], [636], [651], [671], [754], [765], [776]

2014 - [14], [459], [27], [40], [47], [59], [74], [146], [183], [200], [254], [291], [398], [417], [449], [610], [491], [501], [559], [583], [587], [613], [615], [637], [641], [652], [666], [717], [744], [780]

2015 - [56], [64], [65], [66], [73], [78], [91], [162], [182], [208], [249], [267], [308], [317], [318], [348], [377], [382],[396], [418], [432], [444], [447], [483], [515], [557], [560], [562], [578], [584], [640], [655],[661], [694], [704]

2016 - [5], [45], [84], [87], [138], [141], [144], [168], [173], [181], [402], [193], [771], [255], [362], [367], [378], [409], [410], [414], [457], [469], [474], [540], [574], [597], [625], [700], [703], [739]

2017 - [21], [3], [39], [55], [325], [62],[86], [94], [108], [123], [136], [140], [156], [175], [178], [210], [222], [240], [287], [330], [356], [368], [372], [405], [412], [425], [453], [458], [464], [531], [561], [316], [577], [620], [677], [701], [699], [715], [735], [781]

2018 - [12], [13], [42], [46], [71], [204], [72], [90], [460], [92], [103], [109], [137], [174], [238], [191], [194], [195], [197], [207], [214], [218], [224], [629], [229], [236], [265], [272], [273], [281], [283], [289], [292], [300], [303], [276], [336], [343], [354], [359], [381], [421], [424], [431], [433], [437], [439], [445], [463], [470], [480], [504], [507], [510], [530], [541], [553], [554], [570], [573], [582], [593], [607], [627], [630], [635], [647], [646], [648], [653], [658], [663], [673], [682], [693], [698], [716], [719], [726], [728], [740], [745], [750], [758], [767]

2019 - [8], [17], [38], [57], [77], [93], [99], [117], [119], [120], [124], [135], [139], [150], [153], [155], [161], [169], [171], [172], [180], [199], [228], [244], [261], [280], [288], [309], [327], [338], [339], [345], [350], [366], [369], [380], [391], [399], [411], [416], [427], [440], [448], [450], [462], [482], [505], [513], [514], [523], [528], [529], [545], [558], [564], [571], [572], [580], [595], [601], [624], [642], [643], [645], [649], [659], [687], [692], [705], [712], [714], [721], [725], [727], [742], [748], [755], [764], [770], [784]

2020 - [7], [15], [167], [19], [28], [30], [43], [51], [52], [142], [89], [95], [96], [98], [100], [105], [114], [116], [133], [159], [164], [170], [185], [206], [226], [235], [258], [264], [622], [304], [315], [329], [332], [337], [347], [352], [357], [364], [393], [394], [403], [422], [423], [430], [442], [455], [465], [467], [472], [479], [484], [487], [488], [489], [492], [493], [499], [500], [502], [511], [522], [538], [539], [544], [555], [556], [566], [397], [576], [596], [602], [621], [657], [660], [662], [664], [670], [680], [683], [706], [713], [720], [722], [731], [736], [749], [751], [752], [753], [760], [761], [774], [783]

2021 - [4], [6], [9], [10], [18], [22], [24], [29], [32], [33], [34], [44], [53], [54], [293], [68], [69], [75], [76], [81], [83], [85], [97], [101], [106], [112], [113], [115], [118], [125], [322], [131], [132], [134], [147], [148], [149], [154], [157], [166], [179], [186], [187], [196], [201], [216], [217], [220], [223], [227], [230], [232], [233], [234], [241], [242], [243], [246], [251], [252], [253], [257], [266], [271], [275], [277], [278], [285], [296], [298], [299], [307], [311], [312], [320], [324], [328], [331], [333], [341], [344], [346], [349], [353], [355], [358], [360], [365], [387], [370], [376], [388], [389], [390], [401], [419], [428], [434], [438], [441], [302], [461], [475], [476], [477], [478], [481], [569], [503], [508], [512], [516], [517], [518], [519], [520], [524], [525], [527], [532], [535], [546], [547], [549], [550], [563], [567], [575], [579], [592], [598], [599], [600], [603], [604], [605], [606], [631], [638], [509], [639], [644], [650], [654], [665], [667], [675], [676], [686], [688], [689], [691], [695], [696], [697], [702], [707], [708], [734], [737], [741], [743], [746], [747], [756], [759], [486], [762], [763], [766], [769], [775], [777], [778], [779]

2022 - [11], [20], [2], [23], [26], [31], [36], [37], [41], [58], [61], [67], [80], [82], [88], [102], [104], [107], [110], [111], [121], [128], [129], [130], [145], [151], [158], [163], [176], [188], [189], [190], [198], [202], [203], [213], [585], [221], [225], [231], [237], [245], [247], [248], [250], [259], [260], [263], [268], [274], [279], [282], [284],[286], [290], [294], [297], [301], [262], [305], [306], [310], [323], [334], [335], [340], [351], [361], [371], [373], [374], [375], [379], [383],[384], [385], [386], [392], [395], [400], [612], [404], [408], [415], [420], [426], [429], [435], [436], [451], [160], [456], [730], [466], [468], [473], [143], [498], [568], [506], [521], [533], [537], [536], [542], [543], [548], [551], [552], [565], [581], [588], [589], [407], [594], [608], [609], [611], [614], [617], [618], [623], [626], [782], [632], [494], [656], [668], [669], [672], [674], [679], [681], [684], [685], [690], [709], [710], [711], [718], [723], [724], [729], [732], [733], [738], [757], [768], [772], [773]