# Toward a Standard Benchmark for Computer Security Research

The Worldwide Intelligence Network Environment (WINE)

Tudor Dumitraș
Symantec Research Labs
tudor_dumitras@symantec.com

Darren Shou
Symantec Research Labs
darren_shou@symantec.com

## Abstract

Unlike benchmarks that focus on performance or reliability evaluations, a benchmark for computer security must necessarily include sensitive code and data. Because these artifacts could damage systems or reveal personally identifiable information about the users affected by cyber attacks, publicly disseminating such a benchmark raises several scientific, ethical and legal challenges. We propose the Worldwide Intelligence Network Environment (WINE), a security-benchmarking approach based on rigorous experimental methods. WINE includes representative field data, collected worldwide from 240,000 sensors, for new empirical studies, and it will enable the validation of research on all the phases in the lifecycle of security threats. We tackle the key challenges for security benchmarking by designing a platform for repeatable experimentation on the WINE data sets and by collecting the metadata required for understanding the results. In this paper, we review the unique characteristics of the WINE data, we discuss why rigorous benchmarking will provide fresh insights on the security arms race and we propose a research agenda for this area.

## 1. Introduction

The security-related data sets that are available today are insufficient for answering many challenging questions or for rigorous experimental research. For example, little is known about the origins and prevalence of zero-day attacks, because the existing data on malware dissemination does not reach back in time *before* the discovery of the malware. We currently do not understand how scam sites conceal their presence and move to avoid detection, for lack of historical information on malicious URLs. So far, we have not been able to follow a security vulnerability over the course of its entire life—from a programming bug that evades testing, through its stealth exploitation in zero-day attacks, its discovery and description in a public advisory, the release of a patch for the vulnerability and of anti-virus signatures, the automatic generation of exploits based on the patch, and to the final race between these attacks and the remediation measures introduced by the security community. Answering such questions requires the analysis and the correlation of multiple data sets, collected independently from diversified sensors. The lack of such data sets prevents us from gaining the deep insights needed for tipping the balance of the security arms race from the attackers to the defenders.

Moreover, data sets used for validating computer security research are often mentioned in a single publication and then forgotten. For example, real malware samples are readily available on the Internet, and they are often used for validating research results. However, this experimental method does not accommodate a *sound validation* of the research, because other investigators do not have access to the same collection of samples and cannot reproduce the results. This prevents rigorous comparisons between alternative approaches proposed in the scientific literature. Additionally, the malware samples alone do not tell the whole story. Ancillary *field data* is needed to understand the malware lifecycle and the economic incentives of cybercrime.

We aim to fill these gaps by (i) making representative field data, which covers the entire lifecycle of malware, available to the research community, and (ii) developing a platform for repeatable experimentation around these data sets. We build on the lessons learned in other research fields where benchmarking is well established (*e.g.* networking and databases), while identifying some of the key differences for security benchmarking.

We center our benchmarking approach around the data sets available in WINE[1], Symantec's program for sharing data with the research community. For example, WINE includes information on *unknown binaries* found on the Internet. The users who opt in for the reputation-based security features of Symantec products accept to share the list of binary files downloaded on their machines in exchange for a whitelist of binaries with good reputation. The data includes historical in-

*BADGERS'11* 10 April 2011, Salzburg, Austria
Copyright © 2011 ACM 978-1-4503-0615-7/11/04…$10.00

[1] More information on accessing the WINE data is available at `http://www.symantec.com/WINE`.

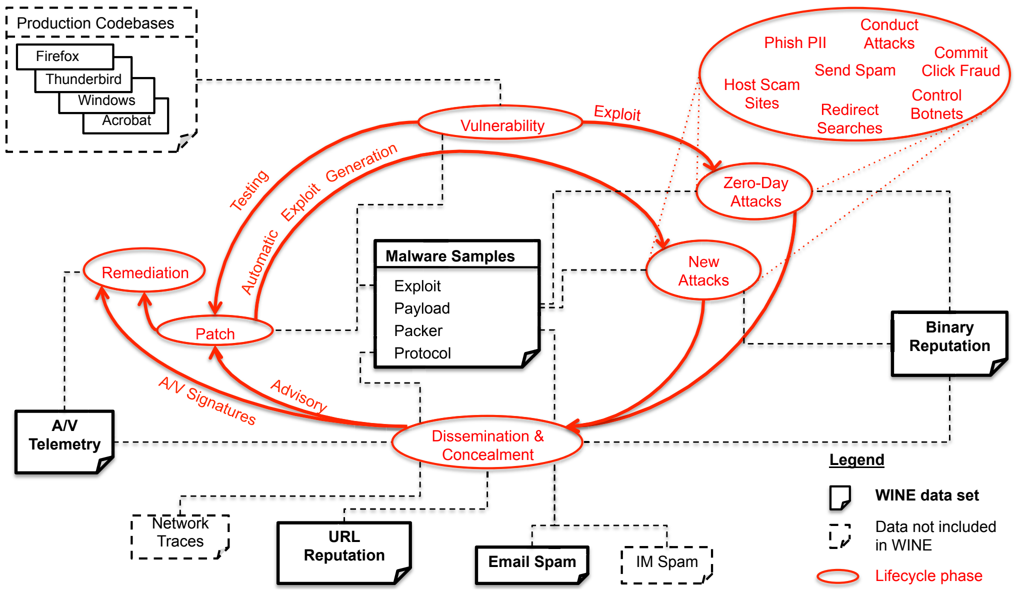

**Figure 1.** The WINE data sets enable the study of the entire lifecycle of security threats. By correlating research findings with additional data sets, available from other sources, experimenters can assemble an end-to-end image of the security arms race.

formation on 1 billion files that the security community has not yet classified as either benign or malware. The historical records start when each file appeared on the Internet (estimated through the discovery timestamps assigned by the 50 million active instances of the reputation feature) and can provide unique insights on the mechanisms of zero-day attacks. Similarly, Symantec tracks the spread of known host-based and network-based cyber threats, filters spam out of approximately one third of the world's email and has assembled perhaps the largest collection of malware samples. By combining five distinct data sets, sampled from this collection of field data, WINE provides an overview of the security-threat landscape (see Figure 1).

We are currently developing a data storage and analysis platform, which aims to ensure *experimental repeatability* by archiving snapshots of the data used in each experiment and by providing researchers with tools for recording all the information required for reproducing the results. This will enable comparisons of the effectiveness, performance and scalability of published techniques. Moreover, WINE will include *metadata* allowing researchers to establish whether a data set is representative for the real-world cyber threats. To protect the sensitive information included in the data and to ensure the reproducibility of experimental results, all the experiments and empirical studies will be conducted on the WINE platform hosted by Symantec Research Labs.

Our ultimate goal is to develop a rigorous benchmark for computer security research. Because defensive mechanisms can make different trade-offs, which might be appropriate for different systems and settings, we will avoid reporting a single number indicating which mechanism is the best. Like the TPC and SPEC benchmarks, which focus on performance evaluation, our security benchmark will not be definitive. The WINE data sets must be updated periodically in order to reflect the frequent changes in the security threat landscape.

While the WINE data sets are currently available to the research community, the data *per se* is not sufficient for defining a rigorous benchmark. In this position paper, our goal is not to present benchmark results or to discuss the lessons learned from this effort. Instead, we make two contributions:

- We propose a research agenda for security benchmarking, by identifying the main challenges (Section 2) and several open questions that could be answered once these challenges are overcome (Section 4);
- We propose an approach for benchmarking computer security (Section 3), which combines the WINE data sets with a platform for rigorous experimentation. We explain WINE's data sharing model, and we outline solutions to some of the key challenges for security benchmarking.

Our data sharing program does not focus exclusively on computer security—enabling, for example, research on software reliability or on machine learning techniques for billion-node graphs. Moreover, the results of experimental research will guide the inclusion of additional data sets in WINE. We believe that, in the future, the WINE data will provide key insights for the fields of security, dependability, machine learning and software engineering.

## 2. Challenges for benchmarking security

Unlike in the systems community, where data sets have sometimes outlived the system for which they were collected,[2] the

[2] For example, in case of the Sprite filesystem trace [Baker et al. 1991].

data sets used for validating computer-security research are often forgotten after the initial publication referencing them. This experimental method does not accommodate an independent verification of results and meaningful comparisons against the prior art. The lack of standard benchmarks for computer security is the result of scientific, ethical, and legal challenges for publicly disseminating security-related data sets. In this paper we focus on the scientific challenges, but we also review other challenges that are likely to have an impact on the benchmarking techniques.

### 2.1 Scientific challenges

ℂ1 *A benchmark for computer security must be based on field data.* Some benchmarking efforts in the past have addressed privacy concerns by generating synthetic data, based on the observed statistical distributions of the raw data samples collected [Lippmann et al. 2000]. Moreover, synthetically generated data provides considerable flexibility, allowing an experimenter to explore all the behavioral corner cases of the system-under-test [DeWitt 1993]. For security-oriented benchmarks, however, it is difficult to relate the benchmarking results to the real-world performance of the system-under-test. For example, the false positive rate of intrusion detection systems is influenced by the background noise, which should be consistent with the background data that the system is likely to encounter in a real deployment [McHugh 2000].

ℂ2 *The benchmarking approach must ensure experimental repeatability.* The data sets used in the experiments must be archived for future reference, and they must be considered again in research projects attempting quantitative comparisons against the prior results. Moreover, in order to make it possible for future projects to reproduce the experimental results, the benchmark must provide tools for recording the *experiment metadata*—*e.g.*, the hypotheses tested, the experimental design, the scripts and procedures used for data analysis, the statistical apparatus employed.

ℂ3 *The benchmark must be representative of the real-world threat landscape.* Any large data collection can ensure the statistical significance of the experimental results. However, the validity of these results can still be questioned in cases where small mutations of the test data can drastically change the outcome of the experiment. The benchmark should provide the *collection metadata* needed for establishing the real-world situations that each data set is representative of. Moreover, the benchmark must remain relevant, in spite of the frequent changes in the cyber threat landscape and of data filtering at multiple levels (see also Challenges ℂ5 and ℂ6). We point out that updating the benchmark regularly does not conflict with ℂ2. The benchmark must specify a predictable process for data collection [Camp et al. 2009], while preserving the reference data sets employed in prior experiments. Similarly, as security metrics are not well understood, the benchmark must suggest metrics in order to enable direct comparisons among similar techniques, but must allow researchers to define improved metrics that are more relevant for the hypotheses tested.

ℂ4 *Experiments must be conducted at a realistic scale.* Security is difficult to measure and assess objectively because it represents an end-to-end property of the system. Some metrics (*e.g.* resistance to intrusions) can not be measured directly and must be approximated through large-scale observations of the whole system, in order to achieve precise estimations.

ℂ5 *Benchmarking must take the information quality into account.* In many large scale collections, uncertainty about the data is explicit. For example, as heuristics and machine-learning techniques are used, increasingly, for detecting polymorphic malware, the labels applied to the binaries analyzed are no longer a black-and-white determination, but, rather, they express a certain level of confidence that the binary is malicious. In a commercial product, where monitoring and logging represent secondary concerns, the submissions are throttled back, and sometimes truncated, in over to avoid overloading the users' machines and to reduce the bandwidth costs incurred. Moreover, the hash functions used for identifying binaries may change, as the products evolve, and the techniques used for identifying user machines are not always reliable. We must develop new query languages and analysis tools that treat such information-quality metrics as first-class entities.

### 2.2 Ethical challenges

ℂ6 *Do no harm.* A benchmark for computer security must include sensitive code and data, which could damage computer systems or could reveal personally identifiable information about the users affected by the cyber attacks recorded. For example, the IP addresses of hosts initiating network-based attacks could point to personal computers that have been infected with malware, while the country codes of the attack destinations reveal further sensitive information. Binary samples of malware must not be made freely available on the Internet. It is challenging to determine, *a priori*, how to sample or filter the raw data collected in order to meet these challenges.

### 2.3 Legal challenges

ℂ7 *Compliance with privacy laws often restricts the data collection, storage and exchange.* The field data needed for security benchmarking (see Challenge ℂ1) is collected from real networks and users. There are several laws that limit access to network traffic or that regulate the storage of this information. In the United States, for example, the Wiretap Act prohibits the interception of content of electronic communications, the Pen/Trap statute prohibits the real-time interception of non-content, and the Stored Communications Act prohibits providers from knowingly disclosing their customer's communications. In contrast to HIPAA, which restricts disclosures of health information but provides means for researchers to obtain relevant information, the privacy laws contain no exceptions for research. The PREDICT project [DHS 2011b], sponsored by the Department of Homeland Security, could provide a framework for addressing this challenge.

## 3. A benchmark for computer security

We build upon the lessons learned from the failures and successes of the previous efforts for benchmarking computer security [for example: Camp et al. 2009, Leita et al. 2010, Lippmann et al. 2000, Maxion and Townsend 2004, McHugh 2000] and for building platforms allowing rigorous measurements and experimentation [for example: DeWitt 1993, Eide et al. 2007, Paxson 2004]. In addition to archiving snapshots of the data sets used in each experiment, we will store the scripts used for aggregating and analyzing the data, and we will maintain a *lab book* that records all the steps taken by the experimenter. This experimental metadata is essential for ensuring the reproducibility of the results (challenge $\mathbb{C}2$). Keeping a lab book is a common practice in other experimental fields, such as applied physics or cell biology.

The selection of the initial data sets for WINE was guided by our goal to establish a benchmark for computer security and by the needs expressed in the security community [Camp et al. 2009]. However, the access to the WINE data is not restricted to security researchers. WINE aims to aggregate the data feeds collected by Symantec in order to enable experimental research across a broad spectrum of disciplines, *e.g.*, dependability, machine learning, software engineering, networking, economics, visual analytics.

### 3.1 Operational model

To protect the sensitive information included in the data sets, WINE will only be accessed on-site at Symantec Research Labs. While researchers will have access to the raw data collected, *we will not create a malware library for anyone to download at will, and we will ensure that private information is not disseminated in public* (challenge $\mathbb{C}6$). Moreover, some aspects of the data collection process, such as the internal operation of the various Symantec sensors, will not be disclosed in detail. A snapshot of the data used in each experiment will be archived, for future reference, and all the analysis and experimentation will be conducted on the WINE infrastructure (described in Section 3.3). The researchers will retain all right, title and interest to the research results.

More information on accessing WINE is available at `http://www.symantec.com/WINE`.

### 3.2 The WINE data sets

WINE will provide access to a large collection of malware samples, and to the contextual information needed to understand how malware spreads and conceals its presence, how it gains access to different systems, what actions it performs once it is in control and how it is ultimately defeated. WINE includes representative field data, collected at Symantec (challenge $\mathbb{C}1$). WINE will include five data sets, summarized in Table 1: binary-reputation data, email-spam data, URL-reputation data, A/V telemetry and malware samples. These data sets enable two research directions: (i) empirical studies for *understanding each phase in the lifecycle of cyberattacks*, and (ii) quantitative evaluations and comparisons of attack prevention or detection techniques, for *benchmarking security systems*.

**Understanding the lifecycle of cyberattacks.** WINE aims to cover the entire lifecycle of malware attacks (see Figure 1). For example, the binary-reputation data set enables—for the first time, to the best of our knowledge—a study of the origins and prevalence of *zero-day attacks*, which exploit vulnerabilities that are unknown or unacknowledged publicly. Searching the history of binary-reputation submissions for files that are known to be malicious indicates for how long the file has existed in the wild before it was first detected (*i.e.*, before the security community created the corresponding anti-virus signatures). The subsequent proliferation of the attack and the *effectiveness of the remediation mechanisms* introduced (*e.g.*, patches for the vulnerability exploited, A/V signatures for detecting and blocking the attack) can be further traced in the A/V telemetry data set.

Similarly, by correlating the URLs recorded in the email spam samples, in the binary reputation and in the URL reputation data sets, we can begin to understand *how scam sites conceal themselves* to avoid detection (*e.g.*, by moving to a different IP address) and the effectiveness of the various *mechanisms for disseminating malware* (*e.g.*, spam, intrusions, drive-by downloads). The malware samples in WINE illustrate the attackers' aims—the actions that malware tries to perform once it takes control of a host—, and by corroborating these observations with data from the real-world victims of these attacks we can gain insight into the economic incentives of cybercrime. The data sets included in WINE are collected independently, from diversified sensors, allowing researchers to examine a phenomenon from multiple perspectives and to improve the confidence in the conclusions we draw from these investigations (challenge $\mathbb{C}3$).

Moreover, by combining WINE with data from additional sources, such as code repositories for open source software that have known vulnerabilities, we can study a security threat from the time when a programming bug introduces a vulnerability until the time when the last exploit of that vulnerability disappears from the A/V telemetry.

**Benchmarking computer security.** Because most of the techniques developed in the security community can serve both sides of the arms race, defensive mechanisms usually aim to force attackers to do more work than defenders have to do. WINE allows testing this, or similar, hypotheses for existing security systems, by defining *macro-benchmarks* that are representative for real-world workloads of systems aiming to fight viruses, worms or botnets. For example, the telemetry data can serve as the ground truth for heuristic threat-detection

| Data set | Sources | Description |
|---|---|---|
| Binary reputation | 50 million machines | Information on unknown binaries—*i.e.*, files for which an A/V signature has not yet been created—that are downloaded by users who opt in for Symantec's reputation-based security program. This data can indicate for how long a particular threat has existed in the wild before it was first detected. Each record includes the submission timestamp, as well as the cryptographic hash and the download URL of the binary. |
| A/V telemetry | 130 million machines | Records occurrences of known threats, for which Symantec has created signatures and which can be detected by anti-virus products. This data set includes intrusion-detection telemetry. Each record includes the detection timestamp, the signature of the attack, the OS version of the attack's target, the name of the compromised process and the file or URL which originated the attack. |
| Email spam | 2.5 million decoy accounts | Samples of phishing and spam emails, collected by Symantec's enterprise-grade systems for spam filtering. This data set includes samples of email spam and statistics on the messages blocked by the spam filters. |
| URL reputation | 10 million domains | Website-reputation data, collected by crawling the web and by analyzing malicious URLs (a simplified interface for querying this data is available at `http://safeweb.norton.com/`). Each record includes the crawl timestamp, the URL, as well as the name and the type of threat found at that URL. A subset of this data was used to analyze the rogue A/V campaigns [Cova et al. 2010]. |
| Malware samples | 200 countries | A collection of both packed and unpacked malware samples (viruses, worms, bots, etc.), used for creating Symantec's A/V signatures. A subset of these samples was used for validating research on automatic malware detection [Griffin et al. 2009]. |

**Table 1.** The WINE data sets.

algorithms that operate on the binary-reputation data set. The data is also amenable to the statistical techniques that have been proposed in the past for insider attack attribution, such as naïve Bayes classification, Markov modeling or temporal sequence matching [Maxion and Townsend 2004].

These macro-benchmarks provide a corpus of field data for present and future experimenters, allowing them to measure multiple characteristics of a security tool, such as its latency, its scalability, and its threat detection accuracy. Because popular benchmarks can have a lasting impact on the design of security systems, we will regularly update the WINE data to ensure that the benchmarks are representative of the threat landscape in the real world (challenge $\mathbb{C}3$).

## 3.3 Experimental approach

The WINE data sets described above represent only half of the security benchmark. To achieve experimental reproducibility, we are currently building a platform for storing and analyzing the data. This platform enables data-intensive applications by adopting a *shared-nothing* architecture, illustrated in Figure 2. The data is partitioned across multiple storage nodes, attached directly to the hosts that execute data analysis tasks. The management infrastructure of the cluster minimizes the amount of data that must be transferred through the local area network by placing, whenever possible, the analysis tasks directly on the nodes that already store the data required. This is achieved by maintaining multiple indexes for each data set and by making these indexes available on all the nodes of the system. For example, the binary-reputation data set is indexed on both the hash of the binary and the download URL, to facilitate the correlation of data with the A/V telemetry, as well as with the email spam and URL-reputation data. This design will allow researchers to run experiments at scale (challenge $\mathbb{C}4$).

The experimental platform allows querying the data sets using either ANSI SQL or MapReduce tasks [Dean and Ghemawat 2004], for greater flexibility. WINE receives updates regularly from Symantec's collection of 240,000 sensors, which are distributed worldwide. Based on the raw data available in WINE, researchers define *reference data sets* that are relevant for their experiments. After the experiments are completed, the reference data sets are archived in network-attached storage, for future comparisons against the results obtained (challenge $\mathbb{C}2$).[3]

This design is similar to other architectures for data-intensive computing, such as MapReduce or parallel databases [Pavlo et al. 2009]. Unlike the prior work, we aim to ensure the *experimental reproducibility*, within the context of Symantec's data collection process. This goal will be achieved, in part, by providing integrated tools to help researchers manage and record their activities, either planned or unplanned [Eide et al. 2007]. These tools will facilitate the development of scripts that repeat the experimental procedure, *e.g.* by recording the interactive terminal sessions, and they will provide a

[3] The malware data set is stored and analyzed in a *red lab*, which does not have inbound/outbound network connectivity in order to prevent viruses and worms from escaping this isolated environment (challenge $\mathbb{C}6$).

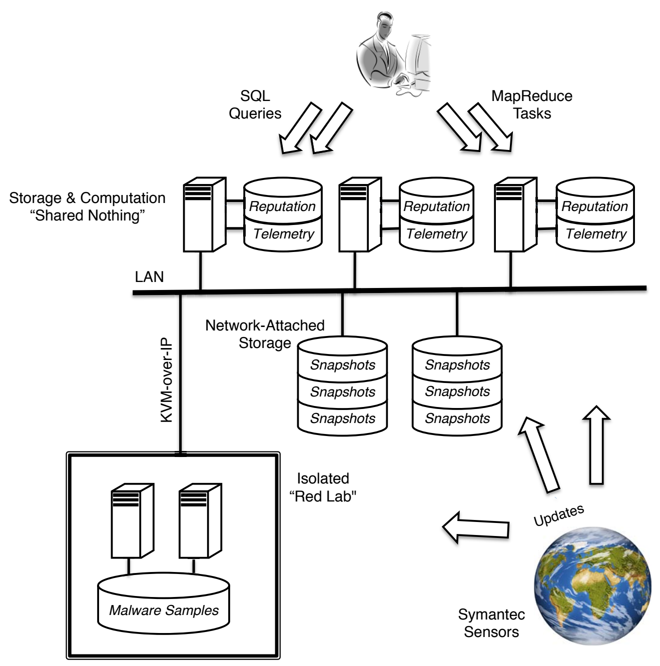

**Figure 2.** Architecture of the WINE platform. WINE is a data-intensive system, which focuses on ensuring the reproducibility and comparability of experimental results.

detailed record of the experiment. However, the lab book will also require a conscious effort from the researcher for documenting the experimental hypothesis and the purpose of each procedural step (challenge ℂ2). For example, when creating a taxonomy of the malware samples included in WINE, the lab book should detail the rationale for the selection of each classification feature.

Moreover, we will implement mechanisms for assessing the *information quality*, which is a measure of how fit the information is for benchmarking purposes [Keeton et al. 2009]. For example, as MapReduce is known to exhibit a significant response-time variability [Zaharia et al. 2008], we will estimate the measurement *precision* by repeating an experiment multiple times and recording the standard deviation of the results [Chatfield 1983]. When supplementing the data sets with information collected on the server-side—*e.g.*, by performing a reverse DNS query on a IP address that is observed to be the source of an attack, in order to determine its network location before the DNS record is deregistered—we will assess the data *staleness* by comparing the collection timestamps. Whenever possible, we will record the throttling rates of the submissions, and we will also maintain updated *aggregate statistics* on all the data sets. Such measures of information quality will allow us to incorporate statistical techniques for handling the measurement errors[4] into our automated tools for classifying, filtering and mining the data and will enable researchers to draw meaningful conclusions from the experiments (challenge ℂ5).

**Proposed metrics.** Several metrics are needed for evaluating the detection accuracy, scalability and responsiveness of systems benchmarked. The *receiver operating curve (ROC)* plots the true-positive detection rate of an algorithm against the rate of false-positive warnings. For data sets where the ground truth is available, a *confusion matrix* tabulates the attack instances, as classified by the algorithm under evaluation, against the true classes of those attacks, and it can provide deeper insights about the strengths and weaknesses of the algorithm. These metrics have been used in the past for comparing the performance of techniques for detecting masqueraders [Maxion and Townsend 2004].

The ability to create reference data sets of different sizes and to provision resources in the experimental platform enables a further investigation of the system scalability. The *scaleup* measures the system's ability to maintain a constant response time when solving increasingly larger problems only by adding a proportional amount of storage and computational resources—*i.e.*, if we double the resources, can we solve a problem twice as large? In contrast, the *speedup* indicates whether adding resources results in a corresponding decrease in the response time—*i.e.*, if we double the resources, can we solve the same problem twice as fast? Both these metrics were introduced for evaluating the scalability of parallel database systems [DeWitt 1993].

Finally, the characteristics of the *response-time distributions* are important for systems where the detection of threats is time sensitive. In these situations, reporting the mean response time is not sufficient, as many data-intensive systems are known to be scalable, but to exhibit heavy-tailed latency distributions [Zaharia et al. 2008]. The *high percentiles* of the latency distributions should also be reported and compared, such as the 95th and 99th percentiles that are commonly used in the industry to specify the guarantees provided in service-level agreements [Google Inc. 2011].

## 4. Discussion

The WINE data sets and the platform for repeatable experimentation provide the opportunity to ask a number of research questions. While a complete list of such questions is beyond the scope of this paper, we provide a few examples to guide the research agenda for exploring this space.

**How to avoid vulnerabilities in computer programs?** The introduction of security vulnerabilities during software evolution was studied by analyzing the revision logs and bug databases of large, production-quality codebases. For example, this approach pointed out how effective the software vendors are in dealing with zero-day attacks [Frei 2009], which vulnerabilities occur repeatedly as a result of software reuse [Pham et al. 2010] and the most common programming errors that lead to vulnerabilities [CWE/SANS 2010]. However,

[4] For example, the precision of estimation can be improved by combining results from multiple instruments, which are characterized by different measurement errors, and results that are likely to be imprecise can be discarded after performing a $3\sigma$ test. Such techniques are widely used in engineering disciplines [Chatfield 1983].

these findings do not discern the security vulnerabilities that are ultimately exploited and that help malware propagate in the wild, which emphasizes a fundamental shortcoming in our assessment of software quality. By correlating data from open-source software repositories with the information provided by WINE, we have the opportunity to gain a deeper understanding of security vulnerabilities. This will allow us to minimize the impact of vulnerabilities by focusing on the programming bugs that matter.

**What are the sources of zero-day attacks?** These attacks exploit vulnerabilities that are not acknowledged publicly, *e.g.*, while the software vendor is working on patching the vulnerability. We currently do not know if malware creators identify vulnerabilities predominantly through a form of fuzz testing [Miller et al. 1990] or from insider information. We could gain insight into the sources and prevalence of zero-day attacks by analyzing the binary-reputation data set and by correlating this information with events recorded in other system logs.

**Is malware installed predominantly through exploits or through voluntary downloads?** This question could be answered by analyzing the telemetry and the binary-reputation data sets and has important implications for understanding the dissemination mechanisms of malware and for validating the working assumptions of current intrusion-detection systems.

**Does the large-scale dissemination of security patches make the world a safer place?** Techniques for exploiting vulnerabilities automatically—by reverse engineering security patches—have been introduced recently [Brumley et al. 2008], but we lack empirical data about their impact in the real world. The telemetry data set can highlight, for instance, if fewer attacks are recorded immediately after the release of updates and, in general, can shed additional light on this aspect of the security arms race.

While these questions originate from a domain that we are familiar with, we believe that the WINE data is interesting from other perspectives as well (*e.g.*, for the economical sciences, storage systems, network performance analysis). By lowering the bar for validating advances in these fields, WINE will promote controversially innovative research, which introduces new ideas with the potential to change the community's perspective. For example, investigating the feasibility of patching unknown software vulnerabilities automatically, at run-time, currently requires laborious and expensive red-teaming experiments [Perkins et al. 2009]. However, these controversial questions are the ones most likely to lead to disruptive innovations in our field. WINE will allow such research projects to establish credibility through rigorous, quantitative validations using representative field data.

## 5. Related work

Camp et al. [2009] compile a "data wish list" for cybersecurity research and emphasize the need for representative field data in the research community. In addition to specific data that is currently unavailable—such as annotated network traces, URLs received in spam emails, representative malware samples—the authors identify the need for a data-sharing process that facilitates the collection of metadata and that addresses the privacy and legal concerns. In this paper, we propose such a process for the WINE data sets. WINE provides many of the items on the wish list, and it also includes unique data sets that were not foreseen by Camp et al. (*e.g.*, historical information on malicious executables extending before the threat identification).

Lippmann et al. [2000] describe the Lincoln Labs data set for benchmarking intrusion detection systems. The data set is synthesized from the statistical distributions observed in the network traffic from several Air Force bases. McHugh [2000] criticizes this work for the lack of information on the validation of test data—such as measures of similarity with the traffic traces or a rationale for concluding that similar behaviors should be expected when exposing the systems-under-test to real world data. McHugh observes that the experimenter has the burden of proof for showing that the artificial environment does not affect the outcome of the experiment. Maxion and Townsend [2004] emphasize the importance of careful experimental design for the ability to identify subtle flaws in the data. These lessons learned endure in the community: the PREDICT data repository [DHS 2011b] was also criticized for the lack of adequate metadata, and Camp et al. [2009] emphasize the need for metadata that allows experimenters to distinguish meaningful conclusions from artifacts. One of the major thrusts in our benchmarking effort is to ensure that all the metadata on experiments and on the data-collection process is included in WINE.

We draw inspiration from other research fields, where benchmarking is well established. For example, Paxson [2004] catalogs the metadata that must be recorded when measuring the performance of network protocols. Eide et al. [2007] report their observations from running Emulab. which underlies the DETER testbed for experimental cybersecurity [DHS 2011a], and emphasize the importance of automatically recording experimental processes for the ability to reproduce the results later. DeWitt [1993] presents the design of the Wisconsin Benchmark, which produced the seminal ideas in database benchmarking. In this paper, we identify the key differences between these approaches and security benchmarking, such as the need for representative field data and for frequently updating the reference data sets, and we propose mechanisms for addressing these challenges.

## 6. Summary

Through WINE, we aim to develop a benchmark that covers the entire lifecycle of security threats. WINE includes five data sets, providing access not only to malware samples, but also to the contextual information needed to understand how malware spreads and conceals its presence, how it gains access to different systems, what actions it performs once it is in control and how it is ultimately defeated. The unique features of these data sets allow us to address several research questions that are still outstanding, such as the prevalence and origins of zero-day attacks. Moreover, by correlating these data sets with information from additional sources, *e.g.* the revision logs and bug databases of open source software, we can follow the entire lifecycle of a security threat from the introduction of a vulnerability in a software component to the disappearance of the last exploit of that vulnerability. We will enable the reproducibility of results by archiving the reference data sets used in experiments, by including the metadata required for determining what each data set is representative of and by providing integrated tools for recording the hypotheses tested and the procedures employed in order to draw meaningful conclusions from experimental results. We believe that this new benchmarking approach will provide key insights for the fields of security, machine learning and software engineering.

## Acknowledgments

We thank Roy Maxion, Dave Brumley, Jon McCune, Ahren Studer, Jim Newsome and Marc Dacier for their early feedback on the ideas presented in this paper.

## References

BAKER, M. G., HARTMAN, J. H., KUPFER, M. D., SHIRRIFF, K. W., AND OUSTERHOUT, J. K. 1991. Measurements of a distributed file system. In *ACM Symposium on Operating Systems Principles*. Pacific Grove, CA, 198–212.

BRUMLEY, D., POOSANKAM, P., SONG, D. X., AND ZHENG, J. 2008. Automatic patch-based exploit generation is possible: Techniques and implications. In *IEEE Symposium on Security and Privacy*. Oakland, CA, 143–157.

CAMP, J., CRANOR, L., FEAMSTER, N., FEIGENBAUM, J., FORREST, S., KOTZ, D., LEE, W., LINCOLN, P., PAXSON, V., REITER, M., RIVEST, R., SANDERS, W., SAVAGE, S., SMITH, S., SPAFFORD, E., AND STOLFO, S. 2009. Data for cybersecurity research: Process and "wish list". `http://www.gtisc.gatech.edu/files_nsf10/data-wishlist.pdf`.

CHATFIELD, C. 1983. *Statistics for Technology: A Course in Applied Statistics*, 3rd ed. Chapman & Hall/CRC.

COVA, M., LEITA, C., THONNARD, O., KEROMYTIS, A. D., AND DACIER, M. 2010. An analysis of rogue AV campaigns. In *International Symposium on Recent Advances in Intrusion Detection*. Ottawa, Canada, 442–463.

CWE/SANS. 2010. Top 25 most dangerous programming errors.

DEAN, J. AND GHEMAWAT, S. 2004. MapReduce: Simplified data processing on large clusters. In *USENIX Symposium on Operating Systems Design and Implementation*. San Francisco, CA, 137–150.

DEWITT, D. J. 1993. The Wisconsin benchmark: Past, present, and future. In *The Benchmark Handbook for Database and Transaction Systems*, J. Gray, Ed. Morgan Kaufmann.

DHS. 2011a. DETER. `http://www.isi.deterlab.net/`.

DHS. 2011b. PREDICT. `http://www.predict.org/`.

EIDE, E., STOLLER, L., AND LEPREAU, J. 2007. An experimentation workbench for replayable networking research. In *USENIX Symposium on Networked Systems Design and Implementation*. Cambridge, MA.

FREI, S. 2009. Security econometrics: The dynamics of (in)security. Ph.D. thesis, ETH Zürich.

GOOGLE INC. 2011. Google Apps service level agreement. `http://www.google.com/apps/intl/en/terms/sla.html`.

GRIFFIN, K., SCHNEIDER, S., HU, X., AND CHIUEH, T.-C. 2009. Automatic generation of string signatures for malware detection. In *International Symposium on Recent Advances in Intrusion Detection*. Saint-Malo, France, 101–120.

KEETON, K., MEHRA, P., AND WILKES, J. 2009. Do you know your IQ? A research agenda for information quality in systems. *SIGMETRICS Performance Evaluation Review 37*, 26–31.

LEITA, C., BAYER, U., AND KIRDA, E. 2010. Exploiting diverse observation perspectives to get insights on the malware landscape. In *International Conference on Dependable Systems and Networks*. Chicago, IL, 393–402.

LIPPMANN, R. P., FRIED, D. J., GRAF, I., HAINES, J. W., KENDALL, K. R., MCCLUNG, D., WEBER, D., WEBSTER, S. E., WYSCHOGROD, D., CUNNINGHAM, R. K., AND ZISSMAN, M. A. 2000. Evaluating intrusion detection systems: The 1998 DARPA off-line intrusion detection evaluation. *DARPA Information Survivability Conference and Exposition,*, 12–26.

MAXION, R. A. AND TOWNSEND, T. N. 2004. Masquerade detection augmented with error analysis. *IEEE Transactions on Reliability 53,* 1, 124–147.

MCHUGH, J. 2000. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security 3,* 4, 262–294.

MILLER, B. P., FREDRIKSEN, L., AND SO, B. 1990. An empirical study of the reliability of UNIX utilities. *Communications of the ACM 33,* 12 (Dec), 32–44.

PAVLO, A., PAULSON, E., RASIN, A., ABADI, D. J., DEWITT, D. J., MADDEN, S., AND STONEBRAKER, M. 2009. A comparison of approaches to large-scale data analysis. In *ACM SIGMOD International Conference on Management of Data*. Providence, RI, 165–178.

PAXSON, V. 2004. Strategies for sound internet measurement. In *Internet Measurement Conference*. Taormina, Italy, 263–271.

PERKINS, J. H., KIM, S., LARSEN, S., AMARASINGHE, S., BACHRACH, J., CARBIN, M., PACHECO, C., SHERWOOD, F., SIDIROGLOU, S., SULLIVAN, G., WONG, W.-F., ZIBIN, Y., ERNST, M. D., AND RINARD, M. 2009. Automatically patching errors in deployed software. In *ACM Symposium on Operating Systems Principles*. Big Sky, Montana, USA, 87–102.

PHAM, N. H., NGUYEN, T. T., NGUYEN, H. A., AND NGUYEN, T. N. 2010. Detection of recurring software vulnerabilities. In *IEEE/ACM International Conference on Automated Software Engineering*. Antwerp, Belgium, 447–456.

ZAHARIA, M., KONWINSKI, A., JOSEPH, A. D., KATZ, R. H., AND STOICA, I. 2008. Improving MapReduce performance in heterogeneous environments. In *USENIX Symposium on Operating Systems Design and Implementation*. San Diego, CA, 29–42.