The Provenance of WINE

Tudor Dumitraș
Symantec Research Labs
tudor_dumitras@symantec.com

Petros Efstathopoulos
Symantec Research Labs
petros_efstathopoulos@symantec.com

ABSTRACT

The results of cyber security experiments are often impossible to reproduce, owing to the lack of adequate descriptions of the data collection and experimental processes. Such provenance information is difficult to record consistently when collecting data from distributed sensors and when sharing raw data among research groups with variable standards for documenting the steps that produce the final experimental result. In the WINE benchmark, which provides field data for cyber security experiments, we aim to make the experimental process *self-documenting*. The data collected includes provenance information—such as *when*, *where* and *how* an attack was first observed or detected—and allows researchers to gauge information quality. Experiments are conducted on a common testbed, which provides tools for recording each procedural step. The ability to understand the provenance of research results enables rigorous cyber security experiments, conducted at scale.

Keywords: *Provenance, cyber security, experimental research, dependability benchmarking, WINE*

I. INTRODUCTION

The independent verification of experimental results represents one of the basic tenets of science. Any scientific study drawing conclusions through the collection, observation and processing of data is expected to provide adequate information on its data sources and experimental methods, which enables reasoning about potential threats to validity. Such information also helps other scientists reproduce the experimental results, increasing the confidence of the scientific community in the findings of the study. Although various means have been employed in the past for the documentation of experiments, nowadays most scientific data collection and processing is performed using computer systems. Consequently, the problem of recording meta-information about scientific experimentation has become an important challenge for computer scientists—both for the purposes of their own research as well as for facilitating the needs of the scientific community. Effectively collecting and maintaining the necessary types and amount of *provenance* (or lineage) information about data produced and analyzed in scientific experiments is key to assessing the threats to validity and to ensuring the reproducibility of results.

Cyber security research relies on the observation and analysis of events on a global scale: malware behavior on the Internet, infection patterns, surges of zero-day attacks, changes in attack surfaces, reassessments of the severity of known vulnerabilities, etc. For example, Symantec collects data from millions of sensors—e.g., anti-virus products, intrusion-detection systems, honeypot deployments, spam filtering appliances, decoy email accounts—distributed worldwide. The lack of centralized provisioning and control for the environments where all these sensors operate makes it difficult to document the data collection process and to record provenance information consistently. Additionally, the ethical, legal and scientific challenges for publicly disseminating security-related data sets have prevented, so far, the establishment of a representative corpus of data for cyber security experimentation and benchmarking. For example, security techniques operate on sensitive code and data, such as dangerous binaries (e.g., malware) and data that could reveal personally identifiable information (e.g., hosts that have been compromised by attackers). In consequence, the data sets used for validating cyber security research are not usually available to the wider research community, and the experimental results seldom receive independent verification. Moreover, because the cyber threat landscape changes frequently, the lack of provenance information on the data sets that are currently available to the research community makes it difficult to relate benchmarking results to the behavior that can be expected when deploying the system-under-test in the field.

Documenting experimental processes can be equally problematic. Legal restrictions may prevent researchers from reporting their exact methods (e.g., intellectual property or security reasons). In other cases, there is limited control over the transformation workflow: when processing is performing using "black box" transformations and third-party software, the stream of experimental provenance information is interrupted and partial. Finally, more often than one would hope, provenance information is unavailable due to plain negligence. Without full, end-to-end control of the experimental process and a trusted environment to operate in, it is hard for researchers to collect adequate provenance information and report it safely.

We have built the Worldwide Intelligence Network Environment (WINE) [1], which assembles multiple data sets,

collected and curated by Symantec Research Labs, covering the entire lifecycle of cyber attacks. WINE represents a benchmark for cyber security and enables experimentation across a broad spectrum of disciplines (e.g., machine learning, visual analytics, software engineering). Our main design goal for WINE is to ensure the *reproducibility of experimental results*, which requires a provenance approach adapted to the unique needs of our system.

WINE could benefit from the recent efforts to formalize the provenance problem in computer systems. For example, the Open Provenance Model (OPM) [2] aims to provide a standardized specification to describe "any thing, whether produced by computer systems or not", as well as a set of interfaces that could be used for that description and the exchange of provenance information among applications. Multi-layered mechanisms for recording provenance information have been proposed [3], including provenance for storage systems [4], but these mechanisms require extensive changes to the software stack—including the operating system. Other candidate approaches for WINE are techniques, studied in the context of database schema evolution, that indicate whether data transformations are invertible and information-preserving [5] and mechanisms for capturing causality in distributed systems [6], [7]. Although each of the proposed approaches has its merits, we believe that the *modus operandi* of WINE lends itself to a simple, yet effective, provenance model: we advocate the use of *self-documenting* data within the confines of the WINE system.

WINE is a self-contained research platform, where we have complete, end-to-end oversight of all operations—except for the initial data recording, performed on millions of end-hosts around the world. Data collection and curation are under our control and so are data views presented to the experimenters. The data that we collect includes provenance information, such as *when* an attack was first observed, *where* it has spread from there and *how* we were able to detect it. While certain attributes of the data are not reliable, such as the timestamps assigned on the end-hosts where we collect the data, WINE allows researchers to gauge the *information quality* for each data record, e.g., by comparing the timestamps assigned on the collection host and on our submission gateway. Moreover, experimentation is performed in a controlled environment at Symantec Research Labs, and all intermediate/final results are at all times kept within the administrative control of the system. For gathering the needed provenance information, and for security reasons as well, exporting and importing data is performed through strictly defined and tightly controlled channels, and experimental workflows can be assembled using a specific set of tools and infrastructure.

Operating in such a predefined, controlled environment allows us to have full knowledge of the experimentation steps and to record all the activities necessary when repeating the experiment. Moreover, this operational model allows us to address the main challenges for sharing security-oriented data sets [1]. In particular, we do not create a malware library for anyone to download at will, and we ensure that private information is not disseminated in public. All the experiments conducted on WINE can be attributed to the researchers who conducted them and the raw data provided cannot be accessed anonymously or copied outside of Symantec's network.

In consequence, we do not focus on developing a generalized provenance model capable of describing "any thing" (since we possess thorough descriptions of our data sets), or a solution that would require invasive changes to applications or the underlying software infrastructure. Instead, our goal is to gather all the information required for ensuring the reproducibility of experiments conducted on WINE.

This paper makes three contributions:

- Drawing on our experiences from building a real system for computer science experimentation, we present an approach for self-documenting provenance.
- We describe the challenges for automatically collecting the provenance information required for reproducing experimental results, and we propose a complementary technique, inspired from the long-standing practice of maintaining a lab book in the experimental scientific disciplines.
- We discuss the benefits of extending these approaches with the ability to exchange provenance information with third-party tools and with mechanisms for assessing the quality of information.

The remainder of this paper is organized as follows: in Section II we provide a precise definition of the problem that we address in this paper. Section III gives an overview of the WINE system. Section IV describes the self-documenting provenance model we are proposing for WINE, while Section V discusses some further concerns and potential for improvements to the current design.

II. Problem statement and goals

Scientific data provenance generally refers to the recording and querying of all data collection operations, transformations, processing, filtering and interpretation of intermediate results in a scientific work flow. In the context of our system, we define provenance as all the information required to reproduce the results of past experiments in WINE.

We distinguish between two types of provenance. The *data set provenance* provides information about the data used for experimentation, including the initial data sources, the collection methods and context, timings, etc. Because the focus of WINE—and of computer science, in general—is to develop and improve the tools used for analyzing and processing the data, rather than to use off-the-shelf software tools in a predictable manner, we also define the *experimental provenance*. This information describes the analysis tools, the experimental methods, the intermediate results

Table I
THE WINE DATA SETS AND SOME OF THEIR ATTRIBUTES, THAT HELP ANSWER PROVENANCE QUESTIONS SUCH AS *when* AN EVENT WAS FIRST OBSERVED, *where* WAS THE DATA COLLECTED AND *how* WAS THE EVENT DETECTED.

Data set	Sources	Description	Provenance attributes
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.	• Submission timestamp • URL, Host ISP, geolocation • Format version, known bugs
A/V telemetry	130 million machines	Occurrences of known threats, for which Symantec has created signatures and which can be detected by anti-virus products.	• Submission timestamp • Host ISP, geolocation • Scanning engine, version of A/V signature definitions
Email spam	2.5 million decoy accounts	Samples of phishing and spam emails, collected by Symantec's enterprise-grade systems for spam filtering.	• Received timestamps • FROM domains, geolocation • Keywords, X-Spam headers
URL reputation	10 million domains	Website-reputation data, collected by crawling the web and by analyzing malicious URLs.	• Crawl timestamp • URL
Malware samples	200 countries	A collection of packed and unpacked malware samples (viruses, worms, bots, etc.), used for creating Symantec's A/V signatures.	• Collection date • Collection source

and the data transformations that lead to a research result. Our goal is to ensure the reproducibility of experiments, regardless of updates to the WINE data sets, of changes to our data collection process or of the effects of software evolution on the analysis tools and on the experimental environment. In this paper, we *characterize the provenance information* required for ensuring reproducibility and we *identify potential sources* for such information.

Non-goals. We do not attempt to build a general-purpose provenance system, and, like most of the prior work on this topic, we do not focus on how to *enforce* the consistent production of provenance information. Instead, we assume that WINE users consent to our aim of experimental reproducibility and *cooperate* with our efforts to record provenance information. While we develop automated mechanisms for recording such information, we do not try to render these mechanisms tamper-proof.[1] Moreover, mechanisms for querying the provenance information, for manipulating recorded workflows (e.g., repeating an experiment but changing one individual step) and for exchanging data with third-party provenance tools are within the scope of the WINE project, but are not covered in this paper as they are the focus of ongoing standardization efforts [2].

III. THE TRUTH IS IN WINE

The Worldwide Intelligence Network Environment (WINE) represents our attempt to define a rigorous benchmark for cyber security [1]. WINE provides 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. Unlike the existing testbeds for experimental cyber security (e.g. DETER [8]), the main purpose of WINE is not to simulate cyber attacks or to replay attacks observed in the wild. Instead, WINE enables benchmarking and data analysis at scale, by providing representative field data collected at Symantec and assembled in five data sets: binary-reputation, email-spam, URL-reputation, A/V telemetry and malware samples.

Symantec collects this data from a multitude of sensors, distributed worldwide (see Table I), and uses it in its day-to-day operations. The binary reputation data set provides information on unknown binaries (i.e., files for which an anti-virus signature has not been created) that are downloaded by users who opt in for Symantec's reputation-based security program. The history of binary reputation submissions can reveal when a particular threat has first appeared, as a zero-day attack, and for how long it has existed in the wild before it was detected. The anti-virus telemetry records occurrences of known threats, for which Symantec has created signatures and which are detected by anti-virus products. This data set includes intrusion-detection telemetry. The spam data set includes samples of spam and phishing emails, as well as statistics on the messages blocked by the Symantec's spam filters. The URL reputation data is gathered by crawling the web and by interacting with malicious web sites.[2] The malware collection includes representative samples of both packed and unpacked malware (e.g., viruses, worms, bots), which are used for creating Symantec's anti-virus signatures.

Rather than collect provenance information separately, we chose to make the WINE data *self-documenting*: each record in the data sets from Table I includes attributes that provide clues about its provenance. For example, a binary reputation submission about a file being downloaded also includes a

[1] We do, however, put in place fault-tolerance mechanisms (e.g., RAID, backups, source code versioning) to ensure that the researcher's code is stored reliably.

[2] Norton Safeweb (http://safeweb.norton.com/) provides a simplified interface for querying the URL reputation data.

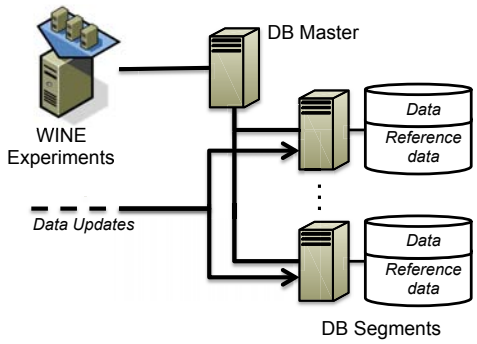

Figure 1. The WINE experimental platform.

timestamp, which indicates *when* the data was collected. The URL of the file and the Internet Service Provider of the host receiving the download indicate *where* the potential attack came from and *where* the data collection occurred. The version of the data format used in the submission protocol, which indicates *how* the data was collected, allows the experimenter to account for changes in the data exchange mechanisms (e.g., the use of newer hash functions to identify files) and for known bugs in WINE.

In addition to these data sets, WINE provides an analysis platform designed for experimental reproducibility. The researchers accessing the WINE data sets upload their analysis scripts to an isolated virtualized environment (see Figure 1). With the exception of the malware data set, the WINE data is stored in a parallel database, in append-only tables that can be queried from a virtual machine using either ANSI SQL or MapReduce tasks [9], for greater flexibility. This platform enables data-intensive applications by adopting a *shared-nothing* architecture, which partitions data across multiple storage nodes, attached directly to the hosts that execute data analysis tasks. The in-database query optimizer tries to place the analysis tasks directly on the nodes that already store the data required. The malware data set is stored and analyzed in an isolated *red lab*, which does not have inbound or outbound network connectivity, in order to prevent viruses and worms from escaping this environment. The results from malware experiments can be correlated with the information in the other WINE data sets by recording the SHA-2 or MD5 hashes of the malware samples analyzed.

The WINE benchmark is available to both internal experts and to the research community at large, and it provides the opportunity for conducting research in a broad spectrum of disciplines, such as cyber security, software reliability, machine learning, visual analytics, etc. WINE must be accessed on-site at Symantec Research Labs,[3] in order to protect the sensitive information in the data sets, and to ensure experimental reproducibility. The WINE data sets receive regular updates from Symantec's collection of sensors, distributed worldwide, which ensures that the data reflects the current cyber threat landscape. Researchers using WINE have read-only access to the raw data collected in this manner, and they start their experiments by defining *reference data sets* based on the raw data available. When an experiment is completed, we archive the corresponding reference data sets and the virtual machines that contain the analysis code, for preserving the ability to reproduce the results in the future and to compare them against newer techniques.

[3]More information on how to access WINE is available at http://www.symantec.com/WINE.

IV. Self-Documenting Provenance

Consider, for example, a hypothetical experiment that seeks to evaluate a novel technique for detecting *zero-day attacks*, which exploit vulnerabilities that are not acknowledged publicly. Usually, such vulnerabilities are not disclosed either because the software vendor is in the process of developing patches or because they remain known only to the hacker who has discovered them. To assess the precision and recall of the new detection technique, the experimenter would use the following hypothetical procedure:

1) Start by identifying a set of binaries associated with known attacks, e.g., by analyzing samples from the malware data set or the A/V telemetry submissions.
2) Identify when the binaries first appeared on the Internet by examining the binary-reputation submissions.
3) Correlate data with an external source of vulnerability disclosure and patch release dates (e.g., the National Vulnerability Database [10]) and determine the time window when each vulnerability was undisclosed,
4) Based on this information, divide the malicious binaries of step 1 into two reference data sets: (i) zero-day attacks and (ii) attacks against known vulnerabilities.
5) Finally, the experimenter would test the new technique using only the information that was available while the vulnerability remained undisclosed. The experimenter could use the attacks against known vulnerabilities to measure the rate of false positive warnings.

To reproduce these results in the future, and maybe investigate the effects of altering individual steps in the experimental procedure, we must fully understand each step, from the data collection to the final experimental results. Additionally, this provenance information allows the experimenter to manage the selection bias and to determine the real-world situations that the reference data sets are representative of.

WINE Data Set Provenance. As explained in Section III, the WINE data is self-documenting because it provides information about when, where and how the data was collected. However, the WINE experiments query a database that receives data through the pipeline illustrated in Figure 2. The WINE data pipeline converts raw data received from the sensors distributed worldwide into a format suitable for

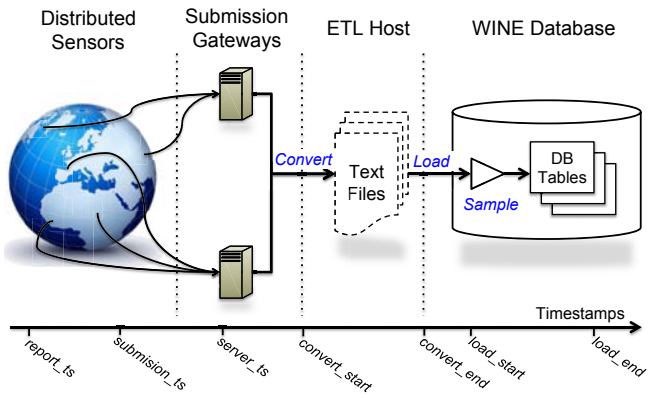

Figure 2. The WINE data pipeline.

loading into the database, and selects a sub-sample of all the records that Symantec has gathered in the field. This process preserves all the attributes of the raw submissions and does not transform or aggregate the data in any way. To maintain the self-documenting provenance property, we use the WINE database to manage the entire data pipeline: we determine the set of raw submission files that have not been loaded or converted yet, and we use database transactions to ensure the mutual exclusion among multiple convert and load processes executing in parallel. Finally, we perform sampling by defining database views over the loaded data, before populating the tables in the public WINE schema. This database schema, augmented with information about each stage in the WINE data pipeline, is accessible to all the WINE experiments. The provenance information produced by the data pipeline includes the sampling views, the raw submission files that have been loaded and the ones that are yet to be processed, the attributes (e.g., name, path, size) of the submission file that originated each row in the WINE tables, the timestamps for when the submission was received, when the file was converted and when it was loaded, the number of rows loaded in the database and the errors encountered during the convert and load steps.

A subtle point is that, to maintain the self-documenting property, all the input data—including data from external sources—must be available locally at the start of the experiment. If the analysis tool interacts with Web services, resolves domain names, queries external databases, etc., during the experiment, the results would not be reproducible in the future because of the transient nature of these Internet resources. Therefore, the WINE experiments have access only to the WINE database and platform, and are otherwise isolated from Symantec's corporate network and from the Internet. In our example, the hypothetical experimenter would first download the list of vulnerabilities into the virtual machine used to conduct the experiment (or, provide us with a script to download the data), and would query this data locally, in order to correlate it with the WINE data sets. This allows us to archive all the input data used in the experiment, ensuring future reproducibility.

WINE Experimental Provenance. To establish the experimental provenance, we collect the tools used to analyze the data. We do not strive to maintain an unchanged environment in order to keep all experimental results *comparable*; for example, in the future we may improve the experimental platform by upgrading the CPU, the memory, the operating system, etc. Instead, we aim to ensure that experiments remain *reproducible*, which similarly empowers the experimenters to compare their results against the prior art. To this end, we require that each experimenter develops a script that runs the experiment end-to-end. Because WINE does not provide mechanisms for transferring data outside the system, we execute this script and we provide the final results to the experimenter. We preserve the analysis tools and scripts by archiving the virtual machine used to conduct the experiment (see Figure 1). As a sanity check, we also record the interactive terminal sessions and the database queries issued during the experiment.

However, recording all the mechanical steps in an experiment is not enough. To reproduce a researcher's conclusions, we must understand the hypothesis and the reasoning behind each experimental step. We achieve this by providing experimenters with an electronic *lab book* (in the form of a wiki) for documenting all the experimental procedures. Maintaining the lab book will require a conscious effort from the experimenter. For example, in the case of our hypothetical experiment on detecting zero-day attacks, the experimenter would have to document the precise definition of a zero-day attack and the metrics used to determine if the experiment was successful. The lab book would also describe the reference data sets, the script that executes the experiment and the output data that we must provide. Keeping such a lab book is a common practice in other experimental fields, such as applied physics or cell biology.

V. Discussion

Because the data collection and curation are under our control, WINE lends itself to the implementation of a self-documenting provenance model. However, our notion of self-documentation is not synonymous with automatic provenance: WINE users must be well-intended and must participate in the process by adhering to the provenance guidelines of the system. This is a reasonable assumption to make, since maintaining these self-documenting properties is in the experimenters' interest as well. Our provenance approach for WINE is a design decision aiming to provide an efficient provenance model that would be less intrusive to the system and its users than the generic provenance models that have been proposed, e.g., OPM [2].

In certain cases, however, it would be useful to express or export provenance data in a standardized format, compatible with other systems. In the future, we can provide WINE

experimenters with an annotation language able to express the hypothesis of a particular experiment, and map it to specific transformations, data objects and code fragments from the scripts that drive the experiment. Thus, researchers will be able to use simple expressions to annotate their code, which could then be translated to a standardized format, such as OPM, using tools provided by the WINE platform. For example, WINE does not currently provide an easy way of re-running experiments automatically and altering individual steps—which is one of OPM's main motivations. By exporting the provenance information in a standardized format, experimenters will be able to use off-the-shelf tools for auditing and manipulating the provenance workflows.

Additionally, although data provenance information gathered using the proposed mechanism will support experimental reproducibility, it may not be enough for assessing the validity of results. In particular, data provenance information may attest to the lineage of data, but may not always indicate which data records are relevant for an experimental hypothesis. However, in many large scale collections uncertainty about the data is explicit. For example, with the use of heuristics and machine-learning techniques for detecting polymorphic malware, the labels applied to binaries are no longer a black-and-white determination, but, rather, they express a certain level of confidence regarding the binary's hygiene. In a commercial product, where monitoring and logging represent secondary concerns, the submissions are throttled, truncated and, in the case of WINE, sampled, in over to reduce the load on users' machines and bandwidth costs. Moreover, the hash functions used for identifying binaries may change, as products evolve, and the techniques used for identifying user machines are not always reliable.

We designed WINE to keep track of all these issues, in order to give experimenters the opportunity to assess the quality of information in their reference data sets. For example, each submission record has timestamps assigned at each step in the data pipeline of Figure 2: on the collection host (when the event is recorded and when the submission occurs), on the submission gateway (when the submission is received), on the loading host (when conversion starts and ends) and in the database (when loading starts and ends). These timestamps allow experimenters to assess, for example, whether the clock on one host in the pipeline is inaccurate. In the future, we would like WINE to provide additional information, capable of constituting a basic measure of information quality.

VI. Conclusion

In this paper, we describe the provenance challenges encountered while designing the WINE benchmark for cyber security. WINE makes field data, collected worldwide by Symantec, available to external researchers, and it provides an analysis platform designed to enable reproducible experimentation. Both the data collection process and the experimental platform are under our control, giving us the opportunity to gather diverse and reliable provenance information on the origins of the data and on the experimental workflows. We aim to make the data collection and the experimental processes self-documenting, by analyzing the data attributes to extract provenance information, and by recording additional information (e.g., timings, errors) on each data processing step. The data attributes provide insight regarding where, when and how the data was collected, while the data processing records enable experimenters to assess the information quality. However, reproducible experimentation in WINE cannot be achieved exclusively through automated provenance collection: replicating one's conclusions requires understanding the hypothesis and the reasoning behind each experimental step. To complement our self-documenting approach, we adopt a standard practice from experimental disciplines outside the realm of computing: WINE provides a lab book, for documenting experimental design and methods.

References

[1] T. Dumitraș and D. Shou, "Toward a standard benchmark for computer security research: The Worldwide Intelligence Network Environment (WINE)," in EuroSys BADGERS Workshop, Salzburg, Austria, Apr 2011.

[2] "The OPM Provenance Model," http://openprovenance.org/.

[3] K.-K. Muniswamy-Reddy, U. Braun, D. A. Holland, P. Macko, D. Maclean, D. Margo, M. Seltzer, and R. Smogor, "Layering in provenance systems," in USENIX Annual Technical Conference, San Diego, California, Jun 2009.

[4] K.-K. Muniswamy-Reddy, D. A. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in USENIX Annual Technical Conference, Boston, Massachusetts, Jun 2006.

[5] C. A. Curino, H. J. Moon, and C. Zaniolo, "Graceful database schema evolution: the PRISM workbench," in International Conference on Very Large Data Bases (VLDB), Auckland, New Zealand, Aug 2008.

[6] E. Thereska, M. Abd-El-Malek, J. J. Wylie, D. Narayanan, and G. R. Ganger, "Informed data distribution selection in a self-predicting storage system," in International Conference on Autonomic Computing, Dublin, Ireland, Jun 2006.

[7] R. Fonseca, G. Porter, R. H. Katz, S. Shenker, and I. Stoica, "X-Trace: A pervasive network tracing framework," in Symposium on Networked Systems Design and Implementation (NSDI), Cambridge, MA, Apr 2007.

[8] DHS, "DETER," 2011, http://www.isi.deterlab.net/.

[9] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in USENIX Symposium on Operating Systems Design and Implementation, San Francisco, CA, Dec 2004, pp. 137–150.

[10] National Institute of Standards and Technology, "National Vulnerability Database Version 2.2," http://nvd.nist.gov.