CSET ’18
The 11th USENIX Workshop on Cyber Security Experimentation and Test

PETER A. H. PETERSON

Peter A. H. Peterson is an Assistant Professor of Computer Science at the University of Minnesota, Duluth, where he teaches and researches operating systems and security, with a focus on R&D to make security education more effective and accessible. He received his PhD from the University of California, Los Angeles, for work on “Adaptive Compression”—systems that make compression decisions dynamically to improve efficiency. He served as the Co-Chair of CSET ’18. pahp@d.umn.edu

The 11th USENIX Workshop on Cyber Security Experimentation and Test (CSET ’18) was held in Baltimore, Maryland, on Monday, August 13th, 2018—one of the co-located workshops preceding the 27th USENIX Security Symposium. The CSET Call for Papers “invites submissions on cyber security evaluation, experimentation, measurement, metrics, data, simulations, and testbeds.” In other words, CSET emphasizes tools and methods in the service of computer security testing and research, making it somewhat unique. This year, our program consisted of 10 papers in four themed sessions and a panel. As usual, discussion was friendly and lively.

Data and Guidance was our first session, chaired by Christian Collberg. Josiah Dykstra presented the methodology and design behind the Cyber Operations Stress Survey (COSS), an instrument for measuring the effects of tactical cyber operations on the stress of operators. Stress, in the context of the COSS, is represented in terms of an operator’s fatigue (measured using the Samn-Perelli Fatigue Scale, or SPFS), frustration, and cognitive workload (using the NASA Task Load Index, or TLX). Other contextual factors about an operation, such as team synergy and duration of the operation, are also captured. The COSS was used in four studies of tactical cyber operations with consistent results, demonstrating that the method was internally and externally reliable. The results of the project were presented in terms of lessons learned through the development and use of the COSS tool, including how low initial enthusiasm was mitigated by presenting initial results to operators to show their value, and how results led to policy changes. The paper also described how some of the contextual factors were chosen through pilot testing; one surprising result was that caffeine had no relationship to stress!

Next, Tyler Moore presented “Cybersecurity Research Datasets: Taxonomy and Empirical Analysis,” a work that analyzed 965 recent papers for information about how the authors used, created, or shared data sets, and how those choices correlated to other factors, such as whether new data sets were shared and the citation count of the papers. They also created a taxonomy based on their observations. While the community values sharing data sets as a service, researchers may be reluctant to do so for privacy, competitive advantage, or other reasons. The authors’ results suggest that there is a self-interested incentive to publish data sets—citation counts. The papers in their pool that created and released data sets received 42% more citations than papers that did not use data sets or used only public data, and 53% more citations per year than papers that created data sets but did not release them. Thus, in addition to altruism, authors may want to consider whether they would benefit more from citations than they might suffer due to competition.

After this, Samuel Marchal from Aalto University in Finland discussed “On Designing and Evaluating Phishing Webpage Detection Techniques for the Real World,” which explores the state of phishing detection research, raising concerns about the effectiveness of detection strategies in terms of detection performance, temporal resilience, deployability, and usability. Too much focus, they said, is placed on numerical accuracy in an artificial environment, without enough consideration of realism, practicality, change in accuracy over time, or comparability between experiments. For non-phish ground-truth examples, they suggest incorporating low- and high-popularity sites and using the full URLs that a user would see. For phish ground-truth examples, they recommend manual inspection to ensure that the data consists only of unique, legitimate phishes. For both, they recommend using a diverse set of current sites (not old data), languages, and types of sites. Older data should be used for training, while newer should be used for testing, and test input should have a realistic ratio of phishing to non-phishing samples. They also recommend longitudinal studies with updated data to evaluate accuracy over time, and that systems should be tested against adversarial machine-learning techniques. Their hope is that, by applying these guidelines to the evaluation of future phishing detection techniques, researchers might obtain meaningful and comparable performance results as well as fostering technology transfer.

The New Approaches session, chaired by Sven Dietrich, focused on novel approaches to experimentation. "DEW: Distributed Experiment Workflows" was presented by Genevieve Bartlett. DEW is a high-level approach to experiment representation that separates the infrastructure of a testbed experiment—the testbed hardware, software, and experimental network—from the observable behavior of the experiment in operation. The goal of DEW is to allow researchers to describe what an experiment should do, allowing DEW and the underlying testbed infrastructure to instantiate that behavior in the manner appropriate for that testbed. Meant to be human-readable, a DEW "program" consists of a scenario describing the general behavior of the experiment in terms of its actions, bindings implementing those actions (e.g., scripts), and constraints defining the required properties of the experiment's components (enforced by DEW), such as network links and computational nodes. The authors intend to produce translators that generate DEW based on an experiment setup, and generators that create an experiment setup from a DEW definition. The DEW paper includes a number of potential benefits, code samples, and a discussion of their NLP-enhanced GUI. The authors welcome communication and feedback from interested parties.

Xiyue Deng presented "Malware Analysis through High-Level Behavior," a paper that describes the Fantasm framework and some results from its use. Fantasm identifies malware type by recognizing patterns in a given malware's network behavior. After all, malware that is supposed to perform scans, exfiltrate keystrokes, or send spam needs to perform those activities, even if the binary is obfuscated. While malware can often detect and thwart monitoring performed through debuggers or virtualization, Fantasm avoids detection by running the malware on real bare-metal systems on DeterLab. Fantasm controls the malware execution on one host and monitors the network using a separate remote host, labeling malware samples based on their observed behavior. As Fantasm monitors network behavior, it can automatically decide whether to impersonate the receiver, forward the message to the original endpoint, or drop the traffic. Their paper details a number of challenges in addition to future work.

Next, Pravein Govindan Kannan told us about "BNV: Enabling Scalable Network Experimentation through Bare-metal Network Virtualization." BNV is a network hypervisor (based on OpenVirtex) that works in conjunction with specially configured switches to allow high-fidelity testing of a variety of large-scale network topologies while using a fraction of the hardware. BNV is a response to the desire to evaluate complex datacenter topologies with high accuracy, without having to physically construct those topologies. Existing approaches for topology testing are often either virtualized (e.g., Mininet or NS) or not flexible enough for their purposes (e.g., CloudLab, DeterLab); for example, BNV can change topologies within a few seconds. To provide for a large number of high-fidelity virtual switches, BNV "slices" physical switches into virtual switches and provides connectivity between these switches using physical loopback links. In this way, BNV can emulate 130 switches and 300 links with just five switches. BNV was evaluated by testing various topologies in both virtual and physical forms under a workload and comparing their performance, with good results. Currently, BNV is in use at the National Cybersecurity Lab (NCL) in Singapore.

After lunch, we held a round-table discussion on "Opportunities and Challenges in Experimentation and Test." In a sense, CSET is a workshop for research that produces tools, data, or guidance that enhances security research. While important, research of this type sometimes doesn't fit neatly into calls for papers or funding proposals. We thought it would be helpful to have experts in the field talk about their experiences and provide advice about doing work in this area. Our panelists—Terry Benzel (USC/ISI), Eric Eide (Utah), Jeremy Epstein (NSF), Simson Garfinkel (US Census Bureau), and Laura S. Tinnel (SRI)—and several members of the audience had a lively discussion on that subject. A full summary of that discussion is not possible here, but a key observation supported by many was that research of this type can often be funded as part of a larger project, where the tool in question is necessary for the research. Similarly, interesting and useful CSET-style papers can often be created by taking a previously created research system and polishing it or extending it to make it useful for others. In any case, papers should not simply be whitepapers describing the system or its creation saga, but should include an evaluation of some kind; this can include new results produced by the system, user feedback, or a validation of the system.

Eric Eide chaired the session on Shiny New Testbeds. Vitaly Ford got things started by telling us about "AMIsim: Application-Layer Advanced Metering Infrastructure Simulation

Framework for Secure Communication Protocol Performance Evaluation." Advanced Metering Infrastructure, or AMI, refers to "Smart Grid" infrastructure devices, such as smart meters, that allow for two-way communication using ZigBee or other methods. Because this wireless communication includes sensitive information about power consumption, security and privacy are of utmost importance. At the same time, the Smart Grid has fairly rigid communication and computation constraints. Building your own mini Smart Grid for research is too expensive for most researchers. Several Smart Grid simulation frameworks exist, but these focus largely on modeling electricity flow and power distribution hardware, not on the communication and computation resources necessary to test security and privacy-preserving Smart Grid protocols. AMIsim fills this gap. Based on OMNet++, AMIsim's performance is timed on a modern PC and then scaled to represent the approximate time a smart meter would take to perform the same computation. In this way, AMIsim opens the door for researchers to design and evaluate security-focused Smart Grid communication protocols in a way that was not previously available. AMIsim is under continuing development.

Our next new testbed was described in the paper "Galaxy: A Network Emulation Framework for Cybersecurity." Kevin Schoonover and Eric Michalak presented this work, which describes Galaxy, a new high-fidelity, emulation-based network testbed supporting quick, flexible, and parallel experimentation. Galaxy includes built-in logging to store results, and strongly isolates experiments so that state from one experiment does not affect another. While Galaxy could be useful for many purposes, it was specifically designed for evolutionary experiments requiring many iterations in succession, something that would be very time-consuming on a reconfigurable physical testbed, like Emulab or DeterLab. Galaxy takes configuration files describing a topology, and instantiates the network using bridging and virtual nodes using *vmbetter*. The snapshot feature of *libvirt* is used to ensure that no state persists between experiments, and the Ansible automation tool is used to instantiate topologies in parallel on a set of distributed computers. With these tools, topologies can be reverted and restarted very quickly and have high fidelity to real network behavior (two properties essential for valid evolutionary algorithm use). Their paper includes a case study using Galaxy as part of the CEADS-LIN project for developing attacker enumeration strategies through evolutionary algorithms, in addition to various enhancements planned for the tool.

The final session of CSET '18, Testbed Enhancements, was chaired by David Balenson. The first paper, "Supporting Docker in Emulab-Based Network Testbeds," was presented by Eric Eide. Given the popularity of Docker, it makes sense for Emulab to support Docker containers as "first class" nodes; it means that popular Docker-based tools used in research and industry can be easily integrated into Emulab experiments. A major challenge for this project was preserving both the Docker and Emulab experiences—testbed users should have the sense that their Docker containers and Emulab experiments "just work." This includes being able to use containers without manual intervention, but with the traffic shaping capabilities, logging, and command-line access common to traditional Emulab nodes. Emulab supports these features by automatically modifying Docker containers (in real time if necessary) to support essential capabilities. This process works well; in their experiments, 52 of the 60 most popular Docker containers could be automatically modified for use with Emulab. These enhancements are available now on Emulab.

The final paper at CSET '18, "High Performance Tor Experimentation from the Magic of Dynamic ELFs," was presented by Justin Tracey. For a variety of good reasons, experimentation on the live Tor network is discouraged. Instead, the recommended approach is to use a testbed of some kind, be it simulated, emulated, or physical. Shadow is a popular discrete-event simulator used for Tor and other types of research. It is popular, in part, because it runs real application code rather than models of the application being tested, the latter of which can lead to mistakes due to user error or when the documentation of a system is inconsistent with the software artifact in actual use. Unfortunately, discrete-event simulators often have the drawback of significant overhead. Tracey et al.'s work eliminated two major bottlenecks from Shadow by doing away with a global logging lock and by creating a new loader that enables more efficient use of CPU resources, in addition to removing restrictions on library and compiler use. In an evaluation simulating Tor networks, the enhancements reduced test time by about half. These enhancements are already part of the current version of Shadow.

Special thanks to the incredible USENIX staff, our panelists, program and steering committee, and the presenters for reviewing these summaries. The 12th CSET will again be co-located with USENIX Security 2019, with papers due in Spring 2019. Please consider submitting to or attending this unique and interesting workshop.