How to Design Computer Security Experiments

Sean Peisert[1] and Matt Bishop[2]

[1] Dept. of Computer Science & Engineering
University of California, San Diego
peisert@cs.ucsd.edu
[2] Department of Computer Science
University of California, Davis
bishop@cs.ucdavis.edu

Abstract. In this paper, we discuss the scientific method and how it can be applied to computer security experiments. We reiterate a number of general scientific principles, such as falsifiable hypotheses, scientific controls, reproducible results, and data quality.

1 Introduction

> "Computer security is also a science. Its theory is based on mathematical constructions, analyses, and proofs. Its systems are built in accordance with the accepted practices of engineering. It uses inductive and deductive reasoning to examine the security of systems from key axioms and to discover underlying principles. These scientific principles can then be applied to untraditional situations and new theories, policies, and mechanisms."[1]

Recently, there has been considerable controversy about the rigor of some experimental work and of the validity of some data sets used in computer security research. In some cases, the problem stems from poor experimental technique. In other cases, it comes from trying to apply the results, which are valid over the selected data sets, to environments or situations not reflected by the data sets.

The experimental method is critical to placing computer science on a firm scientific and engineering basis. In this paper, we reiterate a number of the fundamental tenets of the scientific method and discuss how they apply to computer security.

The scientific method—testing a hypothesis by performing controlled experiments, resulting in measurable, empirical data—must be used to evaluate anything that cannot be proven by pure mathematics or logical syllogisms [AriCE].

[1] *Computer Security: Art and Science* [Bis03], p. xxxiii.

2 The Scientific Method

Classically "science" was performed through pure *deduction*. In the 17th century, the experimental method augmented the syllogism; thus, *induction* joined deduction. This method usually follows the following process [Kuh62,Wik07]:

1. Form hypothesis
2. Perform experiment and collect data.
3. Analyze data.
4. Interpret data and draw conclusions.
5. Depending on conclusions, return to #1 and iterate.

This process requires the procedures to have several qualities. The ones most relevant to computer security are:

1. *Falsifiable.* An experiment must be constructed to test a hypothesis [New87] that is both testable and falsifiable [Pop59].
2. *Controlled.* An experiment must have exactly one variable, or if an experiment has multiple variables, then it must be able to be separated into multiple experiments where exactly one variable at a time can be tested [Lin53].
3. *Reproducible.* An experiment must be reproducible, and results repeatable [Boy61].

One other quality that we do not focus on arises when experiments require that the results of using an automated process be checked. An example would be an experiment to determine whether an intrusion detection system can identify certain attacks. These experiments should be *blind*, so the experimenter does not know whether the data being given to the intrusion detection system contains the attacks. This eliminates any bias introduced unconsciously by the experimenter. When human subjects are used, it is equally important that the experiment be *double blind*, so that neither the experimenter nor the subjects know which subjects are the controls and which are not.

In addition to using the results of an experiment to test a hypothesis, one must validate that the experiment does indeed test the hypothesis. One interesting aspect of experiments in computer science is that, in order to obtain data from the experiment to analyze, one must typically *supply* data to the computer or process. The output data, which is to be analyzed to validate or refute the hypothesis, depends upon some attributes of the input data—the contents of that data, or some attributes such as the statistical distribution of inter-arrival times of messages. Thus, the truth or falsity of every hypothesis is to some extent conditioned on the input data set. That data must be selected appropriately. For example, if the hypothesis says something about the way a system works on the Internet, the data set must reflect the characteristics of Internet traffic. Hence, an additional step in testing a hypothesis is the validation of the input data set. This step is often overlooked, and general conclusions about the validity of a hypothesis are drawn from data drawn from a local environment.

We expand on these ideas briefly.

2.1 Falsifiable

If a hypothesis is falsifiable, it can produce a negative result. Two factors in a hypothesis being falsifiable are *observability* and *measurability.* If something cannot be observed (either directly or indirectly), it cannot be measured or studied. For example, the hypothesis that prayer increases the number of times that one's deity speaks to oneself is not falsifiable, because humans have thus far not developed methods of observing personal divine communication, and thus no experiment could prove the reported results wrong. What is observed must also be measurable. For example, the hypothesis that some software is secure is not measurable because "security" *per se* is not measurable. Saying that the software does not acquire the privileges of user X without requiring user X's password *is* measurable, because one can execute the software with different inputs, and see if the property holds.

A worm can be released and monitored on an isolated network to determine its rate of propagation in the face of specific controls. But simply observing the behavior of that worm in the "wild" will not determine this rate because other factors may confound the analysis. Such an experiment is not testing; it is observation. Similarly, observing that the stock market cycles are correlated with the number of active Internet worms does not mean that a causal relationship exists, because neither can be controlled to be tested. However, a hypothesis about such a relationship, followed by a controlled experiment *could* be valid.

2.2 Controlled

Consider the hypothesis, "if we use intrusion detection system X, we will catch 50% more attacks than with our competitor's product, Y." This leaves a number of open questions. What is an attack? Do both products look for the same type of attacks? Are the additional attacks captured at the expense of additional false positives? A tool that produces zero false positives *or* zero false negatives is trivial to design: either reject or accept every instance, respectively. Are both products tested against the same dataset? If the intrusion detection systems are set up on a public access network, are they reading data from the same locations on the network? Are the IP addresses of their hosts similar? Are the operating systems and versions of their hosts identical? Answering "no" to any of these questions means that the experiment has more than one uncontrolled variable, and therefore, conclusions drawn about the effect of changing one variable are suspect. These questions cannot be enumerated without knowing the nature of the hypothesis and environment in which the experiments will take place, because experimental setup varies widely according to the aims of the research. But the questions must nevertheless be asked by the researcher when designing their experiments.

2.3 Reproducible

Consider the problem of obtaining data sets for analyzing the ability of intrusion detection systems to detect attacks accurately. One of the most common methods is to implement a *honeypot*, or a network of honeypots—known as a *honeynet* [Spi03] —and to wait to be attacked. Unfortunately, for this type of data set to be meaningful, one must show that attacks launched against honeypots are representative of attacks launched against production systems and networks.

Data sets are frequently not published, so the validity of the dataset itself cannot be verified. Among the reasons for this are that the data is proprietary, or contains sensitive information such as login names and passwords, or patient names and medical conditions, or bank account numbers and records of financial transactions. In some cases, data must be kept confidential by law. Even if this data were sanitized, one must show that the sanitization did not alter properties that affect the analysis.

These constraints lead to the development of artificial data sets that match the relevant statistics of real data sets. But what are the "relevant" statistics? An artificial data set may match first-order statistics, but not second-order statistics, of real data. So an experiment about a system that relies only on first-order statistics can use this artificial data set to produce meaningful results. The same cannot be said for a system that uses second-order statistics.

To summarize, enabling others to reproduce experiments requires that:

1. the conditions (assumptions, controls, and variables) of the experiment must be documented thoroughly enough so that the dataset can be re-created, to determine its validity, and
2. the data used must be saved, so that both the tools and methods in question, as well as new tools, can be tested against the existing data.

3 An Example Experiment in Computer Security

As an example, suppose we are testing a new firewall. The vendor claims that it is secure, and can protect any system behind it. However, this is not a question, but a statement. To clarify the claim, the vendor suggests the following hypothesis:

Hypothesis 1: Only an extraordinarily skilled attacker can break into our firewall.

Unfortunately, this hypothesis is not falsifiable, because if an attacker does breach the firewall, the vendor can claim the attacker is "extraordinarily skilled." So, to be falsifiable, we either need to quantify "extraordinarily skilled" or use some other metric.

Accordingly, we refine the question by returning to the original claim and consider what "secure" and "protect any system behind it" mean. After some discussion with the vendor, it becomes clear that the vendor is assuming that access to the controls of the firewall is from a physically connected terminal, so the only avenues of attack are through the traffic transiting the firewall.

Hypothesis 2: The firewall accepts all well-formed packets and sessions, and handles malformed packets and sessions as documented in the firewall's manual.

This hypothesis is both testable and falsifiable. We can bombard the firewall with packets, and begin a session and send incorrect packets as part of the session. If the firewall handles them properly (by discarding them, logging them, and possibly resetting the connection), then the hypothesis is true. If the firewall handles them improperly (by crashing, or by allowing them to corrupt its network stack), then the hypothesis is false.

We now consider an experiment to test the hypothesis. There are two variables involved in any such experiment: the firewall configuration, which dictates what packets should be allowed to transit the firewall, and the data set of incoming packets. We must therefore eliminate one variable.

If the goal of the experiment is to determine whether a particular configuration is the most effective for preventing some class of packets from transiting the firewall, then the variable is the firewall configuration. If the goal of the experiment is to determine whether the firewall correctly handles malformed packets and sessions, the variable is the data set of packets sent to the firewall. As we are interested in the former, we use the default firewall configuration supplied by the vendor and vary the data sent to the firewall.

Experiment 1: Connect the firewall to a local network and send packets, some malformed and some parts of malformed sessions, through the firewall. Record the packets and the firewall's responses.

This experiment is, unfortunately, not reproducible. Although the packets have been recorded, ancillary characteristics (such as timing) have not been. If the packets are arriving faster than the firewall can process them, the firewall may drop those packets or allow them to overwrite packets in the processing buffer, causing valid packets to appear malformed. So we must refine the experiment further:

Experiment 2: Connect the firewall to a local network and send packets, some malformed and some parts of malformed sessions, though the firewall. Record the network traffic, including timings, and the firewall's responses.

This experiment is reproducible, because anyone can take the network traces and reproduce the traffic. Unfortunately, its utility is questionable. To see why, we must consider the type of data set involved.

This particular data set is generated synthetically. But the firewall will not be used on synthetic data; it will be connected to real networks, with real traffic. How do we know that the data set accurately captures the relevant characteristics of the data that the firewall must handle in practice?

The relevance of the question lies in the nature of the firewall. If it is stateless, then each individual packet is handled independently of any other packet. The distribution of timings within the network traffic, though, may vary from site to site. A site whose main network activity is email will have a different distribution of inter-packet arrivals and departures than will a site that uses voice over IP, or transmits or receives videos on demand. The variance may impact the ability of the firewall to handle packets.

However, as the firewall is claimed to be able to detect malformed sessions, it is probably a stateful firewall; if not, how could it detect that packets, well-formed in and of themselves, were in a sequence that violated the state transitions of a well-formed session? This raises other questions, such as whether the distribution of packet content, length, and other attributes could affect the way the firewall handles packets. In this case, the data set needs to capture the statistics of *all* relevant attributes of the data. This strongly suggests using real data, not synthetic data.

Like the question of timings, the distribution of different types and sizes of packets and sessions varies among institutions. Hence our experiment cannot confirm the hypothesis in general; it can only confirm (or refute) it for the particular data set used. It is our responsibility to characterize that data set sufficiently to allow others to determine whether our answer is meaningful to them. As an example, if we used traffic from a site that only sent email over the Internet, and had no other network traffic, our results would not be particularly useful to a site where the network traffic were substantially different.

This, incidentally, is an often-overlooked point. One should not characterize network traffic in terms of the nature of the site from which it was obtained. Universities, for example, do not generate the same type of traffic. A college that focuses on teaching, and does not have a computer science major, might have predominantly web and email traffic. A college that has a computer science department actively researching new network protocols, and working with several commercial firms and the government, will generate a large amount of network traffic that is not easily characterized. So saying some network traces come from an academic site does not characterize it adequately enough to know *what* characteristics it has.

The best way to characterize our data set is to release it. In addition to enabling others to do exactly what we did, others can obtain their own data, determine if its characteristics match ours, and if so, then use it to reproduce the experiment. It also points out limits on our results: if the characteristics do not match, our results may or may not be valid. But providing results that can be shown to be invalid is good science: it allows claims to be refuted or refined, and thus progress made. Providing results that *can* be invalidated is ultimately necessary to develop results that ultimately withstand the test of time.

4 Conclusions

> DR. JONES: *"Archaeology is the search for* fact. *Not truth. If it's truth you're looking for, Dr. Tyree's philosophy class is right down the hall."*
>
> —*Indiana Jones and the Last Crusade* (1989)

The results of experiments can be misleading, especially when statistical analysis is involved. Darrell Huff's marvelous example about increases in California teachers' salaries causing increases in the profits of Nevada casinos underscores the difference between correlation and causation [Huf54]. Simply observing the effects of an experiment without first positing a hypothesis can give absurd results, such as an observation that "stock market cycles and sunspot cycles are roughly in sync, and the stock market peaks and dips slightly before sunspot cycles," so therefore, "stock market cycles cause sunspots." On the other hand, if a researcher could run an experiment in which she had control over either the sunspots or the stock market, then she could form the hypothesis that "stock market cycles cause sunspots" and conduct valid experiments to confirm or refute that hypothesis.

As an example from computer security, a worm can be released and monitored on an isolated network to determine its rate of propagation in the face of specific controls. But simply observing the behavior of that worm in the "wild" will not determine this rate because other factors may confound the analysis. Such an experiment is not testing; it is observation. Similarly, observing that the stock market cycles are correlated with the number of active Internet worms does not mean that a causal relationship exists, because neither can be controlled to be tested. However, a hypothesis about such a relationship, followed by a controlled experiment *could* be valid.

Sometimes there is insufficient data to form a useful or interesting hypothesis. In this case, experimentation can guide the researcher toward developing a hypothesis to test. This type of experiment is fundamentally different than the type discussed before, because it does not "prove" anything. As long as the nature of the experiment is clear, and the results are understood to be useful only in forming hypotheses that *other* experiments will test, this type of exploratory experimentation contributes to the advancement of the field. When the results of these exploratory experiments are used to validate hypotheses derived from their results, though, one reasons circularly—"we obtained the following results, which led us to hypothesize X, and we can confirm X because of the results of the experiment"—and so there is in reality no proof. An independent experiment is required to test the hypothesis.

In order to claim scientifically valid and justifiable results, computer security experiments must follow the scientific method, using high-quality, repeatable and verifiable methods and data sets. Only in this way, non-scientists in the "real world," such as those who make decisions about the security of electronic voting machines, hospital operating room equipment, and airplane software can trust the research and researchers whose technology they are using.

References

[AriCE] Aristotle. *Organon.* 100 B.C.E.

[Bis03] Matt Bishop. *Computer Security: Art and Science.* Addison-Wesley Professional, Boston, MA, 2003.

[Boy61] Robert Boyle. The Unsuccessful Experiment. In *Certain Physiological Essays.* Henry Herringman, London, 1661.

[Huf54] Darrell Huff. *How to Lie With Statistics.* Norton, 1954.

[Kuh62] Thomas S. Kuhn. *The Structure of Scientific Revolutions.* University of Chicago Press, Chicago, 1962.

[Lin53] James Lind. *A Treatise of the Scurvy.* Sands, Murray, and Cochran for A Kincaid and A Donaldson, 1753.

[New87] Sir Isaac Newton. *Philosophiae Naturalis Principia Mathematica.* The Royal Society, 1687.

[Pop59] Karl Raimund Popper. *The Logic of Scientific Discovery*. Routledge, 1959.

[Spi03] Lance Spitzner. The Honeynet Project: Trapping the Hackers. *IEEE Security & Privacy*, 1(2):15–23, Mar–Apr 2003.

[Wik07] The Free Encyclopedia Wikipedia. Scientific method. `http://en.wikipedia.org/w/index.php?title=Scientific_method&oldid=104300855`, January 30 09:59 UTC 2007.