In *The AAMAS workshop on Agent Design Advancing from Pratice to Theory* (ADAPT-09), Budapest, Hungary, May 2009.

Beyond Runtimes and Optimality: Challenges and Opportunities in Evaluating Deployed Security Systems

Matthew E. Taylor, Chris Kiekintveld, Craig Western, Milind Tambe
{taylorm, kiekintv, western, tambe}@usc.edu
http://teamcore.usc.edu/{taylorm,kiekintveld,western,tambe}
Computer Science Department, The University of Southern California

1. Introduction

As multi-agent research transitions into the real world, evaluation becomes an increasingly important challenge. One can run controlled and repeatable tests in a laboratory environment, but such tests may be difficult, or even impossible, once the system is deployed. Furthermore, traditional metrics used by computer scientists, such as runtime analysis, may be largely irrelevant.

This paper introduces a general framework for evaluating deployed systems with a case study in a security domain. Computer scientists can bring substantial expertise to bear on security problems, but we traditionally hold different priorities and standards for evaluating policies than the security community: computer scientists are used to quantitative evaluations in controlled studies, whereas security specialists are more accepting of qualitative metrics because their work is typically deployed. For instance, Lazaric [5] summarized a multi-year airport security initiative by the FAA where the highest ranked evaluation methodology (of seven) relied on averaging *qualitative* expert evaluations. Quantitative evaluations in the real world are difficult for a number of reasons: scientific tests may be prohibitively difficult (or impossible), data may be sensitive and unavailable, and it may be inherently difficult to quantify metrics (i.e., public perception).

The primary contribution of this paper is to provide a preliminary framework to evaluate a deployed system by determining what to measure, how to measure it, and how such metrics determine the system's utility. A secondary contribution of this paper is to help familiarize the agents community with a selection of difficulties inherent in evaluating deployed applications.

2. Case Study: ARMOR

ARMOR (*Assistant for Randomized Monitoring Over Routes*) is a software tool designed to assist police officers at the Los Angeles International Airport (LAX) with scheduling deployments for canine units and vehicle checkpoints [7]. LAX is a large physical environment, and resources are not available to cover the entire area at all times. A key principle in ARMOR is to use randomization so that attackers cannot predict where security resources will be deployed ahead of time, increasing the effectiveness of the resources. The randomization accounts for three key factors: (1) attackers are able to observe the security policy using surveillance, (2) attackers change their behavior in response to the security policy, and (3) the risk/consequence of an attack varies depending on the target. ARMOR accounts for these factors by modeling the problem as a Bayesian Stackelberg game, using the power of game theoretic analysis to predict attack behaviors and to optimize the security policy accordingly. The end result is a randomized police schedule that is unpredictable, but weighted towards high-valued targets. ARMOR has been in use at LAX since August 2007, marking an important transition from theoretical to practical application. The system has received very positive feedback and is considered an important element of security at the airport.

2.1 Current ARMOR Evaluation

The ARMOR system has undergone multiple evaluations before and after deployment. From a game theoretic standpoint, the system has been compared to a uniform random schedule in a variety of settings and shown to be able to handle multiple adversary behavior types. The system's

runtime was verified as sufficiently fast for the application. Recent studies [6] demonstrate that the system's expected performance is similar to measured performance with human adversaries. Qualitative internal and external security reviews indicate that ARMOR is both effective and highly visible. Director James Butts, LAX Police, reported that ARMOR "makes travelers safer," and Erroll Southers, Assistant Chief of LAX Airport Police, told a Congressional hearing that "LAX is safer today than it was eighteen months ago," due in part to ARMOR. Finally, although ARMOR was designed as a mixed initiative system, users choose not to modify ARMOR policies in practice, suggesting that output policies are indeed high-quality.

ARMOR is relatively inexpensive to implement and saves police significant time as they no longer have to hand-design patrol schedules (which were sometimes provably exploitable). For these two reasons alone, it is not difficult to show that ARMOR is an improvement over previous best practices. However, the question this paper begins to address is: how can evaluations conducted by different groups, with different end goals, be combined to asses ARMOR's true utility?

2.2 Related Domains

Many security domains share characteristics that make them difficult to evaluate [1, 4, 5]. For example, airports may employ plainclothes security, an example of a non-visible measure that may help detect attacks but not with attack deterrence. Additionally, police forces face annual budget challenges in which they attempt to justify how dollars spent translate into community benefit and why one program should be funded over another. For instance, it is difficult to show that an increase in anti-drug spending reduced drug sales – although dealer arrests could increase, if the number of active criminals also increases, the police may have caught a smaller percentage overall.

3. Dimensions of Comparison

As discussed in Section 1, there is currently no gold standard for evaluating security applications. In the ARMOR system, the ultimate goal is to maximize the benefit per cost (i.e., *utility*), a quantity not directly measurable. In the following sections we introduce a framework that organizes possible tests by type and quantity measured. Every test makes different assumptions and attempts to measure different quantities; it is important to clearly define such expectations. We first discuss three general dimensions of evaluation (Section 3.1) and then discuss a fourth security-specific dimension (Section 3.2). Note that the proposed categories and metrics are intended to be representative, not exhaustive.

3.1 Types of Tests

TEST CATEGORIES

When determining what evaluation(s) to conduct, an important consideration is whether test assumptions are met in the real world. While a mathematical analysis may be relatively easy to compute and provide theoretical guarantees, it makes many assumptions that may be incorrect [1, 5]. At the other end of the spectrum, situated tests using physical personnel are much more realistic, but are quite expensive and may not be able to directly measure desired quantities. We group tests into the following categories:

- **Mathematical** Formal reasoning is used to determine the goodness of a method
- **Simulation** Abstract or realistic computational simulations of the situated method are tested repeatedly to determine an expected goodness
- **Controlled human studies** Humans in abstract or realistic studies can account for human decision making, which is not always optimal or rational

• **Situated studies** Observing the behavior of the system in the real world (uncontrolled) or testing the deployed system (controlled) leverages true-to-life situations

• **Qualitative expert studies** Domain experts can examine a system and give a holistic evaluation

TEST REPRODUCIBILITY

Different evaluation methodologies will have different expected precision (i.e., the test can be repeated and achieve similar results). A game theoretic model can produce repeatable results: others can run tests with the same assumptions and find the same expected value. On the other hand, a one-shot real world test may be very realistic, but it would be difficult to extract any conclusive results from a single trial.[1]

TEST COST

The cost of evaluation must be taken into account. Studies that require monetary investment, reduce productivity, use domain experts, or involve risk, are less preferable than tests that do not.

3.2 Quantitative Metrics

The previous section discussed general test categories and this section details security-specific metrics due to our domain focus. The goal of a security system is to maximize utility: attack damage, attack frequency, and cost should be minimized. These three *primary* metrics are not directly measurable in all types of tests, but *secondary* metrics often are, each of which is correlated with one or more primary metrics (and therefore utility).

• **# Attacks Prevented** How many attackers were successfully caught? *Pro:* Provides a measurable number of attacks successfully thwarted, an indication of the system's benefit. *Con:* Such a number is not very useful unless the total number of attempted attacks is also known.

• **# Attacks Deterred** How many people considered attacking, but did not, because of security? *Pro:* Attack deterrence may be a primary benefit of security[1, 4]. *Con:* Deterrence is generally impossible to measure directly.

• **Planning Time Required** Do the attackers need considerable planning time? *Pro:* Longer planning time increases deterrence and provides opportunities to detect the attackers' surveillance. *Con:* Sufficiently motivated attackers have spent significant time on reconnaissance in the past.[2]

• **Attacking Resources Required** Can a single attacker with simple equipment cause significant damage? Or is sophisticated equipment and/or multiple attackers required? *Pro:* Like planning time, increased resources require larger attacker efforts, increasing the chance of detection or infiltration. *Con:* Attackers may have sufficient resources, regardless.

• **Cost Estimate (cost)** What are the expected implementation and maintenance costs for a particular measure (including detrimental effects such as inconvenience to passengers, lower cargo throughput, etc.)? How does this cost change when resources are added/removed to/from the security measure? *Pro:* Such a measurement can help decide which security measurements to implement. *Con:* All effects, positive and negative, must be quantified.

• **Expected Attacker Damage** What is the expected benefit to an attacker in a game theoretic sense? If a payoff matrix is accurately estimated, a rational attacker with an average negative payoff should choose not to attack. *Pro:* Game theoretic reasoning is relatively simple to compute and provide guarantees. *Con:* Multiple assumptions must hold about the attackers' behavior and preferences for the reasoning to be correct [3].

1. Many trials would be needed to accurately gauge the likely success. However, multiple trials would likely invalidate the results, particularly in a security domain with human defenders.
2. For instance: `http://www.globalsecurity.org/security/profiles/dhiren_barot.htm`

4. Evaluation Options

Having discussed types of evaluations and a set of possible metrics, this section presents a representative list of evaluation methods for security domains. When deciding how to evaluate an application, the tradeoffs of each test and associated measurement must be carefully weighted — this list suggests how a specific test should be evaluated in terms of type and metric in order to help decide if it should be conducted as part of a comprehensive evaluation.

1 – Game Theoretic Analysis: Given assumptions about the attacker (e.g., the payoff matrix is known), game theoretic tools can be used to determine the attacker's expected payoff. Additionally, deterrence can be measured by including a "stay home" action, returning neutral reward.[3]

2 – Attacker Resources vs. Damage: A game theoretic analysis can evaluate how attacker observation, equipment, and attack vectors can change the expected attacker payoff.

3 – Defense Dollars vs. Successful Attack: A game theoretic analysis can measure how attacker success varies as security measures are added (e.g., implementing a new baggage screening process), or increasing the strength of an existing measure (e.g., adding checkpoints).

4 – Simulated Attacks: A simulator with more or less detail can be constructed to model a specific security scenario. Such modeling may be more realistic than a game theoretic analysis because structure layout, simulated guard capabilities, and agent-level policies[4] may be incorporated.

5 – Human Studies: Human psychological studies can help to better simulate attackers in the real world. Evaluations on an abstract version of the game may test base assumptions, or a detailed rendition of the target in a virtual reality setting with physiological stress factors could test psychological stress. Human tests suffer from the fact that participants are not (one would hope) drawn from the same population as the actual attackers.

6 – # Foiled Attacks: The number of attacks disrupted by a security system can provide a sanity check (i.e., it disrupts a non-zero number of attacks). If the metric is correlated with an estimated number of attacks, it may help estimate of the attacker percentage captured. Enabling and disabling the security system and observing how the number of foiled attacks changes would be more accurate, but this methodology is likely unethical in a real world setting.

7 – Red Team: Tests in which a "Red Team" of qualified security personnel attempt to probe security defenses provide realistic information done in life-like situations using the true defenses (including those that are non-visible). However, such a test is very difficult to conduct as some security must be alerted (so that the red team is not endangered) while remaining realistic, the tests are often not repeatable, and a single test is likely unrepresentative.

8 – Expert Evaluation: Security experts — internal or external — may holistically evaluate a target's defenses, including both visible and non-visible and provide a high-level security assessment.

9 – Cost Study: A cost estimate for a security measure with different levels of staffing (or other resource) may help determine expected utility, but some intangible factors may be very difficult to determine, such as quantifying a decrease in civil liberties.

Current ARMOR evaluations (see Section 2.1) use a subset of the tests above. However, as is evident from Table 1, our evaluation could (and should) be improved. For instance, none of our current evaluations consider the amount of attacker resources required (deterrence and attacker cost), how defensive effectiveness changes with different numbers of units (expected attacker damage and cost), or use any data directly from the deployed system — with the possible exception of qualitative expert evaluations. By enumerating the evaluation possibilities and their expected

3. Some attackers may be set on attacking at any cost and must be modeled without such an action.
4. One exciting direction, as yet unexplored, is to incorporate machine learning into such policies. Such an extension would allow attackers to potentially discover flaws in the system, in addition to modeling known attacker behaviors.

Evaluation Summary

Test	Type	Reproducibility	Cost	Prevented	Deterred	Plan Time	Resources	Cost	Damage
Game Theory	Mathematic	High	Low	×	×				×
Attacker Resources / Payoff	Mathematic	High	Low	×		×	×		×
Defense Dollars / Damage	Mathematic	High	Low	×				×	×
Simulated Attacks	Simulation	High	Low	×					×
Human Studies	Human	Med	Med	×					×
# Foiled	Situated	Low	Low	×					
Red Team	Situated	Low	High	×					×
Expert Evaluation	Qualitative	Low	Med	×		×	×		×
Cost Study	Math. / Qual.	Med	Low					×	

Table 1: This table summarizes our proposed evaluation methods by suggesting where each falls along the three general dimensions and which of the six security-specific metrics are measured.

benefits, we can better determine what evaluations are necessary to show that a deployed system not only functions well, but also to approximate its true utility.

5. Conclusion

Analysis of policies in non-security systems is problematic [2], and security domains have additional difficulties. In truth, it is often hard to evaluate complex deployed systems in general — in our field a test of the prototype often suffices (c.f., Scerri et al. [8]). However, difficulties in evaluation does not excuse a lack of repeatable and detailed evaluations.

While none of the evaluation tests presented in Section 4 can calculate a measure's utility with absolute accuracy, understanding what each test *can* provide will help evaluators better understand what tests *should* be run on deployed systems. The goal of such tests will always be to provide better understanding to the "customer," be it researchers, users, or policy makers. By running multiple types of tests, utility (the primary quantity) can be approximated with increasing reliability.

In the future we plan to use this framework to help decide which evaluation tests are most important to determine ARMOR's utility. Additionally, we intend to continue collaborating with security experts to determine if our framework is sufficiently general to cover all existing types of security tests, as well test how the framework can guide evaluation in additional complex domains.

References

[1] V. M. Bier. Choosing what to protect. *Risk Analysis*, 27(3):607–620, 2007.

[2] R. Blundell and M. Costa-Dias. Alternative approaches to evaluation in empirical microeconomics. *Journal of Human Resources*, Forthcoming, 2009.

[3] I. Erev, A. E. Roth, R. L. Slonim, and G. Barron. Predictive value and usefulness of game theoretic models. *International Journal of Forecasting*, 18(3):359–368, 2002.

[4] S. H. Jacobson, T. Karnai, and J. E. Kobza. Assessing the impact of deterrence on aviation checked baggage screening strategies. *International J. of Risk Assessment and Management*, 5(1):1–15, 2005.

[5] R. Lazarick. Airport vulnerability assessment – a methodology evaluation. In *Proc. of 33rd IEEE International Carnahan Conference on Security Technology*, 1999.

[6] J. Pita, M. Jain, M. Tambe, F. Ordonez, S. Kraus, and R. Magori-Cohen. Effective solutions for real-world Stackelberg games: When agents must deal with human uncertainties. In *Proc. of AAMAS*, 2009.

[7] J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Paruchuri. Deployed ARMOR protection: The application of a game theoretic model for security at the Los Angeles international airport. In *Proc. of AAMAS*, 2008.

[8] P. Scerri, T. Von Goten, J. Fudge, S. Owens, and K. Sycara. Transitioning multiagent technology to UAV applications. In *Proc. of AAMAS Industry Track*, 2008.