Mixed-Initiative Security Agents

Rachel Greenstadt and Sadia Afroz and Michael Brennan
Drexel University
{greenie,sa499,mb553}@cs.drexel.edu

June 29, 2009

Abstract

Security decision-making is hard for both humans and machines. This is because security decisions are context-dependent, require highly dynamic, specialized knowledge, and require complex risk analysis. Multiple user studies show that humans have difficulty making these decisions, due to insufficient information and bounded rationality. However, current automated solutions are often too rigid to adequately address the problem and leave their users more confused and inept when they fail. A mixed-initiative approach, in which users and machines collaborate to make security decisions and make use of complementary strengths rather than weaknesses, is needed.

1 Introduction

Techniques from artificial intelligence (notably bayesian learning and captchas) have achieved great success in helping adminstrators manage automated attacks such as SPAM and network attacks that would overwhelm human capacities [18, 15]. This paper argues, however, that artificial intelligence techniques have an even greater role to play in the security story.

Security decision-making is hard for both humans and machines. This is because security decisions are context-dependent, require specialized knowledge that is highly dynamic (due to sophisticated adversaries and evolving threats), and require complex risk analysis. Multiple user studies show that humans have difficulty making these decisions, due to insufficient information and bounded rationality [1, 17, 7, 19, 5, 16]. As a result, users generally approach this problem in an all-or-nothing way, either retreating from online activities or throwing caution to the wind. The result of this is lost productivity either by missed opportunities when users retreat from a security decision (or are curtailed by administrators) or through the expenses incurred by cleaning up after infections and/or when end-users unwittingly support the computer crime infrastructure with their computational or financial resources.

Current automated solutions are often too rigid to adequately address this problem, and leave their users more confused and inept when they fail [6]. Automation annoys users when it prevents them from getting their primary work done [1].

A **Mixed-initiative approach** refers broadly to methods that explicitly support an efficient, natural interleaving of contributions by users and automated services aimed at converging on solutions to problems [11]. The term comes out of the user interface community and reflects a rejection of the choice between interfaces that allow users to directly manipulate objects and interfaces that sense user activity and take automated actions. We argue that a mixed-initiative approach to

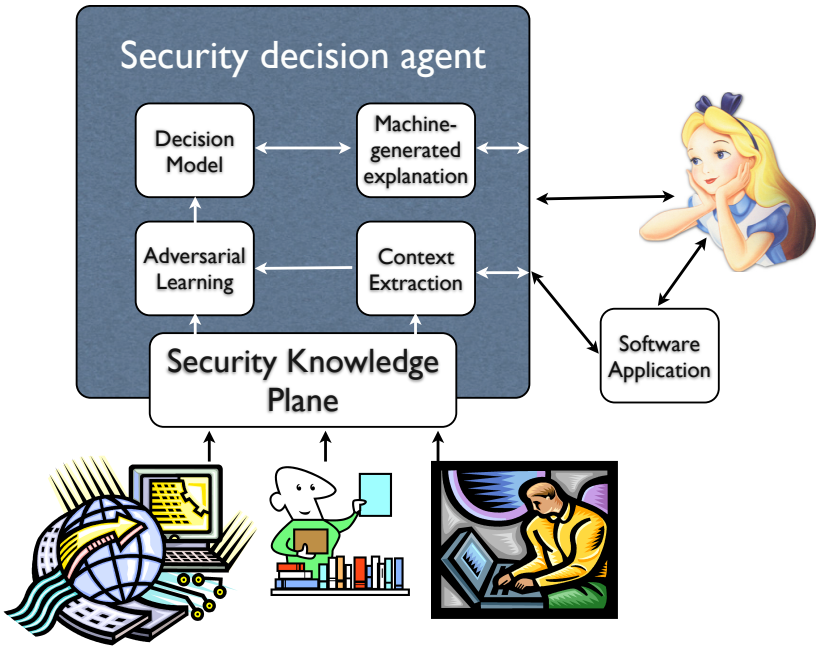

Figure 1: Proposed design of a security decision agent. The agent interacts directly with user Alice and her software application and can also draw on knowledge-bases on the Internet, contributed by experts and other users.

security decision-making is needed, in which users and machines will collaborate to make security decisions, making use of complementary strengths rather than weaknesses. Such an approach will require shared representations of contextual information, well-designed interfaces, adversarially-resistant learning mechanisms, and trustworthy methods for incorporating global information from outside sources.

This paper extends the ideas of adjustably autonomous security agents [8]. A potential design for such an agent is shown in Figure 1. A discussion of how virtualization can be used to protect security agents from compromise can be found in [8].

2 Design of Security Assistant Agents

Recent research has shown that humans have difficulty in consistently performing repetitive tasks, memorizing large amounts of information, and managing the accumulation of small risks or understanding long-term risk [17]. However, machines lack common sense reasoning and often fall behind humans in strategic planning tasks. They can be fully compromised by control hijacking attacks. Finally, there will always be cases when machine automation fails and humans must be relied upon. Currently, computer security often falls in a worst of all worlds in which attackers are able to exploit both machine and human weaknesses.

Decision Models. However, we can and should deconstruct security decision-making to understand which components of the decision are best made by humans and which are best made by

machines. Once this is better understood, we can build security assistant agents, where humans and machines can complement each others' expertise when making these decisions.

Context extraction. When programs ask their human users to make decisions about security, the reason that the decision cannot be fully automated is often due to the lack of contextual information that the program has about the decision. Security decisions often rely on several contextual factors for input including (1) the software and network activities going on concurrently with the decisions, (2) knowledge about the resource being accessed (the webmasters, code authors, or certificate authorities), and (3) the beliefs, desires, and intentions of the human users interacting with the system. When security assistant agents can gather more of this contextual information, we hypothesize that they can much better assist their users in making security decisions.

Machine generated explanations of security risk factors/decisions. If humans and machines are to collaborate in decision-making processes, then the reasoning behind computer input to these decisions must not be opaque to humans. Certain machine learning representations, such as decision trees, are more understandable to humans than other methods, such as SVNs or neural networks. When security agents make recommendations, we need to find ways to translate the reasoning behind these decisions to something humans (and particularly humans without strong technical or security backgrounds) can understand and assimilate into their view of the world. Good work along these lines has been done in manipulating nuanced privacy preferences [14].

Further challenges include adapting AI techniques to adversarial situations such as **adversarial learning** and building a **knowledge plane** that incorporates potentially unreliable or adversarial information from other humans and/or software agents.

The ultimate goal is an agent system like that illustrated in Figure 1. While this may sound daunting we show in the following subsections that minimal amounts of context extraction and understanding of adversarial learning capabilities can make a profound impact on security decisions such as "Should I trust this website" or "Is this piece of writing likely to reveal my identity?"

2.1 Shared representations and phishing detection

Phishing is a web-based attack that uses social engineering techniques to exploit Internet users and acquire sensitive data. Most phishing attacks work by creating a fake version of the real site's web interface to gain the user's trust. Phishing is a web-based attack that uses social engineering techniques to exploit Internet users and acquire sensitive data. Most phishing attacks work by creating a fake version of the real site's web interface to gain the user's trust. Despite the fact that these phishing sites look identical or nearly identical to the real sites they imitate, user studies have shown that users ignore browser-based indicators and often use the appearance of a site to judge the authenticity of sites, just as they use the appearance of physical sites to judge their authenticity. According to the Anti-Phishing Working Group (APWG), there are at least $47,324$ phishing attacks and a top-ten American bank estimates that at least US$300 is lost for every hour that a phishing site remains up [9].

We argue that effective phishing detection mechanisms must detect phishing sites from the user's point of view. That is, the detection should be directly related to the look and feel of the site.

The majority of users provide sensitive credentials to a small set of sites (fewer than 20). Under the assumption that SSL is supported by these sites of interest and secure in both the underlying protocol and the trust model used by the browser, these sites can be whitelisted and browsers can automatically verify their authenticity. The problem with whitelisting approaches [10, 4, 12],

is that the user must know about and remember to check the interface every time they visit the site, and there is ample evidence that this is beyond most users' capacities. However, warning users when the site they are visiting is not among their sensitive subset is also futile, as the vast majority of sites visited by users are not sensitive and such warnings will be quickly tuned out or turned off. What is needed is for the browser to infer the user's *false belief* that she is visiting one of her sensitive sites and only warn (actively and emphatically) in this case. Our hypothesis is that similar-looking content can be detected by automated methods. Preliminary results suggest that this is the case, and that current attacks (as measured by mining `www.phishtank.com` can be detected with 97% accuracy and high precision (less than 1% false positives) [2].

The goal of the attacker will be to make websites that are different to computer algorithms, but (close to) identical to human eyes. If the attackers still prove successful in defeating our matching algorithms, they will have contributed to our understanding of the vision problem in much the same way that spammers have improved statistical machine learning and bots that defeat captchas have improved optical character recognition algorithms. If these techniques succeed, but reduce the efficiency of our techniques, they can be run offline (on email links) or by intermediaries.

We argue the reason that this approach works well is that it creates a shared representation between the user and the browser in the form of a content profile for each trusted site. This allows the browser to recognize imitations and warn the user.

2.2 Adversarial learning and anonymous publishing

The web is full of anonymous communication that was never meant to be analyzed by authorship recognition systems. An anonymous message board, for example, is often not meant to reveal which posts are by which authors, or how many authors exist on the forum in the first place. While posters can hide their IP addresses using anonymous communication protocols such as Tor, the linguistic content of their posts might still give them away.

For this reason, it is important to understand the degree to which machine learning-based authorship recognition techniques are effective. We have found that three authorship recognition techniques, which are representative of current trends in the field, fail when ordinary users try to hide their writing style [3], even though are quite effective when users do not modify their writing. Other recent research has shown that text analysis can be used to find and modify the most salient features in a document in order to protect the anonymity of the author [13].

The results of these studies demonstrate that (a) it is possible to retain privacy against current stylometric techniques, (b) the high effectiveness of authorship techniques on unmodified documents suggests it important for users who desire privacy to take measures to hide their identity, and (c) that automation can be used in order to detect the best means for obfuscating a document.

While it may be easy for an algorithm to modify a document in order to preserve anonymity, it is a much more complicated task to do so in such a way that preserves the semantic content of the text. And while a human can better modify a document without obscuring it's meaning, it is unreasonable to expect a person to perform complex analysis on the text to analyze the most vulnerable features. This paves a clear path for building an agent that assists users in determining when it would be worthwhile to obfuscate their writing and the most effective ways of doing so.

3 Conclusion

This position paper argues for the development of a mixed-initiative approach to security, in which users and machines collaborate to make security decisions and make use of complementary strengths rather than weaknesses.

If a scientific foundation for mixed-initiative security agents can be successfully developed, then integrated into browsers, operating systems, and applications, it will make the work of the attacker much harder. Currently, there is a choice for people between participation in Internet life and risk. Those who are less educated and computer savvy face larger risks and often preyed upon by identity thieves, scammers, and other attackers. They are used to build the infrastructure (botnets) to attack more hardened targets. Improving security decision-making at the end user level can have a broad impact on overall computer security.

Examining these ideas will help us to understand the relationship between the fields of HCI, AI, and security to the benefit of all three. Security decisions provide a good domain for studying collaborative human/agent decision-making as they provide complexity, but also concrete right and wrong answers that can help illuminate how humans and machines should collaborate in other situations.

References

[1] Anne Adams and Martina Angela Sasse. Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures. *Communications of the ACM*, 42(12):40–46, December 1999.

[2] Sadia Afroz and Rachel Greenstadt. Phishzoo: An automated web phishing detection approach based on profiling and fuzzy matching. Technical Report DU-CS-09-03, Drexel University, 2009.

[3] Michael Brennan and Rachel Greenstadt. Practical attacks on authorship recognition techniques. In *Innovative Applications of Artificial Intelligence*, 2009.

[4] Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell. Client-side defense against web-based identity theft. In *11th Annual Network and Distributed System Security Symposium (NDSS '04)*, February 2004.

[5] Rachna Dhamija, J. D. Tygar, and Marti Hearst. Why phishing works. In *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2006.

[6] W. Keith Edwards, Erika Shehan, and Jennifer Stoll. Security automation considered harmful? In *New Security Paradigms Workshop (NSPW)*, 2007.

[7] S. Egelman, L. Cranor, and J. Hong. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. In *CHI*, 2008.

[8] Rachel Greenstadt and Jacob Beal. Cognitive security for personal devices. In *First ACM Workshop on AISec (AISec'08), ACM CCS 2008 Conference*, 2008.

[9] Anti-Phishing Working Group. Global phishing survey: Domain name use and trends in 1h2008. http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey1H2008.pdf, 2008.

[10] A. Herzberg and A. Gbara. Trustbar: Protecting (even naive) web users from spoofing and phishing attacks. *Cryptology ePrint Archive*, (155), 2004.

[11] Eric Horvitz. Principles of mixed-initiative user interfaces. In *CHI '99: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 159–166, 1999.

[12] Waterken Inc. Waterken yurl trust management for humans. http://www.waterken.com/dev/YURL/Name/.

[13] Gary Kacmarcik and Michael Gamon. Obfuscating document stylometry to preserve author anonymity. In *COLING/ACL on Main conference poster sessions*, July 2006.

[14] P.G. Kelley, P. Hankes Drielsma, N. Sadeh, and L.F. Cranor. User-controllable learning of security and privacy policies. In *First ACM Workshop on AISec (AISec'08), ACM CCS 2008 Conference*, 2008.

[15] E Michelakis, I Androutsopoulous, G Paliuras, and G Sakkis. Filtron: A learning-based anti-spam filter. In *1st Conference on Email and Anti-Spam*, 2004.

[16] Stuart Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. The emperor's new security indicators. In *IEEE Symposium on Security and Privacy*, May 2007.

[17] Bruce Schneier. The psychology of security. http://www.schneier.com/essay-155.html, 2008.

[18] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. In *Eurocrypt*, 2003.

[19] Alma Whitten and J.D. Tygar. Why johnny can't encrypt: A usability evaluation of pgp 5.0. In *Usenix Security Symposium*, 1999.