Cognitive Security for Personal Devices

Rachel Greenstadt
Drexel University
greenie@eecs.harvard.edu

Jacob Beal
MIT
jakebeal@mit.edu

ABSTRACT

Humans should be able to think of computers as extensions of their body, as craftsmen do with their tools. Current security models, however, are too unlike those used in human minds—for example, computers authenticate users by challenging them to repeat a secret rather than by continually observing the many subtle cues offered by their appearance and behavior. We propose two lines of research that can be combined to produce *cognitive security* on computers and other personal devices: continuously deployed multi-modal biometrics and adjustably autonomous security.

Categories and Subject Descriptors

I.2 [**Computing Methodologies**]: Artificial Intelligence; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Security, Human Factors

Keywords

Artificial Intelligence, Security Automation, Authentication, Biometrics

1. MOTIVATION AND OVERVIEW

We would like to treat our personal devices as extensions of ourselves, but a single bad interaction (with a website or piece of code) can thoroughly compromise a machine, giving control over its data, resources, and operation to an adversary. The security community has largely responded to this state of affairs by erecting barriers between the user and the device—more passwords, icons, dialog boxes, and warnings to increase the user's vigilance. This vigilance comes at the expense of convenience and productivity and yet is still brittle.

We propose that computers should act more like humans in their security decisions and characteristics. Human minds have evolved *cognitive security*: rich and subtle mechanisms for handling trust and security in social interactions. For example, when presented with authentication documents, human verifiers are trained to examine the humans as well as the documents themselves. The personal interview, harnessing all our subtle cognitive security traits, is still the gold standard for determining malicious intentions in humans [17]. Building intelligent agents with the ability to reason about security could help to bridge the gap between a personal device and its human user.

We propose advancing two lines of research that can be combined to achieve cognitive security for personal devices. First, imprinting and continuously deployed multi-modal biometrics, allowing a device to reliably recognize its owner. To support this, we advocate the use of virtualization and trusted computing, which allow the security device to protect itself while running untrusted applications. The second line of research is adjustably autonomous security, allowing a device to make security decisions with user input for important judgement calls. Progress in these areas will help produce devices that users can safely treat as extensions of themselves.

2. DIFFERENTIATING BETWEEN USERS

To serve its owner well, a device must be able to reliably differentiate between its owner and other users. Conventional methods for doing this are brittle, however, so we argue for two fundamental changes in how a device determines whether its user is its owner. First, rather than using a challenge/response protocol, we argue that a device should recognize its user through ongoing measurements of many streams of readily available biometrics. Second, we argue that no user should ever be allowed to set the identity of the owner; instead, a device should imprint on the first user it has a sufficiently rich interaction with, much like a baby bird imprints on the first sufficiently mother-like object it encounters. These two changes should allow a device to establish a durable privileged relationship with its owner.

2.1 Recognizing the Owner

How can Abacus, a device, tell that it is interacting with Alice, its owner? Conventional security designs generally use a challenge/response strategy. Under this approach, when Alice begins a session with Abacus, it challenges her to prove her identity, often using secret information such as a password. Once she has passed this challenge, however, Abacus never challenges her again that session, except perhaps after long periods of idleness. Although the challenge may be

AISec'08, October 27, 2008, Alexandria, Virginia, USA.

arbitrarily complex and difficult to fake—a password, biometrics, hardware keys, etc—the pragmatics of human usage tend to erode this type of security. A challenge/response between a human and a device is either weak (e.g. an easily memorized password or PIN) or long, difficult, and annoying (e.g. the combination of a hardware key, a pass-phrase, and a biometric).

Humans do not normally recognize one another this way: in most everyday interactions, we recognize people by who they are and how they behave, rather than by the secrets that they know. The cues we use for recognition range from immediate and obvious, such as facial structure, voice, and gait, to subtle and slowly emerging, such as fidgeting behavior and preferred topics of conversation. Nearly all, however, are based on streams of public information that are made available naturally throughout an interaction. Moreover, most users already view these sorts of biometrics as both acceptable and trustworthy[9], particularly for a personal device where privacy is not at stake.

There is already a great deal of research in biometric fusion—the combining of different biometrics in a single challenge response approach [14, 2, 3]—and in using a single continuously deployed behavioral biometric (notably keyboard input[12, 10]) to authenticate a user. What we advocate is combining the two into a continuously deployed multi-modal approach, in which many different low-fidelity streams of biometric information are combined to produce an ongoing positive recognition of a user. With an optimistic and ongoing recognition process, security can be stronger, because attackers must impersonate Alice well enough to satisfy many different cues throughout their interaction with Abacus, and also less intrusive, because Alice can gain privileges merely by interacting with Abacus.

There are potentially many cues available to help Abacus recognize Alice: typing patterns (speed, pause patterns, pressure, frequent mistakes), touchpad/mouse patterns (smoothness of arc, idle/wander patterns, duration and frequency of clicks, double-click interval), camera images (invariant facial structure, height, body shape, tics and motion patterns), posture/device placement detected by accelerometer, high-level usage (favorite web destinations, word/phrase choices, idleness patterns), voice patterns (tone, inflection, speech patterns), and many more. Alone, none of these cues is likely to approach the consistency or reliability of a strong password. As an aggregate, however, they may out-perform challenge/response approaches.

Although many devices will have only a subset of these sensors, there are enough cues available that any large subset ought to provide enough information for a good recognition signal. For example, with 20 cues, each with an independent 20% error rate, a 2/3 vote has only a 1 in 500,000 chance of producing an incorrect decision.

The machine-learning subfield of ensemble learning[13] is explicitly focused on the problem of building strong classifiers from many weak classifiers, and has produced algorithms such as AdaBoost[5] that should be easily applicable to the problem of combining cues to provide good security while never denying Alice access to her own device. Existing work in multi-modal biometrics (e.g. [14], [2]) has already shown that machine learning can be used to fuse data sources and boost the efficacy of person recognition. In a continuously deployed multi-modal approach, we simply greatly multiply the number of sources and the time-frame over which they are considered.

Because cues are based on readily available information, the goal of a cue recognition system cannot be perfect security. It will always be possible for a sufficiently determined attacker to study Alice thoroughly enough and invest enough time and effort in counterfeiting to fool Abacus. Rather, the goal of cue recognition is to make the cost of doing so high enough that an attacker will almost always prefer a different approach: even if the attacker knows how Abacus recognizes Alice, sustaining an appropriate ensemble of cues is likely to be difficult.

In some cases, however, the interaction may not be rich enough to identify Alice with high confidence. It is an open question whether to try to enhance with challenge and response and/or dial down Alice's privileges in these circumstances.

2.2 Imprinting Abacus on Alice

Currently, devices are designed so that a sufficiently privileged user can easily change anything on the machine, including how it recognizes Alice. The only way to avoid this is to have Abacus' recognition of Alice be immutable.

There is a metaphorical similarity here to the psychological notion of *imprinting*. Many animals go through critical periods where they rapidly learn to recognize a special stimulus. Imprinted relationships are extremely durable, often lasting a lifetime. Geese, for example, imprint on the first suitable moving object they see shortly after hatching, and will ever after treat it as their parent[11]. This model has been extended to computational devices in the Resurrecting Duckling security model [18].

When Abacus has its first long and cue-rich encounter with Alice, we argue that it should imprint on her. During that first encounter, it should learn to recognize Alice in as many different ways as possible, fixing her characteristics in a protected memory where not even Alice herself can modify them. Alice can thus trust that even a successful attack on Abacus will not compromise its relationship with her.

If Alice wishes to loan out her device, she can still do so. Abacus can allow delegated users to interact with it at a lower level of privilege based on its ability to recognize them and the trust delegated to them by Alice. Transferring ownership is more complex and requires a "death and rebirth" wherein Abacus reinitializes itself completely, destroying any data that Alice has on it, then imprinting freshly on its new owner.

2.3 An Architecture for Machine Integrity

Building the kind of trustworthy agent devices we envision requires some architectural support. Developments in trusted computing [7], virtualization, and instrumentation provide a strong basis for the development of more sophisticated security models. While there may be other ways to provide the needed security, this combination is already an active area of research (e.g. sHype, Xense, Terra, etc) [1, 6].

Such an architecture is illustrated in Figure 1. The trusted computing base—the software which must be vulnerability-free to protect the system—is shown in yellow. The core of the system is protected by a trusted platform module (TPM) that protects key material and ensures the integrity of the virtual machine monitor (VMM) above it. The security manager, verified by the VMM below it, runs in a virtual machine that manages the rest of the VMs. Abacus

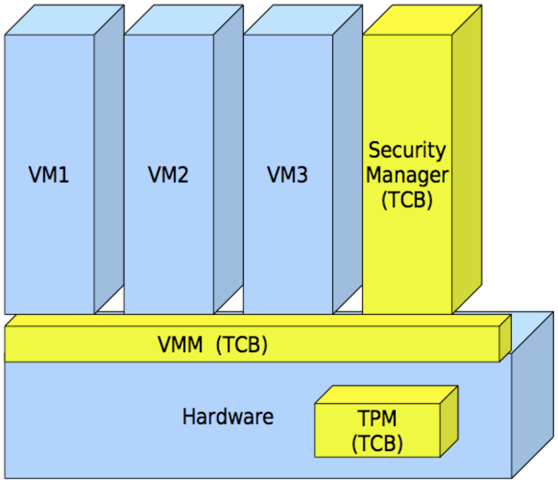

Figure 1: An illustration of the architecture for Abacus. The trusted computing base (TPM, VMM, and security manager) is shown in yellow. The rest of the hardware is untrusted, as are the other virtual machines for running applications.

can spawn virtual machines to run untrusted code in isolation and uses instrumentation to detect malicious actions on the part of this code. Without virtualization, trusted computing can attest to specific software configuration, but such systems are brittle and not easily changed. They provide no safe method for running untrustworthy code. Virtualization alone is not enough either. Without trusted hardware, there is no root of trust to build the system on: the VMM could be resting on top of a stealthy root kit like Blue Pill[15].

While such architectures have generally been limited to PCs, we also envision security managers running on other personal devices, like phones and PDAs. Increasing numbers of phones are running full-scale operating systems, and virtualization is becoming more and more efficient and lightweight. The continuing march of Moore's law will also make more layered approaches to security more feasible in the long term.

3. ADJUSTABLY AUTONOMOUS SECURITY

If Alice understood what Abacus was being asked to do by the programs that she is running on it, she would usually be able to make good judgments about what behaviors are suspicious. For example, if Alice runs a program to send greeting cards to her friends, she would be happy to see it send those cards, upset if it sent a list of her contacts or ads for Viagra to some random address, and might allow it to send its creator email with statistics on how many cards had been created.

Notice that these scenarios place us on the horns of a dilemma. On the one hand, these scenarios cannot be distinguished by what resources are involved, only by the semantic content of their usage, and we cannot expect Abacus to differentiate between them without help from Alice. On the other hand, computers do so many things so rapidly that asking Alice for many judgment calls will quickly overwhelm her and annoy her into disabling security.

As Abacus builds up information about Alice, it may be able to infer her goals and desires, and thereby help her in making security decisions. Currently, many security decisions must be passed off to bewildered users, because applications do not have the reasoning capability or contextual information to make these decisions [4]. Developing an agent that acquires this sort of contextual information may enable applications to make better security decisions.

What we need is a way for Abacus to filter and summarize its behavior so that Alice is only asked to make a few relevant judgment calls. For Alice to be able to rely on Abacus' judgment about what decisions need her input, Abacus needs to be able to judge how valuable and private a piece of information is, know what Alice is expecting a program to do, know what types of program behavior are worthy of suspicion, and simulate Alice's judgment so that it does not pester her when the answer should be obvious. Although this is an unsolved problem, closely related problems are being studied in other domains, such as collaborative planning[16] and adjustable autonomy[8].

Artificial intelligence problems are hard, and this is likely to be no exception. However, it is worth noting that, in other domains, a little bit of knowledge and reasoning capability often goes a long way—even small doses of AI may make Alice's job of securing her machine much easier.

4. CONCLUSION

Cognitive security will lead to better overall security because of the better match between a personal device and its user. This can be achieved by combining three lines of research: trusted computing and virtualization to protect the integrity of the device, imprinting with continuously deployed multi-modal biometrics to allow the device to recognize its owner, and human-like security models to enable responsible decision-making by the device. The greatest challenge is the last: human-like threat perception and response is a problem of potentially limitless complexity, and adversaries will exploit any systematic flaw in the design. Even a partial solution, however, may be effective enough to fundamentally change the trust relationship between users and their devices.

5. ACKNOWLEDGMENTS

This work has been partially funded by NSF Grant #6898853 and the Harvard Center for Research on Computation and Society. Thank you also to Stephen Murdoch for information about the current state of trusted computing research.

6. REFERENCES

[1] S. Berger, R. Caceres, K. Goldman, R. Perez, R. Sailer, and L. van Doorn. vtpm: Virtualizing the trusted platform module. In *15th USENIX Security Symposium*, July 2006.

[2] J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Combining biometric evidence for person authentication. In *Advanced Studies in Biometrics*. Springer, 2005.

[3] Roberto Brunelli and Daniele Falavigna. Person identification using multiple cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(10):955–966, 1995.

[4] W. Keith Edwards, Erika Shehan Poole, and Jennifer Stoll. Security automation considered harmful? In

IEEE New Secuirty Paradigms Workshop (NSPW), 2007.

[5] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, 55(1):119–139, 1997.

[6] Tal Garfinkel, Ben Pfaff, Jim Chow, Mendel Rosenblum, and Dan Boneh. Terra: a virtual machine-based platform for trusted computing. In *Symposium on Operating Systems Principles (SOSP)*, 2003.

[7] The Trusted Computing Group. http://www.trustedcomputinggroup.org.

[8] Eric Horvitz. Principles of mixed-initiative user interfaces. In *Conference on Human Factors in Computing Systems (CHI)*, 1999.

[9] L. Jones, A. Anton, and J. Earp. Towards understanding user perceptions of digital identity technologies. In *ACM Workshop on Privacy in the Electronic Society*, 2007.

[10] G. Leggett, J. Williams, and M. Usnick. Dynamic identity verification via keystroke characteristics. *International Journal of Man-Machine Studies*, 28(1):67–76, 1998.

[11] Konrad Lorenz. *Studies in animal and human behavior*. Harvard University Press, 1970.

[12] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *4th ACM conference on Computer and communications security*, pages 48–56, 1997.

[13] Robi Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[14] Arun Ross and Anil Jain. Information fusion in biometrics. *Pattern Recognition Letters*, 24:2115–2125, 2003.

[15] Joanna Rutkowska. Blue pill project. http://bluepillproject.org/.

[16] David Sarne and Barbara Grosz. Estimating information value in collaborative multi-agent planning systems. In *AAMAS 2007*, 2007.

[17] Andrew Simkin. Interrupting terrorist travel: Strengthening the security of international travel documents. http://www.state.gov/r/pa/ei/othertstmy/84339.htm, May 2007.

[18] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *7th International Workshop on Security Protocols*, 1999.