Patterns for Aligning Security and Usability

[Poster]

Simson L. Garfinkel
MIT CSAIL
Cambridge, MA
simsong@csail.mit.edu

Robert C. Miller
MIT CSAIL
Cambridge, MA
rcm@csail.mit.edu

ABSTRACT

We present a series of related patterns for aligning security and usability based on a substantial body of prior work. These patterns cover the issue of data sanitization, secure messaging, secure operation, and protection from covert monitoring.

1. INTRODUCTION

Patterns are widely used in many human endeavors that require a combination of skill and training. Architect Christopher Alexander pioneered the recognition, naming, and use of patterns while working on urban planning in the 1970s. [1][2] In the late 1980s computer scientists working in the field of object-oriented design discovered Alexander's work.[4] Schumacher [12] argues that there security engineering can benefit from the use of patterns, but he fails to present specific patterns to accomplish this goal. The Open Group has published a book of *Security Design Patterns,*[3], but have not addressed the alignment of human computer interaction with security (HCI-SEC).

We present patterns to address three specific areas of critical HCI-SEC research: the sanitization of information left on computer media; secure messaging; and promotion of secure operation by reducing the danger of covert monitoring.

The sanitization patterns are based on a study involving the purchase of 236 hard drives on the secondary market, interviews conducted with organizations whose drives had been acquired, and through a detailed examination of modern web browsers and reports of sanitization failures.[6, 8]

The secure messaging patterns are supported through an analysis of S/MIME handling in modern email clients, a survey of 469 Amazon.com merchants,[11, 10] and a user study of 43 individuals participating in an anti-spoofing experiment.[9]

Patterns are presented for promoting secure operation and for reducing the danger of covert monitoring. These patterns are supported by the literature review and an analysis of current systems.[5, 7]

In every case considered, it is shown that there are cases where the perceived antagonism of security can be scaled back or eliminated simply by revising the underlying designs on which modern systems are conceived. In many cases these designs can be implemented without significant user interface changes.

It is very likely that additional patterns can be identified in other related areas. These patterns can be directly applied by today's software developers and used for educating the next generation of programmers so that longstanding usability problems in computer security can at last be addressed.

2. THE SANITIZATION PATTERNS

We present five patterns for aligning usability and security in the realm of data sanitization. Following Alexander's dictum that good patterns are those that "actually make human life better as a result of their injection into the system,"[?], we feel that these patterns give people tools to sanitize their computers in a manner that is straightforward and that naturally arises as a result of normal computer operations.

- **Explicit User Audit**
  This pattern holds simply that users should be able to see all of the information that they are responsible for in the system that they are using.
- **Explicit Item Delete**
  This pattern holds that the tools for deleting information should be made available to the user where the information is displayed in the user interface. This is the pattern implemented by the DOS `DEL` and `ERASE` commands, by the Unix `rm` command, and by the graphical interaction metaphor of dragging an item to the trash.
- **Reset to Installation**
  The second way to delete information on a computer system is to reset the system to an installation state. This is analogous to the action of running the Windows `FORMAT` command or performing a "hard" reset on a PalmOS a computer.[?] Although many computer systems provide a Reset to Installation feature,

they do not implement that feature properly. That is because existing implementations of this pattern do not implement the COMPLETE DELETE pattern.

- **Complete Delete**
  Providing deletion functionality is not enough. The system must also assure that information is completely deleted so that it cannot be recovered. The standard way to implement COMPLETE DELETE on a computer system is to overwrite the data that is being deleted. Most computer systems do not do this. Instead they merely remove the link between the data and the memory containing the data, then mark the memory as "free" and available for re-use.
- **Delayed Unrecoverable Action**
  If computer systems are going to have the ability to perform unrecoverable actions, one way to prevent these actions from being performed in error is to institute a delay between the time that the action is invoked and the time that the action is performed. This specific interaction pattern is referred to as the DELAYED UNRECOVERABLE ACTION. It can be implemented with a timer and a mechanism for aborting the requested action.

We have shown, through an analysis of existing products, that implementing some but not all of these patterns can result in systems that have hidden failure modes, compromising the alignment of security and usability.

3. THE SECURE MESSAGING PATTERNS

We present a set of eight interrelated patterns for improving the security of both cryptographic and plain text communications:

- **Leverage Existing Identification**
  Use biometric and PKI-based systems to authenticate pre-existing relationships, rather than creating new ones.
- **Email Based Identification and Authentication**
  Use EBIA as a weak authenticator.
- **Send S/MIME-Signed Email**
  Organizations sending bulk do-not-reply email should send it signed with S/MIME signatures.
- **Create Keys When Needed**
  Software should automatically create keys and self-signed certificates when installed, rather than waiting for users to manually perform these operations.
- **Track Received Keys**
  Keep a local database of received keys and their history.
- **Track Recipients**
  Keep a database of mail recipient's cryptographic mail capabilities.
- **Directly Manage Key/Identity Bindings**
  When an X.509 certificate is received that is not signed by a known CA, directly manage the certificate's trust using the certificate history as guidance.
- **Distinguish Internal Senders**
  Visually distinguish between mail sent from within an email system with mail sent from outside the system that has the same `From:` address as internal senders.

4. OTHER PATTERNS

In addition to the patterns presented here, two more sets of patterns will be presented on our poster.

5. REFERENCES

[1] C. Alexander. *The Timeless Way of Building.* Oxford University Press, 1979.

[2] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: towns, buildings, construction.* Oxford University Press, 1977. (with Max Jacobson, Ingrid Fiksdahl-King and Shlomo Angel).

[3] B. Blakley, C. Heath, and members of The Open Group Security Forum. Security design patterns. Technical Report G031, The Open Group, Apr. 2004. URL `\url{http://www.opengroup.org/publications/catalog/g031.htm}`.

[4] E. Gamma, R. Helm, R. Johnson, , and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison-Wesley Professional, 1995.

[5] S. Garfinkel. The pure software act of 2006. *TechnologyReview.com*, Apr. 7 2004. URL `http://simson.net/clips/2004/2004.TR.04.PureSoftware.pdf`.

[6] S. Garfinkel and A. Shelat. Remembrance of data passed. *IEEE Security and Privacy Magazine*, January 2002.

[7] S. L. Garfinkel. Adopting fair information practices to low cost RFID systems, 2002. Paper presented at Privacy in Ubicomp'2002 workshop, Gotenborg, Sweden, September 29th, 2002.

[8] S. L. Garfinkel. *Design Principles and Pattersn for Computer Systems that are Simultaneously Secure and Usable.* PhD thesis, MIT, Cambridge, MA, Apr. 26 2005.

[9] S. L. Garfinkel and R. Miller. The johnny 2 standardized secure messaging scenario. In *Symposium on Usable Privacy and Security.* ACM Press, 2005.

[10] S. L. Garfinkel, E. Nordlander, R. C. Miller, D. margrave, and J. I. Schiller. How to make secure email easier to use. In *CHI 2005.* ACM Press, 2005.

[11] S. L. Garfinkel, J. I. Schiller, E. Nordlander, D. Margrave, and R. C. Miller. Views, reactions, and impact of digitally-signed mail in e-commerce. In *Financial Cryptography and Data Security 2005.* Springer Verlag, 2005. To Appear.

[12] M. Schumacher. *Security Engineering with Patterns: Origins, Theoretical Models, and Ne Applications.* Springer, 2003. LCNS 2754.