Toward Sensitive Information Redaction in a Collaborative, Multilevel Security Environment

Peter Gehres
peter-gehres@utulsa.edu

Nathan Singleton
nathan-singleton@utulsa.edu

George Louthan
george-louthan@utulsa.edu

John Hale
john-hale@utulsa.edu

Institute for Information Security
The University of Tulsa
800 S. Tucker Dr.
Tulsa, OK 74104

ABSTRACT

Wikis have proven to be an invaluable tool for collaboration. The most prominent is, of course, Wikipedia. Its open nature is not suitable for all environments; in corporate, government, and research environments it is often necessary to control access to some or all of the information due to confidentially, privacy, or security concerns. This paper proposes a method by which information classified at multiple sensitivity levels can be securely stored and made accessible via the wiki only to authenticated and authorized users. The model allows for each page to be viewed at appropriate levels of classification transparently included or excluded based on the user's access level.

Categories and Subject Descriptors

I.7.5 [**Computing Methodologies**]: Document and Text Processing; K.6.5 [**Computing Milieux**]: Management of Computing and Information Systems—*Security and Protection*

General Terms

Design, Security

Keywords

Redaction, Multilevel Security, Collaboration

1. INTRODUCTION

Collaborative, "wiki"-style models of information sharing have proven highly effective, as evidenced partly by Wikimedia's success as the 5th most visited website on the Internet with over 14 million articles in 271 languages [9]. Unfortunately, there are situations in which open collaboration is not possible, be it for privacy, security, or other concerns. Even in such environments, users may still benefit from the wiki collaboration style. Examples abound in military intelligence, where data must be protected in circumstances where the extreme requirements for security and confidentiality are often in conflict with the need for rapid and accurate information sharing.

To accommodate secrecy, a variety of multilevel security models have been proposed and implemented by the military and federal government. Unfortunately, information management systems have not been historically successful in blending support for sharing and collaboration with rigorous multilevel security policies. This paper proposes an architecture for a wiki solution that uses a redaction engine to support information sharing under a multilevel security model.

The remainder of this article is organized as follows. Section 2 of this paper gives background information on multilevel security, the Bell-La Padula model, and textual redaction. Section 3 presents related work. Section 4 introduces our model, Section 5, challenges associated with our research, and Section 6 discusses the challenges and future work.

2. BACKGROUND

2.1 Multilevel Security

Traditionally, multilevel security (MLS) models are found in military, intelligence, law enforcement, and other government agencies. In the MLS model, information is segregated based on its sensitivity, the nature, means and method of its collection; by specific topics; and/or the potential for damage caused by its untimely release. In the classic model, the primary levels are unclassified, confidential, secret, and top secret. Each primary level may have one or more sublevels or categories which further segregate information based on its specific subject matter.

Users, based on their specific duties and their security clearance, are granted access to these "compartments." These compartments contain all the information about a specific subject at the given sensitivity level. For example, an in-

WikiSym '10, July 7-9, 2010, Gdańsk, Poland

dividual working on aircraft jet engines who has a secret clearance is privy to information at the secret level about the engines and data about those engines classified at lower levels, but those with secret security clearance working in unrelated fields are not permitted access to the data. An information system capable of enforcing these constraints on interconnected systems is sometimes called a cross-domain guard.

This same principle applies to industry. For example, a company developing a new aircraft engine and a new radar system for the same new aircraft, likely has personnel working on the engine who do not need access to information about the radar. However, the program manager for the new aircraft does need access to information on both the radar and engine systems. Thus, the program manager can be granted access to both compartments while others are granted access only to the specific project on which they are working. The compartmentalization of this model is fundamentally incompatible with conventional access control and security models for wikis.

2.2 Bell-La Padula Security Model

Our architecture draws heavily from the Bell-La Padula (BLP) security model [2]. The BLP security model was designed for MLS-style systems and specifies an environment in which users are allowed to read information at or below their highest clearance level and write information at or above their clearance level. Writing below one's clearance level is disallowed to prevent users from improperly declassifying information. In the BLP model, each object is described by a 3-tuple, [o, c, k] where o is the object being secured, c is the classification of the object and k is the set of categories, if any, under which the object is classified.

Using the aircraft example from Section 2.1, "Aircraft" is the object, "SECRET" is the classification, and "Engine" and '"Radar" are the categories. The BLP tuple for a new aircraft's engine is therefore [Aircraft, SECRET, Engine]. Likewise, the tuple for the radar is [Aircraft, SECRET, Radar]. An employee with secret clearance working on the engine has the tuple [Aircraft, SECRET, Engine]. Because BLP allows for users to read at or below their clearance level in their categories, the employee could also view data classified with the tuples [Aircraft, CONFIDENTIAL, Engine] and [Aircraft, UNCLASSIFIED, Engine]. The program manager for the new aircraft project who oversees both the radar and engine projects would be given the tuple [Aircraft, TOPSECRET, [Radar, Engine]]. This ensures that the program manager has full access to all information and data concerning both the engine and radar regardless of classification.

2.3 Text Redaction

Redaction is the process of rendering a sensitive document suitable for distribution by removing sensitive information. In the MLS model, this is the removal of certain information to allow for the document to be reclassified at a lower level. Traditionally, redaction is accomplished using black markers to cover, or scissors to remove, portions of a photocopy of the document.

Hard copy redaction is easily understood and easily, if tediously, implemented. Digital redaction, on the other hand, can be difficult, requiring a multi-step process fraught with pitfalls related in part to the sometimes redundant formatting of files. Multiple types of data, including metadata, are all mixed together. The US National Security Agency states, "The complexity makes them potential vehicles for exposing information unintentionally, especially when downgrading or sanitizing classified materials" [1].

3. RELATED WORK

3.1 Intellipedia

In 2005, the United States Intelligence Community developed its own collaborative, wiki-style repository for intelligence-related information called Intellipedia and implemented using three distinct MediaWiki deployments [8]. Hosted on existing, secure, private and entirely separate networks accessible only to people with the appropriate clearances, Intellipedia is not available to the public [3].

Because the implementation of Intellipedia on multiple, distinct networks, a user with a high level clearance will not have access to the full wealth of information without logging in to multiple wikis. Pertinent information at all other security levels, even those a user is authorized to view, may be excluded, because the changes to a lower clearance level wiki cannot propagate to the others. This severely diminishes the advantages provided by the wiki model by requiring useful — perhaps vital — information to be manually replicated as many as three times.

3.2 Tearline Wiki

Tearline Wiki (TLWiki), developed by Galois, Inc., attempts to solve the data replication problem by aggregating the wikis onto a single page with "tearlines" between the differing classifications [5]. This eliminates the need for a user to log into multiple wikis, but still requires the user to read potentially repetitive information. Further design and implementation details are unavailable due to the proprietary nature of the software.

3.3 Multilevel Wiki for Cross-Domain Collaboration

Ong, *et al.*, developed a model with a similar goal based on the Monterey Security Enhanced Architecture (MYSEA) and TWiki, a system that uses a directory structure to model hierarchical classification structures [7]. The goal of the MYSEA project "is to provide a trusted distributed operating environment for enforcing multi-domain security policies" [6]. The authors specifically eliminated as candidates any wiki platform with a database backend in an effort to defer security decisions from the wiki itself to the filesystem layer. The model also does not specify how the multilevel information is presented to the users.

4. SECUREWIKI

We propose a model capable of combining information categorized at different security levels and topics into a single so-called logical "view" representing the most complete information its viewer is allowed to access. This is done by introducing an intermediary authorization and cross-domain guard layer between the data storage and presentation portions of the wiki. Because the system builds its logical presentation (called a view) upon access rights dynamically determined on a per-user basis, it provides desirable security properties while still maintaining its usability as a collaborative tool.

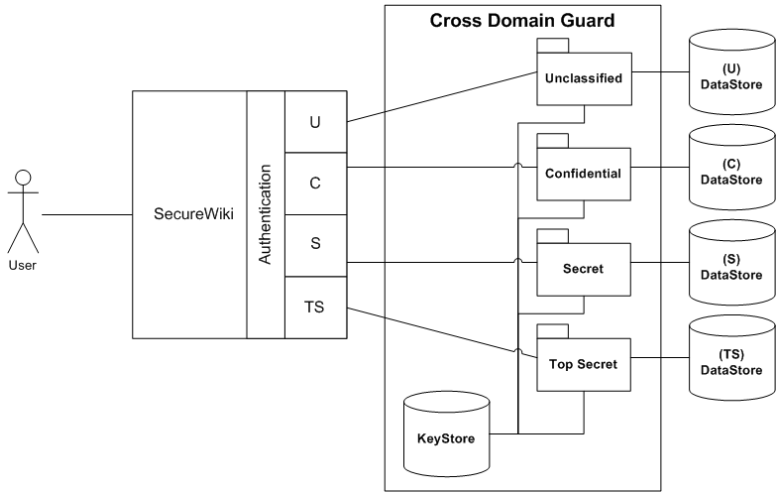

Figure 1: Architecture Overview

Each major security classification designates a data store containing all the necessary data to produce a complete wiki. As shown in Figure 1, top-secret information is stored separately from the unclassified, confidential and secret information. Like Intellipedia, then, this entails the existence of a distinct wiki for each security level. However, unlike Intellipedia, these separate wikis are dynamically combined to build a logical view of the entire knowledge base appropriate to an individual user's credentials.

4.1 Redaction Model

To store the data in such a way that sensitive information is redacted from views at lower clearance levels, we advocate a model that presents data by building a view from the lowest security level and incrementally adding more sensitive data until the authorized level for the user is reached. The storage process for sensitive data, then, must extract this data as it writes changes to the lower security levels.

The writing process ensures that the sensitive data is accessible by placing references to it in the lower security data store in the form of *redaction tokens*. A token can represent anything from a single letter to an entire page, inclusive of pictures, links, more redaction tokens and anything else that can be specified in wiki markup language. To specify information for redaction, the user will tag the data with a classification, either at or above the user's classification level, and any categories. The system will then replace the text with a unique token and insert the redacted text into the appropriate data store.

This process is repeated layer by layer, with each layer containing only information classified at its security level as well as redaction tokens that may or may not reference information at the next higher level.

4.2 System Architecture

4.2.1 Read Down

Once a user has authenticated, a view is prepared specifically for that user that functions essentially like any other wiki deployment; however, although it would be right to call this view a wiki, it is more precisely an abstraction constructed as a result of a layered reading process best illustrated by example. Following through Figure 1, suppose a user with clearance at the confidential level authenticates to the SecureWiki. The user's view, built via the appropriate gatekeepers, can read from the unclassified and confidential data stores.

When the user requests a page, the view first requests the page from the unclassified data store, which contains the base (unclassified) content, and possibly references (as redaction tokens). The render module parses the unclassified version of the page looking for redaction tokens, attempting to match them with data at the next higher security level (in this case, confidential), and either accesses or redacts the associated content.

If the gatekeeper determines that the user is authorized to access information represented by a token (as is the case in this example, because the user is cleared at the confidential level), it repeats the process with the new content and any redaction tokens that content may contain, attempting to match against the next higher security level (in this case secret). Because the user is only cleared for confidential information and therefore connected to a confidential view, the secret gatekeeper will deny all requests for resolution of redaction tokens classified at the secret level, ending the process.

Once all of the information for which the user is cleared has been obtained from the data stores, the render module will render the article and display it to the user.

4.2.2 Write Up

When a user edits or adds information to the wiki he will be asked to provide a classification level and any required category descriptors. This action triggers a write operation to one or more of the data stores (see Fig. 1). By the Bell-La Padula security model, the user may not write to data stores below the user's clearance level, and thus, users with a secret clearance may not write directly to the confidential data stores.

In the event that a user feels that information he has accessed is misclassified at too low a classification level, he will have the ability to raise the classification level of the data. If a user attempts to write to a higher classification level than the user's maximum (as is allowed by Bell-La Padula), the

system will redact the information to the higher level and place it in a temporary container for review by an approving authority at the higher classification level.

5. CHALLENGES

The SecureWiki model proposed in this paper is at a very early stage in development. Many issues have yet to be considered as we have yet to determine the appropriate wiki platform to extend. Potential challenges specific to implementation include the performance of the page reconstruction, issues related to concurrent editing at distinct security levels, and the usability of the application.

5.1 Metadata

There are often circumstances when the mere existence of a program or codeword is itself classified. Traditional wiki styling of link text could potentially violate these constraints. Implementation of the system must therefore take this into consideration and implement methods to prevent this potential security risk.

Additionally, it is possible for multiple pieces of data to combine in a way that the collection is more sensitive than its individual parts would be alone. This can happen, for example, in situations involving health-related information covered under HIPAA [4]. These situations must be addressed.

In a collaborative environment, the ability to see the history of an article can be very useful but also presents another challenge to securing the information. In the event that a piece of text is moved to a higher classification level, that text must also be redacted from the historical versions of the page.

5.2 Limitations of the Bell-La Padula Model

A major criticism of the Bell-La Padula model is that it does not allow for objects to move from a higher security lever to a lower security level in the event of declassification. Further, it prevents users from writing information to a security level lower than their own, regardless of the actual classification of the data. It may therefore be necessary to augment or potentially replace the Bell-La Padula model when implementing SecureWiki.

The model does provide a method by which users can login with a security level lower than their maximum authorized level and without access to certain compartments. This modification in a user's security level allows them to write to files that they would be unable to write to otherwise. This could present a potential security risk because, although the user may not have current access to more highly classified data on the system, the user may still have offline access and could write the information into the lower level. This must be considered as well during the implementation of the system.

6. CONCLUSIONS

Previous work in building collaborative wikis for multilevel security environments face serious usability hurdles, potentially requiring users to view multiple independent wikis covering identical topics at different security levels. SecureWiki constitutes a step toward a more truly collaborative model for high-security environments by integrating multiple security levels into dynamic, per-user views.

Future work must include answers to several remaining questions, such as the problems caused by metadata and inflexibilities in the backing formal security models, as well as implementation details. However, the SecureWiki architecture is an important first step toward improving collaboration and information sharing in high-sensitivity environments, both military and commercial.

7. ACKNOWLEDGMENTS

This paper is an outgrowth of many conversations with Mr. Philippe Beaudette, Head of Reader Relations, Wikimedia Foundation, whose knowledge of everything wiki has been instrumental in the development of this project.

The authors would also like to thank everyone who provided guidance on this paper especially Dr. Rose Gamble and the anonymous reviewers who have helped point us toward wiki-resources unknown to security researchers.

This material is based on research sponsored by DARPA under agreement number FA8750-09-1-0208. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, or DARPA or the U.S. Government.

8. REFERENCES

[1] U. S. N. S. Agency. Redacting with confidence: How to safely publish sanitized reports converted from word to pdf. page 54, December 2005.

[2] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report Technical Report 2547, MITRE, East Lansing, Michigan, March 1973.

[3] T. Clive. Open-source spying. *New York Times Magazine*, page 54, December 3, 2006.

[4] Federal Register, (P.L.104-191). *Health Insurance Portability and Accountability Act of 1996*.

[5] Galois, Inc. Information assurance: Capabilities, 2008.

[6] C. Irvine, D. Shifflett, P. Clark, T. Levin, and G. Dinolt. Monterey security enhanced architecture project. *Naval Postgraduate School*, April 2003.

[7] K. L. Ong, T. Nguyen, and C. Irvine. Implementation of a multilevel wiki for cross-domain collaboration. *Naval Postgraduate School*, 2008.

[8] S. Vogel. For intelligence officers, a wiki way to connect dots. *The Washington Post*, page A.23, August 27, 2009.

[9] Wikimedia Foundation. List of wikipedias, March 14, 2010.