

Managing Access Rights for Terminated Employees

Dennis Heimbigner
(Presenting Author)

Computer Science Department
University of Colorado
Boulder, CO 80309-0430
dennis.heimbigner@colorado.edu
Voice: 303-492-6643
Fax: 303-492-2844

Keywords: Adaptive security, password, access lists.

Abstract

A recurring security scenario involves the problem of removing the access rights of an employee upon employee termination. A solution is presented in the form of procedures and an infrastructure. Operationally, sensors track security-related user operations (e.g. password use and object accesses) to determine the true access rights associated with a user. Security administrators combine that information with recorded access controls and invoke actuators to perform actions necessary to remove the complete access rights of a user.

1. Introduction

When any employee leaves an organization, it creates the potential for a breach of security [3]. That employee will have had permission to access a number of resources, including potentially valuable ones. The employee will have knowledge of passwords, including some that are shared with others. In practice, it is difficult and tedious for security administrators to make sure that all permissions for a terminated employee are removed at the time of termination¹. Ideally, this removal

should be performed rapidly to minimize the opportunities for an employee to perform unauthorized activities. However, current approaches are mostly manual and slow and hence increase the period in which harm can occur.

The fundamental problem is that security-related information such as policies, authority grants, passwords, and access lists are distributed across multiple domains and across multiple machines in a network. For example, the set of objects (e.g., files and directories) whose access lists include the terminated user may be very large and the objects themselves may be distributed widely over a network of machines. Further, there may be many informal paths by which access is granted but not recorded. It is not uncommon for people to share passwords without bothering to go through any formal mechanism. The justification is usually that someone needs some information “now”, and that person is considered trustworthy.

Solving this problem requires the development of operational procedures combined with a comprehensive infrastructure capable of collecting the necessary information about users and making it available in a usable format for security administrators. The goal of this paper is to define that infrastructure (architecture and operation) for a system to help track and capture security-related activities, both formal and informal, in order to help security administrators more rapidly remove a

¹ The term “terminate” is used broadly to refer to firing, layoffs, and resignations.

terminated user from their system. We will use the term “excise” for this process of removing a user’s security rights from the system. Note that the excision may be a limited operation. That is, the affected user may still have legitimate access to the network, but only have his rights removed from a subset of resources on that network. Additionally, he may be removed from a single domain of a multi-domain network. So excision must be viewed as a *selective* removal of access rights.

2. Architectural Overview

Figure 1 illustrates the high-level architecture for our system. The bottom of the figure illustrates the network of software systems and machines being managed by the excision control system. This monitored system consists of machines, operating systems, and applications.

Above the monitored system is the excision control system itself. Physically, it operates on a superset of the machines comprising the monitored system. It has software agents on the network of machines, but it presumably includes some extra machines used solely to execute the control system software.

The excision system operates in a basic sense-plan-act control loop. The left side indicates that the various applications, operating systems, and security systems utilized by users are modified to sense security-related user events and forward those events to the control program managed by the security administrator. As a rule, event generation is *re-active* in that it occurs in response to specific activities such as login or password use.

This control program maintains a database to record events and to build up a portfolio about each user. In addition, this control program provides an interface for the administrator to peruse the events and portfolios, and to initiate the activities necessary for removing a user from the system.

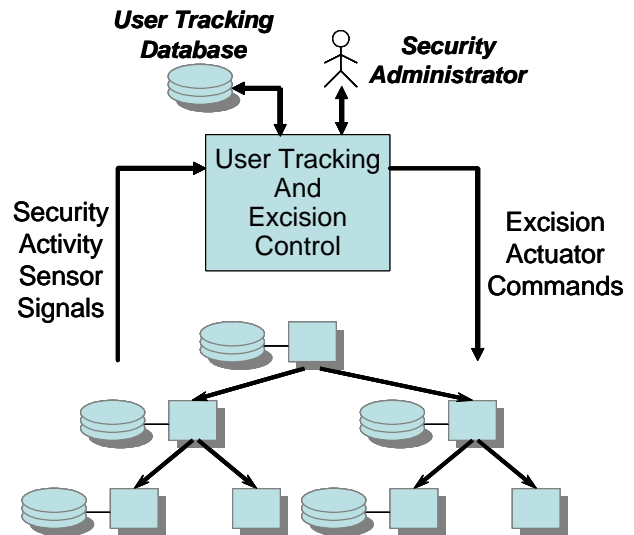


Figure 1. Excision Infrastructure Architecture

The right side of the figure illustrates the actuator path by which excision actions by the security administrator are distributed to various systems. These actions include such things as changing access lists or passwords.

The actuator path supports an important additional capability: *pro-active* sensing. That is, some sensors may only be activated when given specific commands to do so. For example, the periodic determination of what objects can be accessed by a given user may only occur when invoked by the security administrator.

3. Data Collection

The key operational capability required to solve the problem of the terminated user is to track and capture the actual security-related activities. This includes such things as actual password use and actual accesses to controlled objects.

Central to our solution is the database that logically centralizes two kinds of information about users. First, the database must record the actual resources accessed by the user. This includes such things as specific passwords used, specific policy changes (i.e., setting access rights), and specific files read or written. Second, the database must represent the

complete set of resources known to be accessible to each user. Note that this is effectively the inverse of typical security information stored as access lists, which map resources to users. Here, however, we are mapping users to resources. This is often referred to as capability lists [7], although we do not use it here in quite the fine-grained way that true capability lists are used for access enforcement.

3.1. Activity Monitoring

The database is fed by a stream of security-related events captured by various *sensors* inserted into various software programs in the network. These programs include applications, database systems, web servers, operating systems, and security programs (the *login* program for example). The success of the excision process depends on the amount of sensing information that can be gathered, and that depends on the degree to sensors can be inserted into various software systems.

3.2. Accessibility Monitoring

User-oriented access lists constitute the other major class of information kept in the database. This per-user information collects together the knowledge about all the resources explicitly accessible – or explicitly denied – to the user. This information is essential for rapid excision of the user.

Capturing access list information is straightforward in principle, although there are some practical difficulties. The basic idea is to harvest an initial set of rights from all nodes of a network. This initial database of rights is then updated in two ways. First, the sensors may generate access list modification events that can be used to keep the database current. This requires the sensor network to be able to capture *all* such events, which may not be possible for all operating systems and especially for user-level applications such as database systems that may keep their own sets of access lists. Alternatively, the harvesting of access rights

information can be repeated periodically to completely refresh the database.

4. User Identification

When multiple administrative domains are involved in excision, then the problem of user identification arises. The problem is that the real person may be assigned different login names in the different domains.

It is possible to construct heuristic procedures for finding potentially matching names. This procedure takes as input the complete user records from two (or more) domains. By looking for partial matches in the login names, phone numbers, and other fields, the procedure can identify records of potentially similar users. The fall back procedure is, of course, manual matching.

4.1. Single Domain Aliasing

It is possible that the same person will have access to multiple login names even within a single domain. The alias may come about by deliberately establishing a second login, or because one user happens to know the login password for another account, and may use it with no formal acknowledgement that the login is an alias.

In this latter case, detecting the occurrence of such informal aliasing may require significant analysis of the events associated with multiple users. If, for example, the alias login is asserted on the same computer as some user, and further, the alias login is bracketed by logins by the user, then suspicion may be raised that aliasing is occurring.

5. Effecting Excision

At the point in time that an employee leaves an organization, the security administrator must initiate the necessary actions to selectively remove access rights of a user to system resources.

These actions are initiated by the security administrator, of course, but the detailed sets of

actions are performed by the infrastructure. Thus, the administrator may issue the high-level command to remove a user completely from domain D. However, the exact set of resources and machines that are in domain D and the procedures for removing a user from that domain are mostly automated based on information kept in the excision database.

In order to perform an excision, it must be possible to remove or modify every access right for a given user at every relevant machine in a network. The actions may be carried out using remote login and command lines, or they may require agent software that is resident on every machine and that takes its commands from the excision control program.

6. Password Management

Passwords represent an important special case for excision. This is because there is no way to track which users have knowledge of a given password except at the point in time when the password is used to access a resource. This is in contrast to access lists, which can be examined and analyzed at any time (but with some cost) to determine which users have access to which resources.

The primary capability required to manage passwords is the ability to generate events recording each use of a password on a per-user basis. This is not a capability currently provided by any system of which we are aware. Implementing such a capability again requires inserting sensors into the appropriate applications and operating systems in order to capture password usage.

In order to remove a given user's access to password protected resources, it must be possible to modify the password associated with a given resource. This is often a difficult procedure to automate because many GUI-based systems do not support any sort of automated programming interface for changing passwords.

There is a secondary issue involved in changing passwords when the controlled

resource is shared by multiple users. In this situation, changing the password will prevent access by the targeted user, but may also prevent access by all other users who also know and legitimately use that password. So as part of the excision process, it must be possible to generate and apply a new password for shared resources. It must then be possible to notify the other affected users about the password change. This is possible if the appropriate sensors are in place so that the control database (Section 3) can determine all of the users who access a given password-protected resource.

7. Legal Issues

We expect that our solution to the terminated employee problem will raise a number of legal issues. We are not legal experts so this discussion is of necessity tentative.

The primary problem is that we are observing, in real-time, the behavior of employees. There is some precedent for this with respect to employer observation of email. In addition, we have all telephoned to organizations and heard that "calls may be monitored for quality control purposes." However, the legal foundations for the kind of detailed monitoring we are proposing here have yet to be explored.

Another issue involves inter-domain notifications when a user is fired. There are currently limits to what one organization can tell another about the employment status of an employee. For example, suppose that a computer support company fires one of its hardware maintenance people. In that situation, it should notify all of the companies with which it consults to excise all access because the person was fired. This may be illegal. What is required is some form of trust between the organizations such that when the consulting organization says to remove someone, client organizations trust it sufficiently to do the removal without asking for justification.

8. Related Work

The infrastructure we define here represents an example of what is coming to be called “adaptive security.” Adaptive security involves the sense-plan-act process but specifically oriented to security. The current security state is sensed, it is analyzed to determine how to repair or protect it, and then actions are taken to carry out the repairs. The excision system described in this paper represents the third example of such a system that we have developed. The other two are as follows.

1. Willow. The Willow system [6][9] was our first adaptive security system. The goal was to sense attacks against a distributed software system and to repair or protect it using dynamic reconfiguration of the target system.
2. Insider-Threat. The insider-threat system [1] has as its goal the sensing of document transactions, their analysis to detect suspicious user actions, and the execution of actions to slow down or completely thwart the malicious insider activities.

The excision system described in this paper follows in the Willow and Insider-Threat path. It is looking at a different set of sensed events, and it supports control and action to achieve different ends.

It should be noted that some security systems capture a subset of the information needed for excision. Almost universally, however, this information is stored in log files and rarely viewed or utilized again (except sometimes for forensic purposes). It does point out, however, that logs can provide a fruitful source of events to feed into our excision system.

The access revocation problem [4] is closely related to this work. Revocation research has mostly focused on the theoretical issues involved in revoking access to resources. The theory is, however, of limited use until infrastructure exists that can provide the

relevant subject-object information upon which revocation algorithms operate. Further, such access revocation algorithms are of limited use for password protected resources because formal access lists may not exist. Our excision infrastructure provides such infrastructure and hence is a good complement to that revocation research.

The problem of user identification is a special case of the more general federated database integration problem [2]. The solution we propose (Section 4) is essentially the same as was taken in those earlier systems.

Recently [8], Microsoft has developed an architecture for managing access to enterprise information. They refer to it as the Rights Management System (RMS). RMS involves a centralized repository of access rights information about documents. Every application that wishes to access a document also accesses the repository in order to validate the access to that document.

The RMS system is currently only focused on enforcing access control policies regarding documents. As such it is not currently capable of performing the kinds of excision capabilities identified in our work. However, it is probable that it could be extended to support such a capability if it was merged with the infrastructure described here. This would be advantageous for everyone because Microsoft can provide a deep integration of the necessary sensors into its own products.

9. Status

Development of the excision infrastructure described here is in its earliest stages. It is intended to operate as an extension of our existing Insider-Threat infrastructure [1], which provides the basic support for sense-plan-act. The difference then comes in the kinds of sensed events, the control operations, and the actuation operations.

10. Future Work

As indicated in Section 7, the legal issues need to be explored and this work must be done jointly with legal experts.

An interesting addition to this work is to combine it with some of the Willow work to detect attempts by terminating users to insert Trojan horses into programs. The sensors would be augmented to record the files accessed by the user just before termination. Separately, these files would be carefully scrutinized using, for example, Tripwire [5], to make sure that no malicious modifications had occurred.

11. Acknowledgements

This material is based in part upon work sponsored by DARPA, SPAWAR, and AFRL under Contracts N66001-00-8945, F30602-00-2-0608, and F49620-01-1-0282. The content does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

12. References

- [1] Anderson, K., A. Carzaniga, D. Heimbigner, and A.L. Wolf. "Event-based Document Sensing for Insider Threats". University of Colorado, Computer Science Technical Report CUCS-968-04. February 2004.
- [2] Heimbigner, D. and D. McLeod. "A Federated Architecture for Information Management". *ACM Transaction on Office Information Systems* 3(3):253-278 (July 1985).
- [3] Jones, A.K. "Summary of Discussion at a Planning Meeting on Cyber-Security and the Insider Threat to Classified Information". Computer Science and Telecommunications Board of the National Research Council of the National Academies. November 1-2, 2000.
- [4] Karger, P. and A. Herbert. "An augmented capability architecture to support lattice security and traceability of access". *IEEE Symposium on Security and Privacy*, pages 2-12, 1984.
- [5] Kim, G. and E. Spafford. "Writing, Supporting, and Evaluating Tripwire: A Publicly Available Security Tool". *Proc. of the USENIX UNIX Applications Development Symposium*, pages 88--107, Toronto, Canada, 1994.
- [6] Knight, J., D. Heimbigner, A. Wolf, A. Carzaniga, J. Hill, and P. Devanbu. *The Willow Survivability Architecture*. *Proc. of the Fourth Information Survivability Workshop*, Vancouver, B.C, March 2001.
- [7] Lampson, B. "Protection". *Proc. Of the 5th Princeton Conf. on Inf. Sciences and Systems*, Dept. of E. E., Princeton University, Princeton, N. J., March, 1971, pp. 437-443.
- [8] Microsoft Corporation. *Microsoft Rights Management Solutions for the Enterprise: Persistent Policy Expression and Enforcement for Digital Information*. February 20, 2003.
- [9] Wolf, A.L., D. Heimbigner, J. Knight, P. Devanbu, M. Gertz, and A. Carzaniga. *Bend, Don't Break: Using Reconfiguration to Achieve Survivability*. *Proc. of the Third Information Survivability Workshop*, Boston, Mass., October 2000.