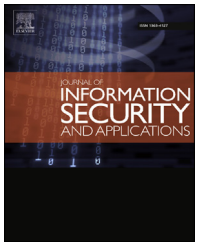
JOURNAL OF INFORMATION SECURITY AND APPLICATIONS ■■ (2016) ■■–■■

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/jisa



Identification of irregularities and allocation suggestion of relative file system permissions

S. Parkinson *, *A. Crampton*

Department of Informatics, School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK

ARTICLE INFO

Article history:
Available online

Keywords:
Access control
Auditing
Decision support

ABSTRACT

It is well established that file system permissions in large, multi-user environments can be audited to identify vulnerabilities with respect to what is regarded as standard practice. For example, identifying that a user has an elevated level of access to a system directory which is unnecessary and introduces a vulnerability. Similarly, the allocation of new file system permissions can be assigned following the same standard practices. On the contrary, and less well established, is the identification of potential vulnerabilities as well as the implementation of new permissions with respect to a system's current access control implementation. Such tasks are heavily reliant on expert interpretation. For example, the assigned relationship between users and groups, directories and their parents, and the allocation of permissions on file system resources all need to be carefully considered.

This paper presents the novel use of statistical analysis to establish independence and homogeneity in allocated file system permissions. This independence can be interpreted as potential anomalies in a system's implementation of access control. The paper then presents the use of instance-based learning to suggest the allocation of new permissions conforming to a system's current implementation structure. Following this, both of the presented techniques are then included in a tool for interacting with Microsoft's New Technology File System (NTFS) permissions. This involves experimental analysis on six different NTFS directory structures within different organisations. The effectiveness of the developed technique is then established through analysing the true positive and true negative values. The presented results demonstrate the potential of the proposed techniques for overcoming complexities with real-world file system administration.

1. Introduction

File systems are common amongst the majority of computer operating systems, and from a user perspective, their primary use is to store files in an organised and accessible manner. Modern, multi-user computer systems contain high quantities of data that require strong access control mechanisms to restrict data access to intended users. Different operating systems provide different implementations of access control. However, common to the most prevalent is that they provide a customisable architecture for access control. This is implemented through the use of both *coarse*- and *fine*-grained permissions (De Capitani di Vimercati et al., 2003). Coarse-grained permissions are predefined levels (e.g. read, write, full control, etc.) and fine-grained permissions are customised

* *Corresponding author.* Department of Informatics, School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK. Tel.: 01484 472525.
E-mail address: s.parkinson@hud.ac.uk (S. Parkinson).

http://dx.doi.org/10.1016/j.jisa.2016.04.004

Please cite this article in press as: S. Parkinson, A. Crampton, Identification of irregularities and allocation suggestion of relative file system permissions, Journal of Information Security and Applications (2016), doi: 10.1016/j.jisa.2016.04.004

permissions created from a set of predefined attributes to represent highly customised access control rules.

Administration of file system permissions on large file systems is both a challenging and a cumbersome task as there is a large volume of information to consider. Due to the complexities of managing permissions, unforeseen weak and incorrect allocations are often made. These complexities are usually the result of there being a large number of directories to secure, a large number of users that need to be correctly assigned, and a large number of access control rules. There is a wide range of literature discussing these complexities (Beznosov et al., 2009; Cao and Iverson, 2006; De Capitani di Vimercati et al., 2003), and many factual guides have been produced for different operating systems (Solomon, 2005; Thomas, 2010).

When analysing for vulnerabilities within file system permissions, they can be divided into two groups of (1) *known* system vulnerabilities and (2) those *relative* to the access control structure implemented within a system. A simplistic example of a known system vulnerability is that users should not have access to an important system directory (e.g. C:\windows\system32). These are programmatically easy to find by using a predefined knowledge base of potential vulnerabilities. Identifying such vulnerabilities is at most $O(n \times v)$, where n is the number of access control entries to examine and v is the number of known vulnerabilities. An example of a relative vulnerability is an incorrect assignment of permission with respect to an organisation's implementation of access control. For example, the *anomaly* of one user having write privileges on a directory where all other users have only read access. Such anomalies are very difficult to identify within access control as there is no quick method of determining potential vulnerabilities. Throughout this paper the words anomaly and irregular are used interchangeably to state a permission which deviates from what is identified as usual permissions within a given directory structure.

In addition to the examination of permissions, allocating new permissions whilst complying to known good practice is typically part of a system administrator's job and will often require collaboration (Haber et al., 2011). Maintaining good practice when implementing new permissions is achievable through the use of well defined guides (Govindavajhala and Appel, 2006; Russinovich et al., 2012; Russinovich, 2012). However, less well defined, and often more complicated to perform, is the implementation of new permissions in-keeping with the current implementation of access control.

The aim of this paper is to introduce a technique for identifying irregular file system permissions, as well as suggesting suitable allocation methods that are relative to a system's access control structure. The research hypothesis addressed in this paper is: using statistical methods on file system permissions can assist in identifying potential vulnerabilities without programatically encoding knowledge. This is accomplished by developing suitable models and techniques and then testing them on Microsoft's New Technology File System. The primary contributions presented in this paper are:

- **A novel technique for identifying irregular file system permissions**. This technique takes an *object-centred* modelling perspective to file system access control. Following this, a statistical analysis (χ^2) technique is used to identify permissions which fail a test of dependence. Empirical analysis is then provided to evaluate the effect of directory complexity and the frequency distribution of permissions.
- **Novel use of an instance-based learning algorithm for allocations suggestion of relative permissions**. A technique is developed which implements a k-NN algorithm and can aid with the implementation of new file system permissions by suggesting suitable access control rules based on the system's current implementation. Once again, empirical analysis is then provided to evaluate the effect of directory complexity and the frequency distribution of permissions.
- **A novel tool for interacting with Microsoft's NT file system**. Although the techniques provided in this paper are file system independent, a file system dependent tool (ntfs-r.exe) has been implemented for interacting with NT file systems. This tool implements a depth-first recursive directory search and implements an algorithm for efficiently calculating the *effective* permission of an interacting object.
- **Empirical analysis and case study**. The significance of the proposed techniques and tool is then evaluated through the use of many real file systems to determine the detection results (i.e. false positive and false negative rate). These results show an accuracy fraction of 0.911 for permission analysis, and 0.805 for permission allocation (see Section 6.2 for explanation).

The remainder of the paper is organised as follows: In Section 2, information regarding the structure of access control on different platforms is discussed. This then leads to a discussion regarding the administrative complexity that is a result of their implementation. Section 3 contains a survey of related work and discusses state-of-the-art in permission administration and auditing. Section 4 presents the development of a model which is then used alongside a statistical test of independence and is used to determine irregular permissions. Examples and empirical analysis regarding performance and accuracy are also presented. Following this, a model is then developed in Section 5 to be used alongside an instance-based learning algorithm to aid with the suggestion of new allocations. In Section 6 the development of a tool which implements the proposed techniques for the NT file systems is presented. Following this, the tool and the proposed techniques are evaluated in Section 6.2 on multiple real-world file systems to determine their efficiency and correctness.

2. Background and overview

Access control is typically defined as a relational model over the following domains: O the set of objects (i.e users), the set of resources R and the set of permissions P. Access control is a characteristic function on the set $A \subseteq S \times O \times R$. A subject s is granted permission r over resource o iff $\langle s, o, r \rangle \in A$. Access control models are typically called the access matrix. In many operating systems the access matrix is stored as an access list, which is associated with a resource object and is used to list all subjects and their permissions. In NTFS, access lists are implemented as Discretionary Access Control List (DACL)

Please cite this article in press as: S. Parkinson, A. Crampton, Identification of irregularities and allocation suggestion of relative file system permissions, Journal of Information Security and Applications (2016), doi: 10.1016/j.jisa.2016.04.004