Expert Systems With Applications 55 (2016) 274–283

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Auditing file system permissions using association rule mining

S. Parkinson*, V. Somaraki, R. Ward

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

ARTICLE INFO

Keywords:
Access control
Auditing
Association rule mining

ABSTRACT

Identifying irregular file system permissions in large, multi-user systems is challenging due to the complexity of gaining structural understanding from large volumes of permission information. This challenge is exacerbated when file systems permissions are allocated in an *ad-hoc* manner when new access rights are required, and when access rights become redundant as users change job roles or terminate employment. These factors make it challenging to identify what can be classed as an irregular file system permission, as well as identifying if they are irregular and exposing a vulnerability. The current way of finding such irregularities is by performing an exhaustive audit of the permission distribution; however, this requires expert knowledge and a significant amount of time. In this paper a novel method of modelling file system permissions which can be used by association rule mining techniques to identify irregular permissions is presented. This results in the creation of object-centric model as a by-product. This technique is then implemented and tested on Microsoft's New Technology File System permissions (NTFS). Empirical observations are derived by making comparisons with expert knowledge to determine the effectiveness of the proposed technique on five diverse real-world directory structures extracted from different organisations. The results demonstrate that the technique is able to correctly identify irregularities with an average accuracy rate of 91%, minimising the reliance on expert knowledge. Experiments are also performed on synthetic directory structures which demonstrate an accuracy rate of 95% when the number of irregular permissions constitutes 1% of the total number. This is a significant contribution as it creates the possibility of identifying vulnerabilities without prior knowledge of how to file systems permissions are implemented within a directory structure.

1. Introduction

File systems are common amongst the majority of computer operating systems and from a user perspective their primary use is to securely store files. 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, but 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, Paraboschi, & Samarati, 2003). Coarse-grained permissions are predefined levels (e.g read, write, full control, etc.) and fine-grained permissions are customised permissions created from a set of predefined attributes to represent highly customised access control rules. Using a mixture of both types of permissions provides a flexible architecture which can accommodate a large verity of different access control levels. However, this flexible nature also provides the possibility for complex permission relationships and anomalous permissions which often go undetected.

Administration of file system permissions on large file systems is both a challenging and cumbersome task as it is often difficult to conceptualise the large volumes of information available. 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, Inglesant, Lobo, Reeder, & Zurko, 2009; Cao & Iverson, 2006; De Capitani di Vimercati et al., 2003), and many factual guides have been produced for different operating systems (Solomon, 2005; Thomas, 2010). However, even with such guides, users are left to analyse the large amount of access control information independently. The level of their knowledge regarding access control and their system's configuration often directly determines the quality of their audit, and thus produces a heavy reliance on expert knowledge.

* Corresponding author. Tel.: +44 1484472525.
E-mail addresses: s.parkinson@hud.ac.uk, parkinson.sl@gmail.com (S. Parkinson), v.somaraki@hud.ac.uk (V. Somaraki), r.ward@hud.ac.uk (R. Ward).

http://dx.doi.org/10.1016/j.eswa.2016.02.027
0957-4174/

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 trivial example of a known system fundamental 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 read access. Such anomalies are very difficult to identify within access control as there is no quick method of determining potential vulnerabilities. The consequences of both types of vulnerabilities can be severe and can be generated from both user and software actions. For example, if a user has an elevated level of permission on a network directory structure, they could unintentionally modify or remove important data. A more severe consequence would be if the user could access data which is sensitive, as this could result in the organisation breaching the Data Protection Act. It is also possible that software (such as viruses, etc.) executing under the user's credentials could exploit their file system permissions to perform malicious activity.

Trend mining, in the form of Association Rule Mining (ARM) (Ma, 1998), has been extensively used to identifying anomalies and irregularities in many different application areas (Cheng, Lu, Liu, & Huang, 2015). ARM is a method of automatically identifying interesting relationships amongst variables in large datasets. Interesting relationships are often those that frequently occur; however, in some applications, such as the one presented in this paper, an interesting relationship is one which occurs infrequently. There have been many successful applications of ARM for identifying interesting (both frequent and infrequent) relationships in large datasets. For example, in finance (Barak & Modarres, 2015; Yu, Chen, Wang, & Lai, 2009), medical data (Somaraki, Harding, Broadbent, & Coenen, 2010; Somaraki, Vallati, & McCluskey, 2015), and cellular networks (Khatib, Barco, Gmez-Andrades, Muoz, & Serrano, 2015). The use of ARM is therefore a natural selection for applying to the challenge of identifying irregularities in file system permissions.

The work presented in this paper is tailored towards Microsoft's New Technology File System (NTFS); however, it can be easily adapted to other file systems. The majority of multi-user environments in organisations will use Microsoft's NTFS for providing a distributed mechanism of file storage. There are many complexities associated with administrating and auditing file system permissions on Microsoft's NTFS which can result in the creation of vulnerabilities. The complexity of identifying these vulnerabilities has resulted in a unhealthy reliance on expert knowledge. The work presented in this paper helps to remove this unhealthy balance on expert knowledge and provide a method of auditing file system permissions for all NTFS users. This paper first provides a review into related work in the area of aiding auditing of file system permissions. The next section the provides a description of NTFS access control structure, highlighting auditing complexities. At the end of this section, detail is provides on how file system permissions are modelled in the work presented in this paper, including the use of an algorithm to combine permissions to determine the effective permission. This then leads to a description of association rule mining techniques which are useful for identifying irregularities in file system permissions. In this section the chosen technique is also discussed and justified. Empirical observations are then provided where the developed technique is tested on five diverse real-world file systems. In this empirical observation, the results are compared to those of a domain expert.

2. Related work

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) models, where only the owner of a resource is authorised to change its access permissions. However, in multi-user environments it is possible for any user with sufficient permission to take ownership of a resource or change its permission.

Other operating systems also implement access control lists to manage file system permissions. Unix (including Max OSX and Linux) and Portable Operating System Interface (POSIX) compliant systems have a simple system for managing individual file permissions. This is the ability to assign a predetermined set of coarse-grained permissions (often called "traditional Unix permissions"). Most of these systems also support the use of Access Control Lists (ACL), such as POSIX.1e ACLs (coarse-grained) (Nemeth, 2010) or NFSv4 ACLs (fine-grained) (Pawlowski et al., 2000). Interestingly, the implementation between NFSv4 and NTFS ACLs is very similar and also caters for fine-grained permissions. The ability to create fine-grained permissions significantly increases the complexity of the access control implementation. For this reason, and due to strong similarities between NFSv4 and NTFS, the rest of this section will describe in more detail the access control implementation of NTFS.

There are many tools available to assist with the examination of NTFS permissions allocation (Microsoft, 2006a; 2006b). However, there is one common weakness in that they all still require expert knowledge to analyse the output of the tools to determine if there are any weaknesses. Previous work in the area of file system permission analysis resulted in the development of a novel tool for permissions administration that allows users to easily view file system permissions for large directories (Parkinson & Crampton, 2013; Parkinson & Hardcastle, 2014). The tool provides features to help the user identify vulnerabilities and view necessary information to make better informed permission allocations. For example, the reduction in reported permissions by not displaying inherited permissions, and allowing the user to filter and show effective permission for a specified user, are two prominent features.

Identifying anomalies or irregularities in system security is by no means a new topic (Bhuyan, Bhattacharyya, & Kalita, 2014; Lazarevic, Ertöz, Kumar, Ozgur, & Srivastava, 2003). For decades, researchers have been developing techniques and tools to identify security anomalies. For example, recent works have covered topics from the identification of anomalous user behaviour in social networks (Viswanath et al., 2014), anomalies in network traffic (Catania, Bromberg, & Garino, 2012; Mahoney, 2003), anomaly detection in wireless networks (Islam & Rahman, 2011; Xie, Han, Tian, & Parvin, 2011), and the anomaly detection in power station security (Ten, Hong, & Liu, 2011). All these studies generate good results and demonstrate the potential of using machine learning and statistics to identify anomalies. Recent research in the area of file systems includes the construction of a Bayesian network and neural network from the predetermined knowledge of the manipulation of file system artefacts for file system forensic analysis (Khan, 2012). Research has been carried out into developing tools