A methodology to minimise excessively permissive security configurations

RUKSHAN ATHAUDA, GEOFF SKINNER, BRIAN REGAN
School of Design, Communication and IT
The University of Newcastle
Callaghan, NSW 2308
AUSTRALIA
{Rukshan.Athauda, Geoff.Skinner, Brian.Regan}@newcastle.edu.au
http://www.newcastle.edu.au/school/design-communication-it/

Abstract: - Today's complex IT systems and multitude of possible permission configurations create a challenge for IT administrators, especially in determining optimal permission configuration for user groups. This is further exaggerated with the users' privilege requirements not being clearly specified or available. This typically leads to excessively permissive security configurations in IT systems which results in security vulnerabilities. This paper proposes a methodology and high-level architecture for a system that enables to elicit and deploy IT permissions in a convenient and secure manner avoiding many pitfalls that exist today. The proposed methodology's applicability is illustrated using two scenarios: a typical organisation with complex security requirements and a collaborative online environment.

Key-Words: - privilege requirements, security configurations, excessively permissive security configurations

1 Introduction

Today's IT environments typically consist of a multitude of IT systems and networks. This results in the IT administrator with the choice of configuring thousands of security configurations variables and rules. The IT administrator's goal of providing optimal access and security to IT resources is a challenging task. In practice, security permissions are not configured optimally [10] resulting in vulnerable systems and networks. Further, it is typical that different IT systems require different sets of permissions for different user groups and also these permission requirements are rarely specified clearly further exacerbating the situation. This paper outlines a methodology and an implementation architecture whereby users' security requirements are elicited in a convenient secure environment circumventing many issues faced today.

This paper is introduced with a scenario from a university environment that discusses how privileges are granted to students by IT administrators. The scenario describes issues pertaining to eliciting and deploying security permissions in today's complex system configurations. We revisit the scenario throughout the paper for illustration purposes. Later (in section 4), we extend our discussion to online collaborative environments illustrating the proposed methodology's applicability in different IT environments.

Scenario: The academic staff members from the Department of IT provide requirements to the IT support staff at the beginning of every semester so that the computer laboratories are configured for providing access to IT resources for students. The requirements include installation and configuration of wide ranging software on different OS platforms. These requirements are typically sent via email to the IT support staff. Before the images for the lab machines are deployed, the parent imaged machine is available for IT academics to test before deployment.

Typically, the requirements pertaining to the specific software packages are tested and verified for smooth functioning of labs. However, the security and access privileges are rarely scrutinized. A preferred security policy is to "Allow access only to the required resources to authorised users". A point to note is that the policy while giving access to allowed resources for a user, does not allow excessive privileges than required (i.e. privileges to access unauthorised resources), which results in a vulnerability. Specifying such access privilege requirements for IT systems is neither simple nor straight-forward, especially with multiple applications and complex configurations.

Consider the following instance where SQL Server is installed on a Windows Operating System. The laboratory requirements may require students to create databases, users, logins for SQL databases and include backing up/restoring databases. These specifications require students to have relevant privileges on SQL Server. There are a number of possible ways to grant these privileges:

- *Option 1*: Granting relevant privileges at SQL Server level to student logins
- *Option 2*: Providing database administrator privileges to student logins which provides unrestricted access to the SQL Server database server installed locally
- *Option 3*: Providing Windows administrator privileges on the local machine, which automatically maps as an administrator to the locally installed SQL Server

It is obvious that following options 2 (database administrator) and 3 (Windows administrator) approaches provides more security privileges than required to perform tasks in the laboratory resulting in security vulnerabilities. Administrators however may opt for options 2 or 3 and option 3 preferably for a number of reasons. Some of these reasons are discussed below.

Reasons for providing excessive privileges: There are a number of reasons that may lead to excessive privileges being granted to user groups:

- *Hard for users to specify security requirements*: There is an inherent difficulty in specifying all the different access privilege requirements. The typical users may not even be aware of specific privileges needed as this requires an understanding of configuration and privilege requirements for different applications/packages.

For instance, in the University scenario, the academic staff members are typically logged on with administrator privileges on their office machine. The academics use the software tools and resources unaware of the permissions needed in completing the lab work or an idea of the privilege requirements for such work in a laboratory environment. Therefore, such access privilege requirements are typically not clearly specified to IT administrators.

- *Complexity of today's systems:* With a large variety of software tools and configurations, it is sometimes impossible for IT administrators to keep track of the different access privilege configurations.

For instance, permissions may need to be granted at OS level and also at various application-levels (e.g. database-level permissions and others). It is unreasonable to expect IT administrators to be trained and knowledgeable in all these different software packages and configurations. This is especially true in a university environment where a plethora of applications on multiple platforms are installed for educational purposes.

- *Works fine*: If the security is configured to be more permissible, it is typically not noticed unless audited or investigated usually after a detected attack. The system "'works fine' so why bother?" mentality of IT administrators. On the other hand, if security is configured without adequate permissions, this might interfere with getting work done conveniently or not at all [1, 10]. This situation typically can inundate IT administrators with user inquiries/requests. Thus, the approach to configure more permissive systems is encouraged and practiced.

- *No reward*: Typically setting up, configuring a secure system needs time and effort. However security set up may seem to contribute nothing to output [1]. Typically, a securely configured system doesn't show its value in terms of functionality.

- *Not worth the risk*: Administrators may consider the effort not worth the risk as security breaches are "rare". Typically, after a catastrophic incident is detected, more attention is paid to such procedures. Also, research points out that "individuals are often less than optimal decision makers when it comes to reasoning about risk" [2].

- *IT Administrator plays multiple roles, thus lack of focus on security*: Typically, IT administrators play wide ranging roles in the IT department – including IT support, installation and configuration, procurement and others. In terms of security, the main focus of IT administrators is typically configuring firewalls, updating patches and configuring up-to-date virus protection software which is straight-forward and frequently more attention is given to external threats. Optimal access privilege configuration has less focus and lower priority.

- *Unaware of risks of permissive systems:* Typically, it is internal users with malicious intent

that take advantage of poorly configured excessively permissive systems. Such incidents are usually kept and dealt with confidentially within an organisation. These types of incidents have the potential to cause embarrassment and loss of face to the organisation involved. Therefore, there is a lack of awareness of the importance with respect to poorly configured permissive systems and its implications, when compared to other types of security attacks that are external in nature - such as virus attacks, exploitation of security holes in software and others which attracts a lot of attention.

The reasons outlined above are not limited to the University scenario presented earlier but applicable to many IT environments in business organisations today. For example, in Section 4 of this paper we discuss an extension of the methodology for application to an online collaborative environment (CE). In CE's member entities frequently require access to other member entities data sources (primarily in the form of databases). Likewise, when a new entity joins the digital collaboration they also require access to established member entity data sources. It is not surprising that many IT administrators for various types of digital environments are guilty of configuring excessively permissive systems to users due to some or a combination of reasons outlined above.

In [3], the authors summarise security configurations in organisations and about the use of security tools by IT administrators as follows:
"The problem is that many organizations fail to take the steps that would protect them – not because they do not care about security, but because they lack people with the skills or the time to address the problems. In order to help reduce these burdens, a wide variety of tools are available that can analyze systems and identify potential security weaknesses."[3]. Although research exists on models for security policy verification (such as [4]), extraction of privileges in role-based access control (such as [5]) and other areas, the authors are unaware of any methodology or tool that discusses eliciting and/or configuring optimal privileges for users as discussed in this paper.

This paper proposes a methodology and presents a high-level architecture for a tool addressing the issue of excessive permissively configured systems. The methodology allows users to automatically specify access privileges required by simply using the pre-configured system in a secure environment. The IT administrators are provided with the required access privileges list of different applications/systems used by the test user for evaluation and approval prior to deployment in the real environment. This methodology addresses most of the drawbacks that exist in today's environment which discourages and prevents IT administrators from configuring optimal access privileges to user groups.

2 System Architecture and Methodology

2.1 System Architecture

The high-level system architecture of the tool is shown in figure 1:

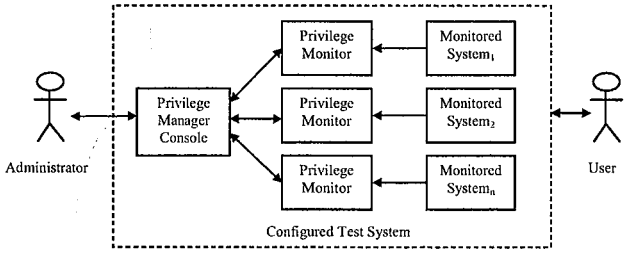

Figure 1. High-level system architecture

The shaded boxes show the components of the proposed system.

- *Privilege Manager Console (PMC)*: PMC is used to set up the Privilege Monitors to the different systems that require setting permissions by the IT administrator. Also PMC displays the privileges the user requires to perform the different tasks in the monitored system. The privileges are reviewed and approved based on IT administrator's discretion.
- *Privilege Monitor (PM)*: PM observes the permissions required by users when performing actions in the monitored system. The privilege list is fed back into the PMC for reviewing by the IT administrator. The monitored system, for example, can be an Operating System or a specific database server that is installed in the test environment.

2.2 Methodology

This section discusses the process of eliciting privilege requirements by user groups for evaluation and approval by IT administrators.

The steps outlined below are followed in determining the required privileges:

Step 1: In the initial step, the system is configured in a test environment. All relevant packages/applications are installed and configured.

In the University scenario discussed initially, this pertains to setting up the parent image for the lab machine with all necessary software packages installed.

Step 2: Next, the Privilege Manager Console and Privilege Monitors are installed and configured for the different systems where the privilege settings are required. Different PMs are typically configured to monitor the OS, database servers and others relevant packages. The PMs will monitor activities performed by users and record the necessary permissions required to perform such activities.

Step 3: A test user from a user group is provided unrestricted access to perform any tasks that he/she might perform using the applications installed in the configured system. Note that this step is performed in a test environment, thus providing unlimited access to the test user doesn't pose a risk to the real environment. The user will perform all the different tasks that he/she would need in a working environment (for example, writing to folders, reading from folders, updating database tables, etc.). The PMs reviews these activities for the pre-configured systems and determines all the privileges that the user requires to perform his/her tasks. An noteworthy point is that all necessary actions required by the user needs to be performed at this stage in order to generate a complete set of user privileges.

In the University scenario, this step is equivalent to academics testing the parent image where they perform the different tasks a student would perform to accomplish his/her laboratory work.

Step 4: In this step, the different Privilege Monitors provide the privilege requirements for the monitored systems to the Privilege Manager Console. PMC presents the privilege list in a user-friendly, flexible manner to be reviewed by the IT administrator. The IT administrator verifies for any vulnerabilities or excessive permissions. At this stage the IT administrator may add or remove additional privileges and even seek clarifications from the test user. Finally, with the IT administrator's approval, the privileges are configured on the test system for deployment or scripts developed to configure permissions in the real environment.

In the University scenario, the parent image is configured with appropriate privileges to the different user groups before it is deployed across to the lab machines.

Note that although this methodology was discussed in the context of a single computer image in the university scenario, the approach is scalable for multiple applications in a distributed environment.

2.3 Advantages of approach

There are a number of advantages of the proposed approach:

- From a users' perspective, the entire process of users needing to specify access privilege requirements is eliminated. The access privileges are automatically gathered by the Privilege Monitors while users perform tasks in a test environment. Also, only the required permissions are enumerated for the tasks avoiding overly permissive privileges being specified.

- From an IT administrators' point of view, the tools (PMC and PMs) eliminate many obstacles in existing approaches:
  - *In depth-knowledge of security configurations*: The IT Administrators do not need to learn the in-depth knowledge of different security configurations for each application and system. The Privilege Monitor does this. This saves the IT Administrator significant effort and time.
  - *Avoid risks and non-optimal security configurations*: Because the Privilege Monitors generate privilege lists based on users actions, it does not allow additional privileges to be enumerated than is needed by the user to perform his/her tasks. Thus the possibility for excessively permissive privileges being granted is reduced avoiding vulnerabilities creeping in.
  - *Flexibly and conveniently evaluate security privileges*: The IT Administrator is provided with a list of security privileges which he/she can evaluate in a user-friendly and secure environment.
  - *Flexible application of security privileges*: The IT Administrator evaluates and decides the required privileges and applies them in an efficient manner (i.e. to the parent image or by generating scripts).

Overall, substantial time and effort required for specifying and determining privileges is saved by both IT users and IT administrators.

3 Implementation

This section discusses a high-level implementation of a PMC and PM where the monitored system is Microsoft's SQL Server database server [6]. The prototype PM will be implemented as a separate

module without the need to modify SQL Server or its implementation.

SQL Server is a client-server relational database engine. All operations to SQL Server are passed as SQL statements. The prototype implementation uses SQL Server Profiler [7], a tool provided with SQL Server, which is configured to monitor SQL commands passed to the server. The Privilege Monitor for SQL Server is able to decipher the trace files generated by SQL Server Profiler and determine the required privileges for user actions.

Figure 2 depicts a high-level flow diagram of such an implementation.

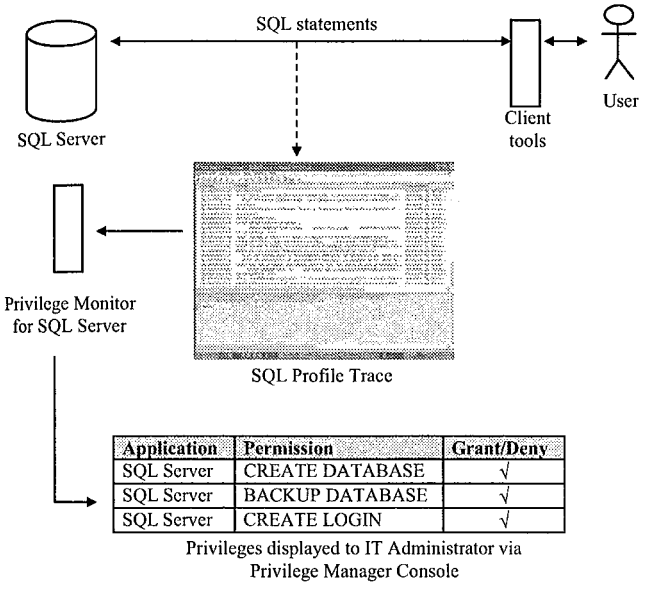

Figure 2. High-level flow diagram of implementation consisting of Privilege Monitor for SQL Server

4 Application to Online Collaborative Environments

The proposed methodology is not limited in its application to homogeneous isolated systems and networks. Rather, the solution can be applied to many different types and configurations of information systems and networks, including online collaborative environments. The remainder of this section details how the methodology and its key components, the Privilege Manager Console (PMC) and the Privilege Manager (PM), can be adapted for use in digital collaborative environments for managing access to shared databases. As is normally the case in Collaborative Environments (CEs) the databases are usually of a disparate nature. This feature serves to highlight the fact that our proposed solutions are technology independent. That is, for example the PM and PMC can be applied equally effective to say Microsoft SQL Server or Oracle Relational Database Management Systems (RDMS).

CEs by their very nature can require a substantial amount of database access control management and administration. As one of CE's main functions is to facilitate the sharing of data, information and knowledge it is important to also ensure proper data security and information privacy measures are in place [8]. Included in the security practices should be effective administration of privileges and permissions for access to member entity databases by other member entities of the collaboration. As different member entities have different access permission requests, that is in terms of what data they may wish to have access to at any given time; it can become very cumbersome to have to manually customise each type of member privilege. To address part of this problem the service providers (member entities managing a database they a willing to share with other members) can apply the methodology we have developed.

Each service provider within the digital collaboration can configure their respective system architectures to integrate the Privilege Manager Console (PMC) and Privilege Monitor (PM) tools into their functional specifications and operation. After which member requests for access to a collaboration service provider, configured with compatible system architecture, can be evaluated and reviewed using our methodology. If the requesting member is from a larger group or organisation within the collaboration, the approved account can: 1) provide a baseline configuration for other members of the same group, hence utilising a form of Role Based Access Control; or 2) be used to delegate authorities to other group members.

When our methodology is used with the principles of delegated authority, then the application of our scheme to distributed CE's compliments the work detailed in [9]. Li and Wang explain their development of a systematic methodology for information sharing in distributed CE's. Where their methodology is based on the use of role-based delegation and revocation, our methodology uses a set of automated tools (PMC and PM) for a test user to establish a 'benchmark' set of access permissions and privileges. Once the test user actions are reviewed by the PM's in step 3 and 4 of our methodology and approved, the requested and reviewed set of permissions and privileges can be

activated in a live distributed collaborative environment. Further requests from other member entities that are also in the same group of the tested members can be granted delegated authority from that member entity. The additional benefit of this method of application is that when needed the initial account privileges can be revoked, along with accounts having delegated authorities.

5 Conclusion

Determining optimal permission requirements for different user groups in today's complex IT systems is a difficult task requiring time, effort and expertise by IT administrators. Therefore, in practice, excessive permissively systems are configured. This results in security vulnerabilities providing opportunities for users with malicious intent.

In this paper, we have outlined a methodology and discussed a high-level architecture for a system that captures privilege requirements of users in complex IT configurations. The system requires the development of a Privilege Monitor Console (PMC) and Privilege Monitors (PMs) for the configurable systems. The methodology allows eliciting optimal privilege requirements in a user-friendly, secure environment to be evaluated and deployed by IT administrators. The methodology is applicable to many different IT environments. The paper illustrates its applicability in a typical IT organisational environment as well as online collaborative environments.

A high-level implementation for the PMC and a sample PM for Microsoft's SQL Server database is discussed. Future research work include: (i.) implementation of Privilege Monitors for different types of systems (such as OSs); (ii.) analysis of empirical results of such implementations; and (iii.) considering applicability of methodology for different IT environments.

References:

[1] B. W. Lampson, "Computer Security in the Real World," *IEEE Computer*, vol. 37, pp. 37-46, 2004.

[2] B. Schneier, "The psychology of security," *Commun. ACM*, vol. 50, pp. 128, 2007.

[3] S. Furnell and S. Bolakis, "Helping us to help ourselves: Assessing administrators' use of security analysis tools," *Network Security*, vol. 2004, pp. 7-12, 2004.

[4] H. Sakaki, Y. Kazou and R. Ogawa, "A Model-Based Method for Security Configuration Verification," in *Advances in Information and Computer Security*, vol. 4266/2006: Springer Berlin / Heidelberg, 2006, pp. 60-75.

[5] S. L. Osborn, L. K. Reid and G. J. Wesson, "On the interaction between role-based access control and relational databases " in *Proceedings of the Tenth Annual IFIP TC11/ WG11.3 International Conference on Database Security: Volume X: Status and Prospects*, 1997.

[6] Microsoft Corporation's SQL Server, "SQL Server Home Page", *Microsoft Corporation*. [Online] Available: http://www.microsoft.com/sqlserver. [Accessed: Aug 30, 2008].

[7] Microsoft Corporation SQL Server Profiler, "SQL Server Profile ", *Microsoft Corporation*. [Online] Available: http://msdn.microsoft.com/en-us/library/ms173757.aspx. [Accessed: Aug 30, 2008].

[8] G. Skinner, "A privacy augmented collaborative environment (PACE)," in Proceedings of the 7th Conference on 7th WSEAS International Conference on Applied Computer Science, Venice, Italy, 2007, Vol. 7, pp. 360-365.

[9] M. Li and H. Wang, "Protecting information sharing in distributed collaborative environment," in proceedings of 10th Asia-Pacific Web Conference Workshop, 26-28 April 2008, Shenyang, China.

[10] L. Bauer, S. Garriss, and M. K. Reiter, "Detecting and Resolving Policy Misconfigurations in Access-Control Systems," in Proceedings of the 13th ACM Symposium on Access Control Models and Technologies. Estes Park, CO, USA: ACM, 2008, pp. 185-194