

White Paper

PCI compliance: Privileged User Management and Monitoring on UNIX and Linux systems

Abstract:

The typical, mid to large scale, contemporary organization is subject to pressures from legislative compliance, security concerns, and the search for best practice to find a way to manage and monitor superuser and other privileged accounts. The Payment Card Industry Data Security Standards (PCI DSS) put the onus on organizations handling credit cards to protect access to cardholder data. It is difficult to be PCI compliant, if privileged users can gain unrestricted access to that data.

This white paper discusses the way that Super User Privilege Management (“SUPM”) software such as Privileged User Manager® (“PUM”) can provide a cost-effective way to manage and monitor those privileged accounts.

Contents:

Executive Summary 3
The 'root' account..... 4
Standard operating system facilities 5
Commercially available software..... 6
Summary 8
About the author..... 9

The Information in this document is subject to change without notice. This document is provided for informational purposes only and Applecross Technologies makes no warranties, either express or implied, in this document. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Applecross Technologies.

All brands or product names are trademarks of their respective holders.

Executive Summary

The Payment Card Industry Data Security Standards (PCI DSS) specify a set of requirements for any organization that processes, stores or transmits credit card data. Such an organization must be PCI DSS compliant or they risk losing the ability to process credit card payments. Merchants and service providers must validate compliance with an audit by a PCI DSS Qualified Security Assessor (QSA) Company.

The PCI DSS were developed by the founding payment brands of the PCI Security Standards Council, including American Express, Discover Financial Services, JCB, MasterCard Worldwide and Visa International. The purpose was to help facilitate the broad adoption of consistent data security measures on a global basis to help prevent credit card fraud, hacking and various other security issues. The PCI DSS is a multifaceted security standard that includes requirements for security management, policies, procedures, network architecture, software design and other critical protective measures. This comprehensive standard is intended to help organizations proactively protect customer account data.

The core tenets of the PCI DSS are the following requirements:

- **Build and Maintain a Secure Network**
 1. Install and maintain a firewall configuration to protect cardholder data
 2. Do not use vendor-supplied defaults for system passwords and other security parameters
- **Protect Cardholder Data**
 3. Protect stored cardholder data
 4. Encrypt transmission of cardholder data across open, public networks
- **Maintain a Vulnerability Management Program**
 5. Use and regularly update anti-virus software
 6. Develop and maintain secure systems and applications
- **Implement Strong Access Control Measures**
 7. Restrict access to cardholder data by business need-to-know
 8. Assign a unique ID to each person with computer access
 9. Restrict physical access to cardholder data
- **Regularly Monitor and Test Networks**
 10. Track and monitor all access to network resources and cardholder data
 11. Regularly test security systems and processes
- **Maintain an Information Security Policy**
 12. Maintain a policy that addresses information security

The use of UNIX systems to host databases and financial applications, and the rapidly increasing adoption of Linux systems for mission critical applications, both present a PCI-compliance problem with relation to the administration or 'root' account. IT operations and administration personnel must use the 'root' account to perform many critical tasks on these operating systems. Unfortunately, the 'root' account allows

effectively unlimited and unrestricted access to the system(s) and its contents, including audit trails and data.

In short, it is difficult to be PCI-compliant unless this access is managed and monitored. It is difficult to see how an organization can protect their data if administrators of the systems holding those records have unrestricted access. In particular, organizations may fall foul of the following requirements as listed above:

3. Protect stored cardholder data
- 7: Restrict access to cardholder data by business need-to-know
8. Assign a unique ID to each person with computer access
10. Track and monitor all access to network resources and cardholder data

This paper investigates this problem, and demonstrates how one low cost commercial offering can provide a solution.

“68% of employees bypass their employer’s information security controls in order to do their jobs – IT Governance.”

The 'root' account

The 'root' account is the most privileged account on a Unix or Linux system. This account grants the 'superuser' the ability to carry out all facets of system administration, including the addition of user accounts, the changing of user passwords, examining log files, installing software, etc. The 'root' account has virtually unlimited access to all programs, files, and resources on a system.

The 'root' account is the special user in the '/etc/passwd' file with the user ID (UID) of 0. It is not the user name that makes the 'root' account so special, but the UID value of 0. This means that any user who has a UID of 0 also has the same privileges as the 'root' user.

When using this account, administrators are taught to be as careful as possible, as the account has no security restrictions imposed upon it. This means that a skilled administrator can perform administrative duties without inconvenience, but it also assumes that they know what they are doing, as the underlying system will do exactly what is requested - no questions asked, and no easily read audit trail.

Notwithstanding the concerns about internal users, the protection of the 'root' password is an obvious necessity, as access to this password would provide similarly unrestricted access to any intruder. A common attack method of potential hackers is to obtain the 'root' password. Unfortunately, it is often necessary to grant access as 'root' to external contractors and suppliers if they are to fulfil their duties.

“The goal of a hacker is usually to obtain the super user status ('root')”

Although many users may require only occasional access to a sub-set of the commands that require 'root' access, the lack of granularity in the administration accounts available means that all such users are typically granted 'super user' access, with all its ramifications. They not only obtain the permissions that they require to carry out their job function, but are also granted unnecessary additional access.

Unless privileged user management software is used, system administrators should only operate as the 'root' user to perform system administration functions that require root privileges. For all other operations, they should return to their normal user account. Routinely operating as the 'root' user can result in damage to the system because the 'root' account overrides many safeguards.

The same problem arises with other privileged accounts of course, 'administrator' on Microsoft Windows systems, and those required to manage relational database management systems etc. In all such cases it can be seen that such unrestricted and unaudited access can be detrimental to the search for PCI-compliance.

A solution is required to control 'root' access, monitor and manage its use, and audit all activity when it is used.

Standard operating system facilities

The problem of securing the 'root' account is not a new one, of course. It is therefore surprising that the various flavours of UNIX and Linux offer little by way of control and auditing in this area. The situation remains one whereby you can do everything if you have 'root' access, but only act on your own account if you do not. There is no granularity of access control within those two extremes. Anybody gaining access to the 'root' password may therefore have access to critical data, so rendering the system potentially non-compliant through a lack of access control and assured integrity of data.

The requirement for auditing is also not properly addressed. UNIX and Linux variants maintain a number of log files that keep track of what's been happening to the computer. Early versions of UNIX used the log files to record who logged in, who logged out, and what they did. Newer versions of UNIX and Linux provide expanded logging facilities that record such information as files that are transferred over the network, attempts by users to become the superuser, electronic mail, and much more. Log files are an important building block of a secure system: they form a recorded history, or audit trail, of the computer's past, making it easier to track down intermittent problems or attacks. Log files also have a fundamental vulnerability. Because they are often recorded on the system itself, they are subject to alteration or deletion. And the 'root' user is able to edit the log files at any time. The requirement for due diligence under PCI DSS, only provable by auditing, is therefore absent. The sheer volume of information logged also makes it difficult to extract meaningful data.

Commercially available software

There are a small number of quality software packages available from commercial vendors which tackle the problem of managing 'root' and other privileged users in a manner which can provide a greater level of security, control, and compliance with the PCI DSS and other prevailing legislation. This paper will discuss only one such product - Privileged User Manager ® ("PUM") from Applecross Technologies, the author of this white paper.

PUM provides comprehensive functionality, scalability for larger users, a higher level of security, granular control, an easy to use web browser interface, comprehensive and indelible auditing, and most importantly - commercial-strength support and maintenance; all for a small annual fee. It offers a proven solution that is worth evaluating when the penalties for non-compliance are taken into account.

PUM allows an organization to delegate the privileged access of UNIX and Linux systems to staff, suppliers or contractors without having to disclose the 'root' password. Without the 'root' password the users cannot log into the managed systems to access the 'root' or 'superuser' account. No login, no risk. However, some users will need to perform tasks on those systems which require that enhanced level of privilege. The goal is to provide the privilege to the task, not to the user.

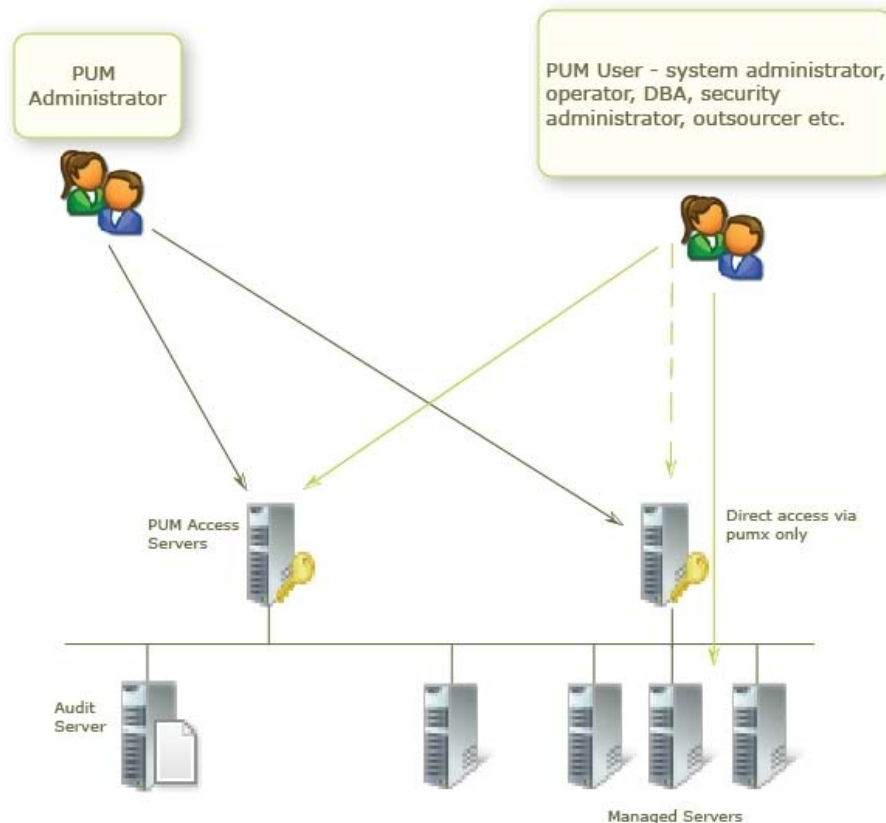


Figure 1 – PUM physical architecture

In order to achieve the goal of controlled audited access, PUM provides for a facility whereby all privileged access to computer systems is either routed through, or validated by, a PUM Access Server. These Access Server(s), normally provided in the form of a virtual appliance, may be dedicated to the task in larger organizations, or perform a shared purpose in smaller sites. By centralizing all access and/or validation, access policy is easier to apply and change, and security loopholes are easier to close.

PUM uses SSH-2 (Secure Shell) as a communications method between the Client and Access Servers, and between the Access Servers and Managed Servers, so that the need to install further agents on the managed systems is avoided. It is effectively agentless as SSH is installed by default on most modern variants of UNIX and Linux.

Administrative users who require privileged access to one or more servers must be preauthorized for such a session. Provided that the user is preauthorized then the administrative user requiring a privileged access session will either:

- a) log into a PUM Access Server which will then automatically log the user into the Managed Server without their knowing the password, or
- b) log into a Managed Server directly using their own user account, and then perform privileged tasks by preceding the commands with the 'pumx' prefix.

In either case, anything that the user tries to do is validated against the Access Server(s). The user may only carry out a pre-authorized subset of privileged commands on a subset of managed systems, at certain times. It effectively allows for a policy of "superuser access only how, where and when access is required". All resultant activity is audited in an indelible manner. Of course, nothing prevents the organization from allowing fully trusted individuals full, unrestricted access if this is required. All such activity would still be audited.

PCI-compliance is therefore improved by creating a situation whereby only those personnel who should have 'superuser' access are able to gain that access, and only then in a potentially limited form on systems to which they need to have that access, and at times when that access is allowed. Although users gain limited superuser access, they never become a 'superuser' or 'root' user in the traditional sense, as they are never granted the 'root' password or unrestricted access. They never login to the system they are managing as the 'root' or 'superuser'. Importantly, the use of an indelible audit trail allows an organization to "prove" the controls that are in place.

All activity is directed to one or more Audit Servers which normally resides on a dedicated system to which nobody other than compliance officers, data auditors or security managers have access. It cannot, therefore, be altered by anybody else and retains an indelible and complete record of all administrative commands, and the data returned for those commands. It becomes possible to know at any time, who was granted access to what by whom and when, and any session may be played back at some future time by those with the appropriate authority.

PUM can be used with equal facility to control privileged access to any application that has a command line interface for special administrative commands.

Access Policy

The Access Policy is a collection of related objects that determines how a User gains access to a privileged account.

- Sessions**: Define which PUM Users can run which commands on which Managed Servers and when.
- Servers**: Host systems that are managed or utilized by PUM.
- Roles**: A job function or responsibility.
- Schedules**: A set of Calendars that collectively define when a Session may be run.
- Shell**: A restricted shell specifying the commands the user can run during a privileged session.
- Users**: A person who needs to have access to PUM as a PUM User or PUM Administrator.
- Calendars**: A set of date and time constraints.

Click on any entity above to view those objects.

© Applecross Technologies Pty Ltd. 2005-2008

Figure 2 – PUM Access Policy screenshot

For a full description of the architecture please visit <http://www.applecrosstech.com>

PUM is available by subscription licensing only, at a cost of US\$365 per year per server. This price includes support and is therefore an extremely cost-effective option. Site licenses are available for larger users.

Summary

The use of an unrestricted and unmonitored 'root' account to manage UNIX and Linux systems that hold protected information on credit card users puts an organization into a situation of potential non-compliance with the Payment Card Industry Data Security Standards (PCI DSS). Organizations who have to maintain compliance should seek out a solution that allows them to control who has access to the 'root' account, what they can do when they have the 'root' account, prevent users from deviating from those controls, and audit all activity.

The standard UNIX and Linux operating systems provided by the major and minor vendors offer little by way of tools to help in this regard.

Commercial software such as Privileged User Manager ("PUM") from Applecross Technologies can provide a comprehensive solution at minimal cost and is highly recommended for larger users.

About the author

Applecross Technologies Pty Ltd. (www.applecrosstech.com) is dedicated to the provision of low cost, high quality systems and security management software for UNIX and Linux systems.

It sells and supports Privileged User Manager® (PUM) directly and through its strategic partner, Open Systems Management Inc. (www.osminc.com) through offices in the USA, Europe and Australia.

Contact us at info@applecrosstech.com



Applecross Technologies Pty Ltd.
P.O. Box 1562
Applecross
Western Australia 6153
89 North Lake Road
Myaree
Western Australia 6154

Tel. +61 (0)8 9317 6855
Fax. +61 (0)8 9317 6866

North America:

Open Systems Management Inc.
1511 Third Avenue
Suite 905
Seattle, WA 98101
USA

Tel. (sales) +1 206 583 8373
Tel. (support) +1 888 OSM TECH
Fax. +1 206 583 8374

Email: info@osminc.com

Europe:

Open Systems Management Ltd.
9 Millars Brook
Molly Millars Lane
Wokingham
Berks. RG41 2AD
United Kingdom

Tel. (sales) +44 (0)1189 070330
Tel. (support) +44 (0)1189 070338
Fax. +44 (0)1189 070341

Email: info@osm.co.uk