

Privileged Password Management

Best Practices Guide
December, 2008

Overview

In datacenters worldwide and probably in your organization too, it is common practice to hard-code passwords and user IDs in applications and scripts. Auditors and IT groups knowingly allow application-to-application (A2A) passwords and user IDs to remain shared among administrators, developers and contractors.

This practice is starting to change. Leading IT organizations are now recognizing and resolving the risks of unmanaged and exposed passwords lurking in their datacenters. The increasing frequency and growing impact of insider attacks, as well as more demanding regulatory compliance requirements, means you can no longer ignore this known risk. You must address the threat hiding in plain sight.

Hard-coded passwords also cost time and money. A simple password change requires you to update and redeploy applications, which may cause synchronization problems and server outages. Multiply these issues across hundreds or thousands of servers and applications, and you may have tens or even hundreds of thousands of unmanaged passwords which create huge costs and risks.

This paper reviews the security risks associated with hard-coded passwords and helps you to:

- > Gain insight into the security vulnerability that lies on every server
- > Learn why IT organizations struggle with access controls in the datacenter
- > Master the security challenges beyond access controls
- > Learn how to secure the datacenter through application password management
- > Discover solutions for secure centralized password management for application servers

The security vulnerability on every server

You've probably invested a great deal of time, money, and effort in deploying network perimeter defenses and User Identity Management policies and solutions. The goal has been to ensure the security of corporate resources. However, these essential expenditures do not address threats from internal sources. Over half of the list of published US Department of Justice cases on computer fraud have been perpetrated by disgruntled or former employees.¹

¹ www.cybercrime.gov

The pressing need to address User Identity Management has deflected attention from another use of user IDs and passwords, which is the practice of hard-coding them into applications so that an application-to-application or application-to-database connection can be established.

Unlike a human, an application lacks the ability to enter a password through a keyboard or authenticate using a second factor token. Applications must therefore authenticate using a stored password. Typically, these passwords are hard-coded into the application or script, or are stored in a configuration file.

Research shows that fully 90% of application authentication in a datacenter remains password-based. Considering that these hard-coded passwords are “in the clear”, are known by many, and are rarely changed, organizations should be concerned about the risks of this practice.

When assessing the effort required to reduce the risks associated with hard-coded passwords, ask the following key questions:

- > How many server scripts and applications hosted in your enterprise datacenters use hard-coded user IDs and passwords to access other server applications?
- > Does your organization require that the same security practices be applied to the passwords hard-coded within applications, as must be applied to users' passwords?
- > When was the last time that all of your application-to-application passwords were changed?
- > How many developers know the passwords to your database and application servers?
- > Are all passwords changed when a developer leaves the organization, or after a contractor leaves at the conclusion of a project?

The threat hiding in plain sight: hard-coded application passwords

The most common reason that application-to-application passwords are not being changed regularly and often is, quite simply, cost. The human cost to maintain and redeploy the hundreds or thousands of applications that contain hard-coded passwords grows unsustainably as the number of passwords and applications grows. Furthermore, the cost of server or application outages caused by unsynchronized or incorrectly changed passwords can exceed the cost of changing the passwords in the first place.

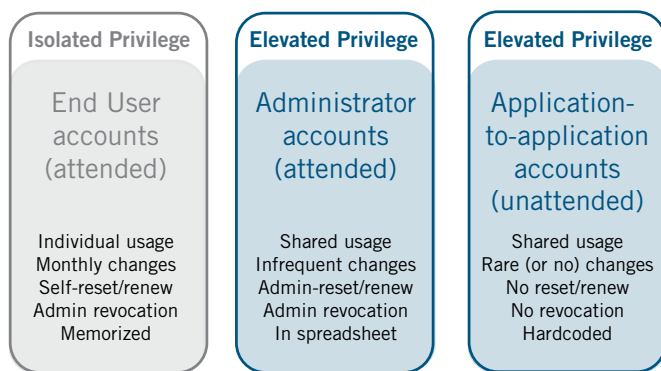
Eliminating hard-coded application passwords may seem like a simple problem to solve, but it raises many new security chal-

lenges when you consider the potential insider threats that could compromise your systems.

The next sections describe the security building blocks and approach needed to remediate the issue of Application Password Management while increasing operational efficiencies and up-time, reducing the risk of credential breaches, and tightening audit compliance and control.

A short primer on account types

The figure below shows the three account types found on a network and the characteristics of user IDs and passwords in each. Malicious insiders target elevated privileged accounts—which include unattended application-to-application accounts—because these accounts give greater access to systems and data.



Legislation pressure on CIOs and auditors

Industry and government legislation such as Payment Card Industry, FISMA, Sarbanes-Oxley, Senate Bill 1386, and others require changes in how organizations run. Auditors are interpreting the applicable legislation to establish the policies and practices to which their organizations must adhere. Cost has not been an accepted reason for failing to comply with a particular piece of legislation. Your CIO knows that your application-to-application passwords are not being secured or changed, are known to your developers and contractors, and are visible in plain text inside of scripts, applications, and configuration files across all of the servers within your datacenter. Your CIO is responsible for the Internal Controls for Financial Reporting as they relate to the Sarbanes-Oxley Act of 2002, and is required to sign off on Section 404. Knowing all of this, how can he or she sign off?

The auditor interprets all applicable legislation and compares the intent of those documents with the organization’s policies and procedures. Most organizations are tired of having the leg-

islative hammer poised over their heads; however, the daunting prospect of having to publicly disclose any lapses in security or of being tried in a court of law, and the risk of losing revenue and customers have driven security practices to the top of the IT spending list.

While most forms of legislation provide general guidance, few—if any—explicitly state the mechanisms to achieve compliance or conformance. It is hard for an auditor to review and interpret the specific details of each of the legislative documents that may affect your business. Establishing the best practices that cost-effectively allow you to gain compliance is even harder. It is important for the audit community to raise the awareness of this threat hiding in plain sight. Hard-coded application passwords is a legacy problem, yet developers continue to hard-code application passwords into new applications. Why?

A big reason is that cost-effective, efficient, secure, commercially-supported solutions to the application-to-application password management challenge did not exist until recently.

Struggling with access controls in the data-center

Today, most applications rely on the “trusted network” of your organization to control who or what has access to maintain or execute the applications resident on your servers. The trusted network is the internal network of your organization that employees and contractors authenticate to in order to complete the tasks associated with their role. Few organizations have protected their internal networks beyond using operating system file access controls.

When you consider that an internal threat comes from a person who has, or had access to your internal network, you begin to realize that they also have had the time to plan an attack, understand the value of their objective, know the systemic defenses and reporting mechanisms, and enjoy a presumption of innocence because they are an “insider”.

It has been shown that internal attacks, while fewer in number, are far more financially damaging than external attacks. Of the listed Department of Justice computer fraud cases that were perpetrated by insiders, most included the exploitation of weak, unchanging passwords on servers to which the insiders had some level of access—because they were members of the trusted network.

Even Public Key Infrastructure (PKI) systems are challenged by unattended applications. PKI systems must protect the private keys used to authenticate, authorize, and digitally sign. But how does an unattended application protect its private keys

stored on disk? With a password! And this password is typically embedded into the application or script, or is stored in a configuration file—completely defeating the purpose of using a PKI for strong authentication. In this case, security becomes a “chicken and egg” issue: How are private keys protected while in use in memory by an unattended application? The answer is they aren't. While PKI is an elegant solution for strong authentication, digital signatures, and non-repudiation, it suffers many challenges in an unattended environment that is subject to internal attack.

To effectively solve the password management challenge, first eliminate the passwords from the scripts and applications that use them. To do this, establish a central location from which scripts and applications are able to retrieve passwords when needed. An important benefit of using a central password repository is it provides a single point of control over the release policies for passwords. This was not possible before. Clearly, strong security techniques are needed to protect the passwords stored in the central repository. Data encryption is not enough. An attacker will attempt to monitor server memory or breach the software libraries that contain or utilize the keying material, to decrypt the data in the repository. Techniques to hide keys and algorithms are essential to a secure password management solution.

It's also vital to secure the end points of connections to the central repository. As these end points are expected to operate unattended, it is not enough to rely on physical security alone. The end points must be capable of protecting their identities, protecting the keying materials used during cryptographic operations, and detecting attempts to tamper with scripts and applications that execute upon them.

Security challenges beyond access controls

For a system to have the confidence needed to release a critical credential (such as a password) to an unattended application, and resistance to both external and internal threats, it must be capable of application self-authentication and systemic self-protection. Just like the human biometric which uniquely identifies a person, there are many runtime environmental details that can be collected during application execution.

Combining these application “biometrics” with cryptographic techniques delivers a means to authenticate and authorize the release of critical credentials to uniquely identifiable and registered applications. This “biometric” comparison of an application against the authenticated application's profile can be used to also ensure that the calling application has not been altered.

This validation ensures that credentials are not disclosed inappropriately.

Security requirements for password management

By combining security techniques with best security practices, it is possible to outline the specific security requirements of a centralized password management system for unattended servers and applications. They are:

- > Central server authentication
- > Client/agent authentication
- > Protected central repository
- > Session and message-level encryption
- > Tamper-resistant libraries and applications
- > Server scope control
- > Secure local caching
- > Protected keying materials

Building blocks of an effective password management system

The next sections describe the security techniques you should apply together to achieve the security requirements listed above.

INTEGRITY VERIFICATION

Through integrity verification, the centralized password manager determines that the calling application, as well as the password management system, remain as originally developed and deployed. Verification techniques check components statically on-disk and dynamically in-memory. Calling applications must prove their integrity before the centralized password manager releases a credential.

FINGERPRINTING

A server's fingerprint is a unique “biometric” element produced from a combination of hardware characteristics like CPU serial numbers, network IDs and other items. By dynamically calculating the fingerprint of the computer executing a script or application, a centralized password management system can validate the physical machine identity of the credential requestor. By registering all requestors to the system, the fingerprint becomes a critical factor in controlling the scope of authenticating servers.

VALIDATED CRYPTOGRAPHY

Passwords and encryption keys must be protected from unauthorized disclosure, and validated cryptographic modules ensure that this is done securely. Any weaknesses in the means used to protect these critical credentials potentially exposes the entire enterprise to attack right where it is most vulnerable. By employing validated encryption mechanisms, these critical components of an agency's information security architecture are provided with assured protection from any possible unauthorized disclosure.

TRANSFORMATIONS

Code transformations are mathematical transforms that are applied to data flow and control flow within a program to hide the original information and algorithms. The technique prevents reverse-engineering and creates interdependencies and complexity that prevent tampering. Impact on performance and code expansion is acceptable.

RENEWABILITY

This security technique contributes to the overall effectiveness and security of the password management solution by limiting the lifetime of critical elements of the system. Limiting the lifetime of these elements shortens the amount of time that an attacker has to successfully breach the component before it is replaced.

Automated renewability is applied to:

- > **Passwords.** Renewing passwords frequently is a significant step toward enhanced data protection. For years, organizations have pressured users to renew their passwords while rarely changing hard-coded passwords in scripts and applications. Improving security requires a centralized password manager that automates the renewal and retrieval of long, strong and random passwords for datacenter applications.
- > **Repository Encryption Keys.** In addition to renewing passwords, it is important to renew key materials used to protect those passwords while they are statically stored in a repository. A centralized password manager must allow customer-controlled regular and ad-hoc repository key renewals.
- > **Session Authentication Keys.** To gain access to the repository, it is important to control and renew the authentication keys used by the connecting client agents. A centralized password manager must allow customer-controlled regular and ad-hoc renewals of SSL client private keys.
- > **Message Encryption Keys.** Building on repository key renewals, it also is important to renew individual keys used to transmit information securely between the centralized password manager and its connecting agents.
- > **Agent Software Renewal.** Each of the thousands of application servers in a datacenter will run a copy of the software, which validates the integrity of the requesting application. Updating each of these servers to distribute maintenance patches, updates and upgrades is an essential maintenance function of any production environment. Any system to manage A2A passwords must include a software renewal mechanism that allows customer-controlled, automated (or manual) patching of the agent software.
- > **Agent Secure Cache Renewal.** Maintaining a secure local cache of retrieved passwords greatly improves performance and contributes to a high-availability design. However, as passwords are changed, the cached information must be renewed automatically.

The combination of the above security techniques delivers a comprehensive approach to the secure management and release of the elevated privilege account passwords that protect access to your most critical data. Managing these passwords proactively gives a reasonable assurance that you can promptly detect and prevent any unauthorized acquisition, use or disposition of sensitive data in your datacenter.

Summary

In most organizations, the presence of unmanaged and exposed passwords and the resulting insider threat has not yet been a security focal point. But increasing instances of insider attacks, the higher impact of such attacks, and increasing compliance requirements are forcing organizations to address the issue.

Traditional security approaches to these insider threats have proven to be inadequate and ineffective. Only through secure centralized password management is an organization able to effectively address these threats and deliver on compliance requirements, while permitting insiders to remain productive. Centralized password management requires a combination of both centralized access control and robust technology for application-level security.

The risk of a credential breach is dramatically reduced when scripts and applications can gain run-time access to the validated accounts and passwords that they need to execute without divulging those accounts and passwords to developers, or hard-coding the passwords in scripts and applications. The approach

ABOUT CLOAKWARE

Cloakware, an Irdeto company and part of the Naspers group, provides innovative, secure, proven software technology solutions that enable customers to protect business and digital assets in enterprise, consumer and government markets. Cloakware's two main product lines include; Cloakware Datacenter Solutions which help organizations meet governance, risk management and compliance (GRC) objectives for privileged password management while ensuring business continuity and the security of mission-critical data and IT infrastructure. Cloakware Consumer Product Solutions protect software and content on PCs, set-top boxes, mobile phones and media players. Protecting over one billion deployed applications, Cloakware is the security cornerstone of many of the world's largest, most recognizable and technologically advanced companies. Headquartered in Vienna, VA and Ottawa, Canada, Cloakware has regional sales offices worldwide.

also eliminates costly application maintenance and outages when passwords need to be changed. Furthermore, the approach reduces the cost of changing passwords to the point that frequent and regular password changes are easily affordable, which in turn greatly reduces risk. The secure local caching of application user IDs and passwords achieves execution performance which is nearly equal to that of hard-coded application passwords. Automated cache expiry enables frequent password changes and eliminates the problem of stale cached passwords.

CONTACT INFORMATION

Corporate Headquarters

Cloakware Inc.
8219 Leesburg Pike, Suite 350
Vienna, VA, USA 22182
Tel. +1.703.752.4830

Canada

Cloakware Corporation
84 Hines Road, Suite 300
Ottawa, ON, Canada
K2K 3G3
Tel. +1.613.271.9446

www.cloakware.com

