An Enhanced Approach to using Virtual Directories for Protecting Sensitive Information

William Claycomb[1] and Dongwan Shin[2]

[1] Sandia National Laboratories, Albuquerque, NM, U.S.A.

[2] Secure Computing Laboratory, New Mexico Tech, Socorro, NM, U.S.A.

Abstract. Enterprise directory services are commonly used in enterprise systems to store object information relating to employees, computers, contacts, etc. These stores can act as information providers or sources for authentication and access control decisions, and could potentially contain sensitive information. An insider attack, particularly if carried out using administrative privileges, could compromise large amounts of directory information. We present a solution for protecting directory services information from insider attacks using existing key management infrastructure and a new component called a Personal Virtual Directory Service. We show how impact to existing users, client applications, and directory services are minimized, and how we prevent insider attacks from revealing protected data. Additionally, our solution is supported by implementation results showing the impact to client performance and directory storage capacity.

1 Introduction

Enterprise directory services (EDS) are commonly used in enterprise systems to store information pertaining various directory objects, such as users, computers, or contacts. EDS are used to share information with others, such as address books, or as authoritative sources for authentication and access control. In most cases, this dual role is combined in the same directory service instance.

Generally, organizations seek to establish interoperability and seamless communication between heterogeneous systems in enterprise systems, and directory services enable them to do so. Using standard communication protocols, such as Lightweight Directory Access Protocol (LDAP), EDS may provide authentication services or information to multiple requestors, independent of platform or implementation. This is particularly advantageous to organizations seeking to consolidate multiple authoritative information sources into a single repository.

However, this move towards centralized information services has serious drawbacks. In particular, as the amount of information stored increases, the potential for storing sensitive information increases. This is especially true in instances where certain pieces of information are not necessarily sensitive when stored separately, but become sensitive when combined. Consider attributes such as department number and security level. Knowing which people are in a certain department is not necessarily sensitive.

Shin D. and Claycomb W. (2009).
An Enhanced Approach to using Virtual Directories for Protecting Sensitive Information.
In *Proceedings of the 7th International Workshop on Security in Information Systems*, pages 36-45
DOI: 10.5220/0002202800360045

Likewise, knowing which people hold a certain security level is not necessarily sensitive, particularly if many people hold the same level. However, if combined, then a very specific set of people can be identified - those in a particular department holding a higher security level. That information could be used to specifically target those individuals for context-aware, or spear phishing [1] attacks, where individuals are targeted and the attack appears to come from a legitimate sender, such as a colleague.

Another potential hazard when consolidating multiple directory services into a single EDS is the inclusion of certain information not meant to be shared among larger sets of users. Such tightly controlled information could include attributes considered to be personally identifiable information (PII) such as identification numbers, or other confidential information such as bank account numbers. In these cases, the intended users are a small subset, such as the human resources or payroll offices only.

Protecting this information is critical, and most directory services solutions provide methods for limiting access. However, such measures can usually be circumvented by system administrators, or those with elevated privileges. These users may obtain access to sensitive directory information in more than one way. For instance, they might override existing access control methods, or they may impersonate an authorized user to gain access to the information. Another method would be to simply copy the entire directory information store to attempt extraction of sensitive information.

We propose a method for protecting sensitive directory services information from all users, including system administrators, using encryption. Furthermore, we base our solution on existing infrastructure commonly used in enterprise systems. Our main contributions are the introduction of a new type of *virtual directory* service, called a *personal virtual directory service (PVDS)*, which interfaces with a key management system (KMS) and handles encryption and decryption of sensitive information at the client level. Additionally, we show how our solution's impact to existing directory services is minimal, in terms of directory size and performance. Finally, we demonstrate how our solution mitigates an insider attack, where the attacker uses domain administrator privileges to attack a directory service.

The remainder of this paper is organized as follows. Section 2 presents related work and previous approaches, followed by Section 3, which details our approach. In Section 4, we discuss the advantages of our solution, list various attack models, and show implementation results. Section 5 concludes the paper with suggestions for future work.

2 Background

The threat of unauthorized access of sensitive data by employees or other authorized users, known as "dedicated insiders", is well documented [2–4]. In January 2008, the U.S. Secret Service and CERT issued a report titled "Insider Threat Study: Illicit Cyber Activity in the Government Sector" [2]. This study outlines a multi-year project, started in 2002, that explores the activity and threats posed by insiders. Among the key findings of this study are the following:

- Most of the insiders had authorized access at the time of their malicious activities
- Access control gaps facilitated most of the insider incidents, including:

- The ability of an insider to use technical methods to override access controls without detection
- System vulnerabilities that allowed technical insiders to use their specialized skills to override access controls without detection

2.1 Previous Approaches

The structure of EDS is hierarchical, with each leaf representing an *object*. Objects are described by individual *attributes*, such as name, title, password, etc. Solutions for protecting directory services are generally implementation specific, and rely largely on per-attribute access control lists (ACLs). Other directory services instances generalize this approach by using the concept of *confidential* attributes [5], but the underlying implementation is still ACL-based. The use of encryption in directory services is very limited, with specific implementations employing encryption for every instance of certain attributes across the entire directory [6]. However, this approach uses a single server-based key for encrypting all attributes.

A user-centric approach to protecting directory attributes is described in [7]. This method is not dependent on a particular directory implementation. Rather, it utilizes user-based public/private keys to allow users control of encrypted of attributes related to their own directory information. This solution describes different methods for using public/private keys to ensure either data authenticity alone, or data authenticity combined with confidentiality. Specific solutions are proposed for scalability and usability purposes. However, key control is maintained by the user, which raises issues of maintainability and ease-of-use.

2.2 Virtual Directories

"A *virtual directory* functions as an abstraction layer between applications and data repositories". [8] In contrast to *metadirectories*, virtual directories do not maintain data in a standalone data source. Rather, virtual directories reference various data sources and present a consolidated view to the end user. This has the advantage of not requiring data synchronization - the data presented is always real-time, directly from the source. Most virtual directory implementations have the additional capability of acquiring data from sources other than directories, such as databases, and presenting this information to end users via LDAP [9].

2.3 Recent Works

Virtual directory instances can be highly customized to modify, or *transform*, data prior to client use. Recently, we proposed a solution for protecting information in directory services using virtual directories [10]. This approach uses a three-layered architecture, placing a virtual directory server (VDS) between the client and destination directory server. Using a user-protected key, information is encrypted by the VDS before being stored in a destination directory. Similarly, information is decrypted before being delivered to a requesting LDAP client. We show how to delegate access to other users as well, by including a simple ACL with the protected information. The form of the protected data in the destination LDAP directory is as follows:

$$\{data \| H\{pwd_o\} \| ACL\}_{K_{cv}}$$

Here, $H\{pwd_o\}$ is the hash of the owner's password, ACL is a list of authorized users (identified by their password hashes), and K_{cv} is a shared secret key between a client and the VDS. Clients supply additional information to the VDS using components of the LDAP *bind* operation, in the form of an *authentication string*, which is as follows:

$$ID_c \| \{K_{cv} \| H\{pwd_c\}\}_{K_v}$$

Here, ID_c is the identity of the client in the destination directory, $H\{pwd_c\}$ is the hash of the client's password, and K_v is a secret key managed by the VDS. Note that in this case, all operations on the authentication string, including initialization and modification, are controlled entirely by the VDS.

This solution relies heavily on user-known and user-protected passwords, as well as the secret key K_{cv}, which is known to all authorized clients, to prevent the compromise of information in the directory store. Usability is a major drawback to this scheme, with changes required to the protected data any time the user (or a delegate) changes a password. Similarly, lost or forgotten passwords become a liability, because no "back-door" method exists for retrieving information without an original password. Using the shared key K_{cv} also adds an additional component of risk, because it could potentially be used to compromise information taken directly from the directory store itself. For more detailed implementation information, please see [10].

3 Approach

We propose an approach to protecting sensitive directory services information using encryption which does not rely on user-protected shared keys or passwords. We build on previous work by addressing the significant usability challenges and security concerns. Additionally, we simplify the model by eliminating the middle component, the VDS, and replace it with a novel approach to virtual directory technology, which we call a *personal virtual directory service (PVDS)*.

3.1 Personal Virtual Directory Service

Not only does the VDS component of previous works require additional configuration and administration, but it serves as a target for attacks. If the server hosting the VDS is compromised, then all protected information it processes could be revealed to an attacker. We propose moving the virtual directory concept from a centralized configuration to a more distributed configuration. This is accomplished by running what is essentially a simplified VDS on each client machine - the PVDS.

The purpose of the PVDS is to handle communication between an LDAP client application and the destination directory. It is only used in cases where sensitive information needs to be processed - not all LDAP communications between the client and directory services need to be protected, however. In cases where sensitive information is not processed, standard communication should not be interrupted. This basic architecture is shown in Figure 1.

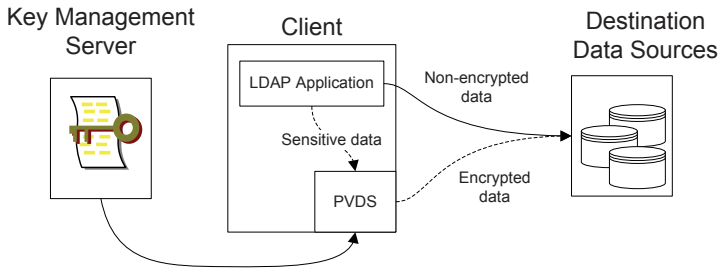

Fig. 1. System architecture for a client system using a personal virtual directory service.

The PVDS is configured at a client level, instead of a centrally managed VDS. This has the potential to increase administrative overhead, as distributed applications tend to do. However, we will show that the function of the PVDS is largely self-contained, requiring no additional configuration once deployed. The necessary configuration to allow protected information to be processed occurs at the client level only, as with previous approaches. Therefore, the overall administrative overhead is actually reduced using our new model by removing the VDS component.

3.2 Using Existing Key Management Infrastructure

Instead of using user-controlled and user-protected keys, our solution makes use of existing key management infrastructure to provide encryption and decryption information to the PVDS. This shifts the burden of key protection from systems not designed to protect sensitive information (directory services, virtual directories, and users) to systems specifically engineered to withstand attacks on keying information. Once a user authenticates to the KMS, the PVDS handles retrieving keying information and using it for data protection, as well as delegation.

3.3 Client Modification

Previous approaches utilize components called authentication strings, controlled by the VDS, which must be changed when any configuration modification occurs. Our approach simplifies this considerably. The only client modifications necessary to enable secure information protection are to replace the destination directory information with the local path to the PVDS, and to concatenate the username with the actual destination directory information. That is, the information contained in the username configuration of the LDAP client would be $ID_c \| dest_{LDAP}$, where ID_c is the username of the client and $dest_{LDAP}$ is the network address and port of the destination LDAP directory instance. No centrally managed authentication string is required.

3.4 Delegating Access

Because we use an existing key management infrastructure, delegation of access is fairly straightforward. Data owners may delegate permission to read and/or write protected data. Granting access, modifying access, and revoking access are controlled via

an interface with the PVDS called a *Delegation Manager*. The delegation manager communicates directly with the KMS to obtain verified delegatee key information. Data protection information is encrypted using verified public keys of delegatees, ensuring that only intended parties have access to encryption and decryption keys.

3.5 Protecting the Data

Sensitive data is encrypted in the directory using a symmetric key S, which is chosen randomly and managed by a component of the PVDS called the *Cipher Manager*. A unique S is used for each attribute encrypted. In contrast to previous approaches, S is never known by the user, nor is it stored by the PVDS on the client machine. It only exists in the destination directory, encrypted by the public key of the owner, K_o, and any delegate users, K_u.

By adding the capability to delegate read/write access, we necessitate additional protection information. A public/private key pair $\{K_{rw}, K'_{rw}\}$ is generated for every protected attribute. K'_{rw} is used by authorized users to digitally sign the encrypted data, $\{data\}_S$. Only users delegated write permission have access to K'_{rw}, which is encrypted (along with S) using the delegatee's public key K_u. K_{rw} is included in the data stored with the attribute, is signed by the data owner, and is used during all read operations to verify the authenticity of $\{data\}_S$. Using the data owner's public key, K_o (available from the KMS), the authenticity of K_{rw} can be verified.

An example of the combined form of all data stored in the directory for a protected attribute is shown as follows:

$$\{\{data\}_S\}_{K'_{rw}} \| \{K_{rw}\}_{K'_o} \| ID_o \| \{S, K'_{rw}\}_{K_o} \| \{S, K'_{rw}\}_{K_{u1}} \| \{S, \varnothing\}_{K_{u2}} \| \ldots$$

In this example, we know that User 1 has read and write access because K'_{rw} is included with S in the information protected by User 1's public key K_{u1}. Similarly, we know that User 2 only has read access, because only S is encrypted with K_{u2}. Additional users' access control information would also be included.

Reading Protected Information. The process for reading protected information begins with the PVDS X retrieving protected data from the LDAP server. The PVDS checks to see if client X can decrypt the protected information by detecting if S has been encrypted using K_X, in which case the PVDS decrypts S using K'_X. At this point, the data can be decrypted using S, but we need to verify it is authentic first. To do so, the PVDS retrieves K_o from the KMS using ID_o to uniquely identify the owner. With K_o, the signature on K_{rw} can be verified, and K_{rw} can be used to verify the signature on the data. Once verified, the PVDS passes the plaintext data to the local application.

Writing Protected Information. The process for writing protected information is similar to the process for reading. However, the PVDS must also verify X has access to write the data, specifically, does X know K'_{rw}? Any client with S can actually encrypt new information. However, only those clients with K'_{rw} can sign it. If K'_{rw} is found encrypted with K_X, it is decrypted and used to sign the modified data.

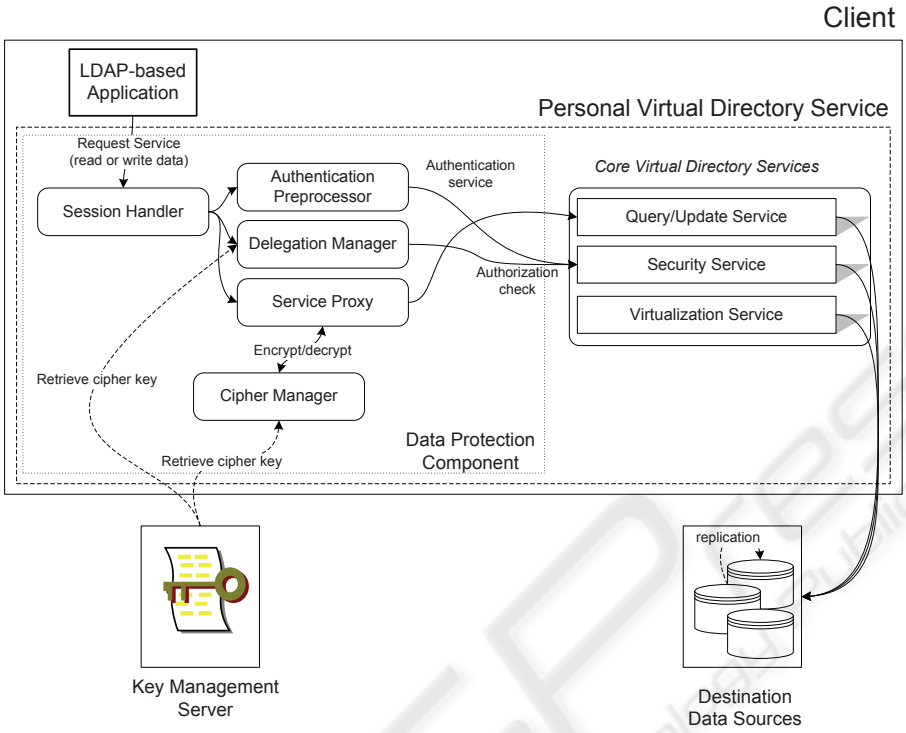

Fig. 2. Personal Virtual Directory Service Framework.

3.6 Putting It All together

The interaction with the KMS when protecting sensitive directory services information requires various core components of virtual directories, as well as the creation of several new components to handle encryption, decryption, and delegation, as described above. We propose a new framework to describe the interaction between these components, which is shown in Figure 2.

4 Discussion

Our solution provides several advantages over existing work for protecting directory information. However, it also raises new possibilities for attacks on the data, which must be considered. Additionally, this approach will have an effect on client performance and destination directory storage size. To address this impact, we show the results from a proof-of-concept implementation of our solution.

4.1 Data Protection

Using this solution, all sensitive information stored in a EDS can be encrypted, preventing the capture of an entire directory store from revealing sensitive information.

The data is protected by a single symmetric key, S, meaning if an attacker could obtain S, the information in the directory would be compromised. However, S is not stored on any client system, nor is it kept unencrypted in the directory server; S is protected using the keys of authorized users. Those keys are protected by an existing KMS, which still has vulnerabilities, but is specifically designed to reduce the risk of compromise.

Additionally, data authenticity is assured through the use of digital signatures. While all authorized users can modify the data using S, only those with K'_{rw} are able to digitally sign the encrypted information. This signature can be verified by all clients using K_{rw}, which itself is verified using a signature based on the key of the owner, K'_o.

4.2 Advantages

There are several administrative advantages to this solution as well. First, it uses existing infrastructure, specifically KMS and EDS, which do not need to be modified in any way to support data protection. The client does need to be modified to include the PVDS, and client LDAP applications must be reconfigured to point to the PVDS, but existing LDAP clients should be able to handle this modification.

Another major advantage over previous approaches is the removal of user-based key protection. Additionally, by not relying on password-based verification for data protection, we eliminate a major cause of administrative overhead. By moving the data protection component from a VDS to the client machine, we eliminate the VDS as a single point of failure (or attack). Additionally, we enhance the capability for interaction between the user and PVDS by placing it on the client machine.

The usability of this solution is quite straightforward, with little user interaction required for both configuration and continued use. Because the configuration of the PVDS is not dependent on any particular destination directory, the advantages of using a centralized VDS are not lost by moving the components to the PVDS. Some bootstrapping is required initially to mark attributes as protected, but this is a one-time occurrence for each protected attribute, and can easily be handled by interfacing with the PVDS. Additionally, our solution is compatible with any LDAP directory store and any LDAP client. In fact, by leveraging the core components of VDS, we also extend the protection capabilities of our solution to additional data sources, such as databases [9].

4.3 Attack Models

Our solution presents some opportunities for attack. Compromising a client machine is still a problem, though this can be minimized through good enterprise security practices. Attacking the information in the EDS would require the compromise of existing cryptographic techniques, such as RSA encryption [11], which is shown to be infeasible using current technology. This prevents an attack where a malicious administrator simply copies the entire directory store to attempt data extraction.

The most promising attack approach would be impersonating a legitimate client or the data owner. To do so, an attacker would need access to the client or owner private key, K_u or K_o. This could be accomplished by either stealing the user's credentials for the KMS, or by resetting the user's credentials in the KMS and using the new credentials. While security of KMS is beyond the scope of this paper, we should point out that

proper use of separation of duties when designing and implementing KMS solutions can prevent a domain administrator from carrying out this attack.

4.4 Implementation

To demonstrate the feasibility of our solution, we implemented a proof-of-concept PVDS and measured various aspects of system performance. Our EDS was implemented using Microsoft ADAM [12], a simulated KMS was constructed, and we measured components such as time to access encrypted attributes, time to access non-encrypted attributes (using the PVDS), and storage requirements for the destination data store. We used real-world data from actual EDS, including approximately 15,000 user objects with 50-75 attributes populated for each object. To mimic data protection conditions, we considered protecting roughly 4% of the attributes for each object, a figure obtained by evaluating current implementations of EDS and considering potentially sensitive information. The average attribute access time (reads and writes) is shown in Table 1, and the size of the underlying directory store when using PVDS-protected attributes versus non-protected attributes is shown in Table 2.

Table 1. Average time to access directory attributes (ms).

Configuration	Time (ms)
No PVDS - no encryption	5.5
PVDS - not encrypting	8.0
PVDS - 4% of attributes protected	12.6

Table 2. Directory size on disk (MB).

Configuration	Beginning size (MB)	Final size (MB)
No PVDS - no encryption	6.3	56.6
PVDS - 4% of attributes protected	6.3	89.9

5 Conclusions

We have presented a solution for protecting sensitive information in EDS. Utilizing existing key management infrastructure, we have proposed a novel approach to the concept of virtual directories, by moving the core components of a VDS onto the client, and introducing new components to manage data protection between and LDAP client and LDAP server. We have shown how our solution prevents insider administrative attacks, a major risk particularly among government institutions. Finally, we have demonstrated that our solution is feasible in enterprise computing environments by showing proof-of-concept implementation results.

Future work in this area should include careful analysis of attack models against KMS. Also, usability of the solution with actual users and data should be analyzed to

determine how easily this solution could be implemented in the workplace. Finally, increasing performance times should be addressed. We believe that protection of sensitive directory information is a critical task facing enterprise system administrators, and we hope that our solution provides a solid step forward in the efforts to secure this data.

References

1. Jakobsson, M.: Modeling and preventing phishing attacks. In: Phishing Panel at Financial Cryptography. (2005)
2. Kowalski, E., Cappelli, D., Conway, T., Willke, B., Keverline, S., Moore, A., Williams, M.: Insider threat study: Illicit cyber activity in the government sector. Technical report, U.S. Secret Service and CERT (2008)
3. Keeney, M., Capelli, D., Kowalski, E., Moore, A., Shimeall, T., Rogers, S.: Insider threat study: Computer system sabotage in critical infrastructure sectors. Technical report, U.S. Secret Service and CERT/SEI (2005)
4. Shaw, E., Ruby, K., Post, J.: The insider threat to information systems. Security Awareness Bulletin (1998)
5. Microsoft Corporation: How to mark an attribute as confidential in windows server 2003 service pack 1. (http://support.microsoft.com/kb/922836)
6. Red Hat, Inc.: Fedora directory server. (http://directory.fedoraproject.org/)
7. Claycomb, W., Shin, D., Hareland, D.: Towards privacy in enterprise directory services: A user-centric approach to attribute management. In: Proceedings of the 41th IEEE International Carnahan Conference on Security Technology, Ottawa, Canada (2007)
8. Radiant Logic, Inc.: Using virtualization to leverage your investment in active directory. Technical report, (Radiant Logic, Inc.)
9. Radiant Logic, Inc.: Radiantone vds. (http://www.radiantlogic.com/main/)
10. Claycomb, W., Shin, D.: Protecting sensitive information in directory services using virtual directories. In: Proceedings of the 5th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom). (2008)
11. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21 (1978) 120–126
12. Microsoft Corporation: Windows server 2003 active directory application mode. (http://www.microsoft.com/windowsserver2003/adam/default.mspx)