Distributed Directory: Security Enhancements

Tik Ning Cheung, Jeannie Li, Henry Ng

Abstract

The main objective of the project is to add security enhancements to the existing Distributed Directory. Distributed Directory is a decentralized listing of user's contact information. Each entry is stored in each user's web server as XML files, known as XML Profiles. The Security enhancements consist of two new features, User-initiated Profile Location Update and a Referral-based Listing Insertion. These new features will aid the directory's administrator in maintaining the integrity of the system, as well as improving the scalability of the system. The security enhancements mainly rely on asymmetric cryptography and the use of nonce for user identification.

1. Introduction

Information technology has been changing our lives ever since the Internet was invented. Distributed Directory takes advantage of this invention and is one of the many attempts to improve the efficiency in sharing information through the Internet. Nonetheless, security is an essential aspect of web applications such as this. The rest of the report will explain the issues with Distributed Directory before the security enhancements as well as the design and usage of the added features.

2. Background

Distributed Directory is an open-source, PHP-based web application that can provide a listing of personal particulars (such as name and address) based on a distributed data source. The significance of the Distributed Directory project is on how the listings information is obtained and maintained. For a traditional directory application, the information is inputted directly by the user, and is stored in the application's centralized database. Now suppose the user wishes to have his/her information listed on 10 different directories. He/She would have to enter the information 10 times. This is not only troublesome, but also error-prone and susceptible to input errors. Distributed Directory, on the other hand, presents a completely new way for personal information handling, and brings the control of personal information back to the user. Under the model employed by Distributed Directory, a user maintains a copy of his/her own personal information in the form of an Extensible Markup Language (XML) file (hereafter referred to as the user's XML profile) stored at a publicly accessible web location. In the event that the user wishes to enroll to the Distributed Directory, all he/she now needs to do is to inform Distributed Directory of the location of his/her XML. Distributed Directory then goes to that location and obtains the information whenever it is needed. This way, not only is the workload on the user dramatically reduced, the correctness of the information across multiple copies of Distributed Directory can also be guaranteed.

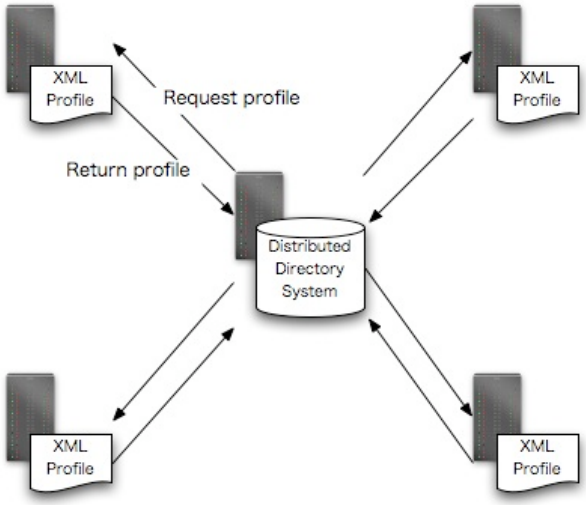

Figure 1. Distributed Directory Data Model

3. User-initiated Profile Location Update

The current version of Distributed Directory does not have a function that would allow users to update the location of their XML profiles. If an update is required, users will need to contact the administrator, who will in turn perform the updates through Distributed Directory's administration interface. This, however, leads to scalability and security issues. These issues will be discussed and addressed in the following subsections.

3.1. Issues

One of the main problems with the current XML profile location update process is the lack of scalability. The intervention of the administrator is required in order to update the location of an XML profile. This approach is fine for a small number of updates. However, note that as the number of Distributed Directory members increases, the number of updates is also likely to increase. The resultant workload makes it impossible for the administrator to maintain a Distributed Directory with a large member base.

Another issue with the update process is the possible loss of integrity. As it currently stands, there is no way for the administrator to verify the origin and the authenticity of update requests. Malicious parties may fabricate update requests that would result in the replacement of legitimate entries with undesirable content. This flaw in validating the update request causes the system to fail in protecting the integrity of the data in Distributed Directory.

Solutions to these problems will be discussed in the next subsection.

3.2. Implementation Decision

To solve the scalability problem, an overhaul of the XML profile location update process is needed. Under the new process, the user will able to initiate and perform the update without involving the administrator, thus satisfying the need for scalability.

As for the issue regarding integrity, the problem would be solved if the user can somehow prove his/her identity. The new process adopts an asymmetric cryptographic solution to address the issue. Asymmetric cryptography is a form of cryptography that involves the usage of a public/private key pair. Messages encrypted with the public key can only be decrypted with the private key, and vice versa. For the application of this project, each user will have a unique key pair, generated using the RSA algorithm. Only the user will know his/her private key. The corresponding public key is published as part of the XML profile provided to Distributed Directory.

The user will identify him/herself by encrypting and signing an update request message using his/her private key. Distributed Directory will then verify the origin of the message by decrypting the message with the user's public key (which is known to the system). To help the user create the encrypted request message, a custom tool was developed and made available to the user. This tool was implemented in Java and deployed in the form of a Java Web Start application. The use of the Java Web Start framework is crucial; it allows users using various operating systems to take advantage of the tool, as long as the Java virtual machine is installed on the operating system.

In order to prevent attacks such as replay attacks (where a valid data transmission is maliciously recorded and repeated), Distributed Directory will generate a nonce (a large random integer) whenever a user initiates an update request. Using the provided custom tool, the user creates an update request message that includes a modified nonce (which is the server nonce minus one. The nonce manipulation is automatically performed by the custom tool). When Distributed Directory receives and decrypts the request message, it will also verify that the included modified nonce value is correct. Each nonce has a validity period of 60 minutes. Once the nonce has expired or used, it is invalidated.

The most essential modification to the Distributed Directory is the ability to decrypt messages that were encrypted with a private key. PHP's OpenSSL module provided the necessary decryption functions to implement the system-side decryption.

3.3. How to Use

Before the user can carry out the update, the user is required to possess a RSA key pair. The key pair can be generated using the OpenSSL toolkit. The keys generated must comply with the PKCS#1 standard in PEM-encoded format. In order to use the custom tool described in section 3.2, the user must also convert the private key from the PEM format to the DER format. This step can be carried out using OpenSSL, and is crucial since the custom tool is only able to read private keys in the DER format.

In addition, the user must have access to the custom tool. The tool can always be obtained via a link on Distributed Directory's website. The user may choose to keep the tool executable on his/her computer and run the tool without re-downloading it every time.

There are three steps in updating the location of an XML profile registered on Distributed Directory. The user first goes to the update page, and enters the original location of his/her XML profile (the one currently registered on Distributed Directory). This allows the system to determine which listing the user wishes to update. The system then displays a randomly generated nonce, and provides a textbox where the user can enter the encrypted update request message, as shown in Figure 2 below.

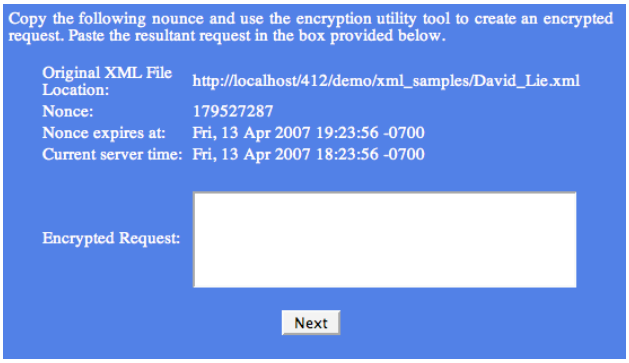

Figure 2. XML Profile Update Page

The second step is to generate the encrypted request message. The user should run the provided custom tool (shown in Figure 3), and provide the private key, the server-provided nonce, and the desired new location of the XML profile. After clicking the "Encrypt" button, an encrypted request message is generated.

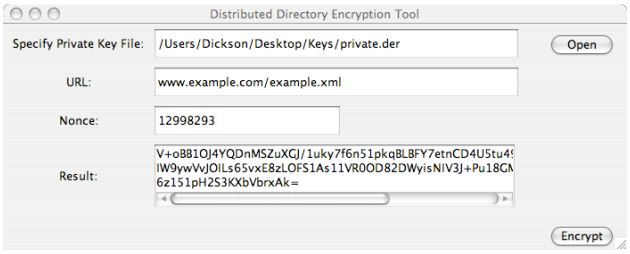

Figure 3. Java Encryption Tool

For the last step, the user should paste the encrypted message to the provided textbox on the Distributed Directory website, and ask the system to verify the message. Once the message is verified, the location of the XML profile will be updated accordingly.

3.4. Future Improvements

Currently, the message encryption process uses a Java program. However, the orginal choice of programming language for the custom encryption tool was Javascript. The original idea was to provide the user with a downloadable Javascript file that the user can run on his/her own computer. This way, the user would not be required to have the Java virtual machine installed; only a Javascript-capable web browser is needed. However, due to time constraints and limited information, a suitable Javascript-based encryption algorithm that matches the decryption algorithm of PHP's OpenSSL module could not be developed in time. To improve usability of the Distributed Directory, it is recommended that a Javascript version of the custom tool be developed for future version of Distributed Directory.

4. Referral-based Listing Insertion

The referral-based listing insertion mechanism is another additional security-related feature added to Distributed Directory. The following subsections will discuss why the feature is needed and how it was designed.

4.1. Issues

An issue with Distributed Directory was that anyone could insert entries into the directory without any form of verification and validation. This presents a great vulnerability of the system, where malicious parties can insert undesirable content (such as advertisements) into the directory. To maintain integrity of the system, only qualified users should be able to add listings to the directory. It is possible to require approval from the administrator for every user wishing to insert a directory listing; however, it lacks scalability. The solution to this problem will be discussed in the following subsection.

4.2. Implementation Decision

To maintain scalability and integrity of the Distributed Directory, the listing insertion process was changed to adopt a referral-based model. When a new user wants to insert a directory listing entry to Distributed Directory, he/she must first know someone who is already an existing user of the system. This existing user will act as the referrer, and ensure that the new user, the referee, is a qualified user. When the referrer gets a request from a new user indicating the intent to add a new listing to the directory, he/she will send an approval to the system. The identity of the referrer and the approval is then verified by the system using the same identification mechanism implemented for the XML profile update process in section 3.

With this referral system, the administrator is relieved from the burden of reviewing every newly added listings in the directory, ensuring their correctness and appropriateness. The existing users will take part in ensuring the integrity of the system, and share the administrator's workload.

4.3. How to Use

The referral process takes mainly three steps to complete. The first step is to initiate a request. When a new user wants to register in the distributed directory, the user first goes the sign-up page in the Distributed Directory website. Then, the user will be asked to enter the location of his or her XML Profile. In the next step, the system displays the information that is acquired from the specified XML Profile in the previous step, and asks

the user to review the information. The system also asks for the new user's name and contact email as well as the referrer's name and contact email. An email containing a link to the User Referral page at the Distributed Directory website will be sent to the referrer.

Upon receiving the email the referrer may carry out the final step. In the email, a link will take the referee to the User Referral page on the Distributed Directory. The referrer is first asked to provide the location of his or her existing XML Profile. This will help the system in identifying the identity of the referrer. The system then asks the referrer to review the information acquired from the referee's XML Profile as shown in Figure 4.

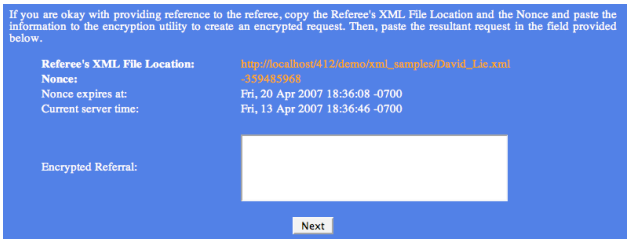

Figure 4. User Referral Page

Similar to the XML Profile's update procedure, when the referrer confirms the displayed information, he or she will generate an encrypted message with the provided nonce, as well as the location of the new user's XML Profile as a confirmation. Once the referrer copies and pastes the encrypted message from the encryption tool to the textbox provided in the User Referral page, the referrer may proceed. Once the system successfully verifies the encrypted message, the referee's new listing will be added to the directory.

4.4. Future Improvements

The current referral system relies on the existing user in qualifying new users and this result in a potential threat to the integrity of the Distributed Directory. Despite gaining an approval from an existing user, a new user may actually be a threat to the system. There is a possibility that the referrer is at fault.

As a precaution to this situation, a referral logging mechanism can be introduced to the system. The system can keep track of each referral that took place, and record both the identities of the referrer and referee. Should a user become a threat to the system, the origin of the user can be traced, and action can be taken.

5. Conclusion

The main objective of this project was to add security enhancements to the existing Distributed Directory. This goal has been achieved by adding two new features to the system, User-initiated Profile Location Update and Referral-based Listing Insertion. Even though there are more improvements that could be made to the system, overall, the security of Distributed Directory is enhanced greatly compared; the objective of the project is well achieved.

6. Bibliography

[1] H. Ng, "Distributed Directory: Implementation of a Web Application for the Listing of Personal Information," EECE 496 Final Report, April 2007.