Proposal

Centralized Key Management for the Alliance PKI

November 25, 2002

1. Introduction

The current Alliance PKI is a traditional PKI. End entities generate and manage their own key pairs. The CA is offline, meaning that a human operator must verify and approve each certificate request and insert a hardware device containing the CA's private key into the CA computer to sign certificates. The Alliance CA consults the Alliance user database during enrollment to verify that the requester is a known Alliance user and to allocate a unique distinguished name for that user.

The Alliance Certificate Policy [1] specifies rules for the operation of the Alliance CA, including the requirement of an annual compliance audit. Certificate subject names are guaranteed to uniquely identify individuals. Certificate Revocation Lists (CRLs) are issued at least once per day. CA operations are recorded in audit logs, which are reviewed at least once a week, and the audit logs are securely archived for two years. A disaster recovery plan provides resumption of CA services within 96 hours of an unanticipated emergency, and a key compromise plan addresses procedures that will be followed in the event of a compromise of the CA's private signing key. Multiple CA operators are required, each with distinct roles, to provide reasonable resilience to insider attack.

The Alliance PKI has been in operation since March 2000. In that time, the Alliance CA has issued approximately 350 certificates, with approximately 250 certificates currently valid. However, the Alliance PKI has proven difficult to use and support. End entities are responsible for managing their credentials, including initially obtaining credentials (i.e., PKI enrollment), renewing credentials, and keeping their credentials secure. Alliance users find this process cumbersome. Enrollment and renewal is error-prone, and users must manually copy their credentials to the systems from which they perform PKI authentication operations. There are also serious security concerns with end entity management of private keys: users don't choose good passwords and don't follow good practices for securing their private key files.

These concerns can be partially addressed by better software and documentation. However, a traditional PKI can exert only limited technical control over client-side operations. The CA has no knowledge of how the user generated his or her private key, cannot technically enforce any rules for choosing good passwords, and cannot technically control how the user protects his or her private key. The best the PKI can do is to specify policy that users are required to follow. For example, the Alliance CP [1] includes a statement of subscriber obligations, stating that subscribers are required to "generate a key pair using a trustworthy method" and "protect the pass phrase (private key) from others" but the CP does not define a trustworthy method for generating key pairs or how the pass phrase and private key should be protected.

We propose an alternative PKI infrastructure for the Alliance that centralizes key distribution and key management rather than requiring end entities to manage their own long-term private keys. We describe two approaches: an online CA and an online credential repository. We believe that centralizing key distribution and key management has the potential to improve usability, manageability, and security of the PKI infrastructure. A centralized Key Distribution Center (KDC) is a well-accepted practice in the Kerberos authentication system [2]. We believe the same approach can be applied effectively to a PKI.

2. Integration with the NCSA Kerberos Infrastructure

Kerberos is the foundation of the NCSA authentication infrastructure. All NCSA users have Kerberos credentials. Do non-NCSA Alliance users have Kerberos credentials?

When a user obtains an allocation on an NCSA resource, the NCSA allocations group sends the user's initial Kerberos password to the user via U.S. mail. This initial password is a shared secret between the user and NCSA. The user logs in to NCSA with that password and changes his or her Kerberos password to something more memorable. However, if the user's password ever needs to be reset by NCSA staff, for example, if the user forgets his or her password or the password is compromised, the user's password is reset to the initial password received in the mail.

Ideally, we would like to leverage the existing NCSA authentication infrastructure within the Alliance PKI to save work. One option is to have users authenticate to the PKI with Kerberos for enrollment. Another option is to use the initial password outside of Kerberos for secure PKI enrollment. Using the password outside of Kerberos has the drawback that the two passwords may become unsynchronized, for example, when the user changes his or her Kerberos password without changing his or her PKI password. However, the benefit is that the PKI is not strictly tied to Kerberos, so the PKI could serve a different community from the Kerberos realm or NCSA could consider moving away from Kerberos at a later time.

3. Online Certificate Authority: KCA/Kx509

An online CA can simplify PKI enrollment and key management by building on existing online authentication mechanisms. Users authenticate to the online CA and issue a certificate request. If the user's authenticated identity matches the identity in the certificate request, the CA signs and returns the certificate to the requester. The identity match may be determined by a simple transformation, i.e., by transferring identity information directly from the authentication mechanism to the certificate, or the online CA can maintain an identity-mapping database.

The online CA must have a CP like any other CA. The CP will state what authentication mechanisms the CA will accept and how identities are mapped from the source authentication mechanism to the PKI.

KCA/Kx509 [3] provides an online CA tied to a Kerberos realm. The distinguished name of certificates signed by the KCA includes the Kerberos ID (username@REALM) with which the requester authenticated. Development is underway to add KCA support for LDAP authentication.

4. Online Credential Repository: MyProxy

An alternative to the online CA model would be to generate credentials using an offline CA and store the credentials in an online credential repository like MyProxy [4]. The CA's CP would need to allow key generation and storage on the user's behalf. Many CPs require the end entity to generate the key pair and maintain sole possession of that key. For example, the current Alliance CP [1] states, "Under no circumstances shall the CA (or any other entity involved in the certificate administration process) have access to the private keys of any subscriber to whom it issues a certificate that references this Policy." Therefore, moving to this model for the Alliance would require moving to a new CA with a CP that allows centralized key generation and storage.

Generating PKI credentials for a user would be a step in the account setup process. When a user obtains a Grid-enabled Alliance account, the NCSA allocations group would generate a key pair for that user, encrypt the private key with the user's default password, have the certificate signed by the CA, and store the private key and certificate in the MyProxy repository. When the user receives his or her account creation notification in the mail, it will include notice that the user has a PKI account. The user runs MyProxy client commands to change the PKI account password and retrieve short-term proxy credentials on demand. If all accounts are Grid-enabled by default, PKI enrollment will be included in account creation and will not require any additional steps by the user to setup (i.e., he or she would not need to request a certificate or fax identification to the CA). The PKI account could be a new account type, in addition to mass storage accounts, AFS accounts, and accounts on specific computers. Requesting a PKI account allocation could follow the same process as these other account types.

5. Analysis

To evaluate the options for the Alliance PKI, we consider usability, manageability, and security.

5.1. Usability

MyProxy and Kx509 both improve PKI usability by removing the burden of enrollment, key generation, and key management. Alliance users can retrieve PKI credentials whenever and wherever they need them via pass phrase or Kerberos authentication.

MyProxy and Kx509 clients can be provided for all Alliance platforms. C and Java versions of the MyProxy client software are available. Kx509 clients can be built on Windows, MacOS, and all Unix platforms. Unlike MyProxy, Kx509 requires Kerberos support on the client side, but we already provide and support Kerberos client software for NCSA users.

Support for Kerberos authentication in MyProxy will soon be available, and adding password authentication to Kx509 would be trivial. Ideally, we would like to provide a true single sign-on for PKI users, where users can obtain Kerberos, AFS, and PKI credentials without entering more than one password. Translation between Kerberos and PKI credentials in both directions would allow bootstrapping from either mechanism. The SSH and GRAM protocols support delegation of only one credential per session, so it will be necessary to transparently obtain other credentials using the initially delegated credential as users delegate credentials between machines.

5.2. Manageability

A significant benefit of the online CA approach is manageability. The online CA must be professionally managed so it is reliable and secure but otherwise does not require significant changes to the NCSA allocations process and does not require traditional CA operations such as manually approving requests and physically managing an offline CA key.

Like the online CA, the credential repository must be professionally managed for reliability and security. However, the credential repository approach has the added cost of managing the offline CA, which is significant but less costly than managing a traditional offline CA because the CA operations can be batched as part of normal accounting operations. Rather than interacting directly with PKI users, signing certificates on demand, the offline CA could sign new certificates once per day or once per week while new account information is being sent in the mail.

Certificate revocation may be unnecessary in either the online CA or credential repository case, which can simplify the PKI deployment. The online CA signs only short-lived certificates, so rather than revoking a compromised certificate, the account can be disabled until the compromised certificate expires. A compromise of the credential repository, since it potentially involves all credentials in the PKI, can be handled like a CA compromise, by abandoning the CA, rather than revoking all the certificates in the repository.

5.3. Security

Moving from end entity key management to centralized key management has a number of tradeoffs. Centralized key management puts the keys in the hands of security professionals, who are more likely to follow good security practices. However, a centralized KDC is an attractive target for attack, as a compromise of the KDC compromises the entire security realm, as opposed to a compromise of a single key on an end entity workstation. In the current Alliance PKI, we have the worst of both worlds, as most users store copies of their credentials on a single NFS filesystem, in addition to copies transferred to their personal workstations.

A compromise of an online CA would allow the attacker to sign certificates for any identity in the PKI. If the CA key is directly accessible, the attacker could steal the key and sign bad certificates from another location at a later time. Alternatively, if the CA key is hardware protected, the attacker would need to fool the hardware device into signing bad certificates during the attack. In either case, the compromised CA would need to be abandoned, and a new CA would need to be created in its place, requiring notification of all relying parties that the old CA certificate should no longer be trusted and a new CA is now in place.

A compromise of a credential repository would give the attacker access to all credentials in the repository. Encrypting the credentials in the repository requires the attacker to perform an additional offline attack on each credential. Sufficiently strong encryption may make the offline attack prohibitively expensive. However, if the credentials are compromised, either all the credentials must be revoked, resulting in a very large CRL, or the underlying CA must be abandoned.

If the credential repository is only populated with credentials for users who want them, a compromise impacts only the PKI users rather than all users in the realm, as would be the case in an online CA compromise. However, this is probably a small consolation.

6. Conclusion

Moving to a centralized key distribution model for the Alliance PKI has the potential to improve usability, manageability, and security. The online CA model appears to have the lowest management cost, while the repository model is potentially more resilient to attack. The transition will require integration with the Alliance allocation process, packaging and documentation of client software, the establishment of a new CA with a new CP, and management of the new PKI components (repository and/or CA).

7. References

[1] National Computational Science Alliance Certificate Policy, Version 0.9.1, June 30, 1999. http://www.ncsa.uiuc.edu/UserInfo/Grid/Security/AllianceCP9.1.html

[2] Neuman, B. C. and Theodore Ts'o, “Kerberos: An Authentication Service for Computer Networks”, IEEE Communications, Volume 32(9), pages 33-38, September 1994.

[3] Kornievskaia, Olga, Peter Honeyman, Bill Doster, and Kevin Coffman, “Kerberized Credential Translation: A Solution to Web Access Control,” USENIX Security Symposium, Washington, D.C., August 2001.

[4] Novotny, J., S. Tuecke, and V. Welch, “An Online Credential Repository for the Grid: MyProxy”, Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.