Brokered Authentication: X.509 PKI

Context

Web services must authenticate clients so that additional controls, such as authorization and auditing, can be implemented. The organization has decided to use *brokered authentication*, based on the need for a single sign on (SSO) solution and to allow multiple Web services to share a standard access control infrastructure. The authentication broker should issue signed security tokens that can be used for authentication.

Problem

How does the Web service verify the credentials presented by the client?

Forces

Any of the following conditions justifies using the solution described in this pattern:

- **The environment includes multiple organizational boundaries or autonomous security domains**. The authentication broker must be able to issue security tokens that can be used across organizational boundaries.
- **The client and the Web service do not trust each other**. The client and the Web service may not trust one another to manage or exchange shared secrets securely. Establishing trust directly between a client and Web service could require offline interactions that can hinder clients and services from interacting dynamically.
- **The authentication broker might be offline or unavailable on some occasions**. The Web service must be able to validate authentication credentials when the authentication broker is not available. This ensures that the Web service can continue to process requests, even if the authentication broker becomes unavailable.
- **Clients that require authentication are implemented on a variety of platforms within the organization, and interoperability is required between those platforms**. Using a standards-based mechanism for authentication helps ensure interoperability between different platforms.
- **The organization may need to trace particular actions to a specific client or service**. A record of transactions allows an organization to provide evidence that a particular action was requested and/or performed. This could be useful if a user denies that he or she performed an action or if a client needs to verify that a service has performed a specific task.

Solution

Use brokered authentication with X.509 certificates issued by a certificate authority (CA) in a public key infrastructure (PKI) to verify the credentials presented by the requesting application.

The client application attaches credentials (or a reference to credentials) to the request message and digitally signs the message with the client's private key. When a service receives the message, it uses the public key, which is included with the X.509 certificate, to validate the signature. Additional validation may be required to ensure that the X.509 certificate has not expired and was issued by a CA that the service trusts.

Participants

Brokered authentication with X.509 certificates issued by a certificate authority in a PKI involves the following participants:

- **Certificate authority (CA)**. A CA is an authentication broker that is responsible for authenticating clients and issuing valid X.509 certificates.
- **Certificate store**. This is where the X.509 certificates are located.
- **Client**. The client accesses the Web service. The client provides the credentials for authentication during the request to the Web service.
- **Service**. The service is the Web service that requires authentication of a client prior to authorizing the client.

Process

A mutually trusted CA must issue an X.509 certificate before brokered authentication using X.509 can complete. You can obtain an X.509 certificate in one of the following ways:

- Purchase an X.509 certificate from a public CA.
- Configure a PKI server, such as Windows Certificate Services, to create an X.509 certificate, and then use the PKI CA to sign the certificate.
- Use a tool such as MakeCert to create a self-signed certificate (this is not suitable for production purposes).

After an X.509 certificate is issued, local repositories, such as a machine certificate store, are used to store information about the X.509 certificate. The actual process of issuing and distributing X.509 certificates is beyond the scope of this pattern. For detailed information, see X.509 Technical Supplement in Chapter 7, "Technical Supplements."

The process of using an X.509 certificate for authentication is shown in Figure 1.10.

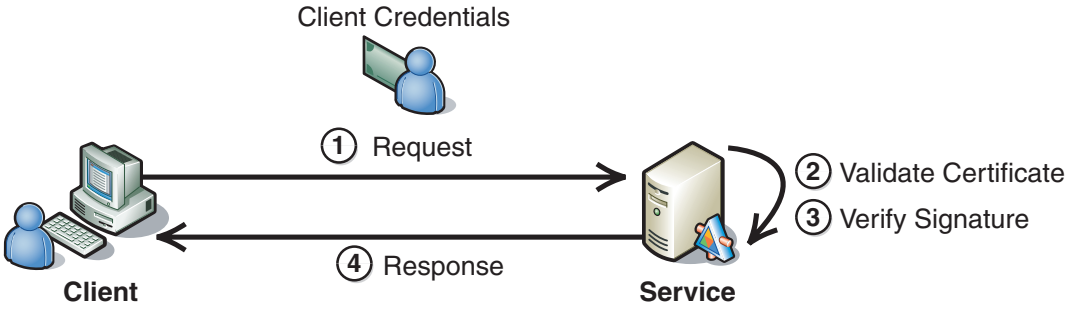

Figure 1.10
Authentication using an X.509 certificate

As illustrated in Figure 1.10, the following steps describe the process of authentication using an X.509 certificate:

1. The client sends a message to the service. The message includes the client's credentials, signed with the private key that is paired with the public key in the client's X.509 certificate. The client can also attach its X.509 certificate to the message if the service does not store or have access to the X.509 certificates out of band. If the X.509 certificate is not attached, the client attaches a certificate identifier to the request message so that the service can retrieve the client's X.509 certificate from a certificate repository and verify the message signature.
2. The service validates the certificate, by performing a number of checks, including:
   - Verifying that the certificate has not expired. If the expiration date in the certificate is past the current date, then the certificate is not valid.
   - Verifying that the certificate is internally consistent. The service checks that the data in the certificate has not been tampered with by verifying the certificate contents against the signature of the issuing CA.
   - Verifying the issuing CA of the client's X.509 certificate. This is done by comparing the issuer signature on the user's X.509 certificate with the X.509 certificate of the issuing CA. For this step to be of any value to either party, the CA that issued the client's X.509 certificate must be trusted by both the client and service.
   - Verifying that the issuing CA has not revoked the certificate. The service checks this by making sure that the X.509 certificate does not appear on a certificate revocation list (CRL) published by the issuing CA. The service can check the revocation status of the certificate by directly accessing it from the CA or by checking against a CRL that was previously downloaded from the issuing CA to the certificate repository used by the service to look up X.509 certificates.

3. The service uses the public key in the client's X.509 certificate to verify the client's signature. This allows the service to authenticate the client and ensure that the signed data has not been tampered with after the message was signed.
4. (Optional) The service may send a response back to the client.

Resulting Context

This section describes some of the more significant benefits, liabilities, and security considerations of using this pattern.

Note: The information in this section is not intended to be comprehensive. However, it does discuss many of the issues that are most commonly encountered for this pattern.

Benefits

The benefits of using the Brokered Authentication: X.509 PKI pattern include the following:

- Authentication can occur over well known Internet firewall-friendly ports through well-known protocols (for example, HTTP/HTTPS over port 80/443).
- X.509 certificates can be used to authenticate clients and protect messages across organizational boundaries and security domains because the X.509 certificates are based on a broadly accepted standard. PKI using X.509 certificates has the capacity to establish a common basis of trust beyond the scope of individual organizations. Only a relatively small number of certificate issuers are widely trusted across public networks, which simplifies the management of trust with those issuers.
- The X.509 CA supports renewal and revocation of X.509 certificates, as follows:
  - An agent, acting on the client's behalf, can renew an X.509 certificate to extend the life time of the certificate. When an X.509 certificate is renewed, a new copy of the certificate is generated with a new expiration date, sometimes along with a corresponding new public/private key pair.
  - X.509 certificates may be revoked if any of the client's information in the X.509 certificate has changed or if the X.509 certificate's private key has been compromised.
- X.509 certificates can be distributed openly and used by anyone to encrypt messages to a client or to verify the digital signature of the client. For more information about protecting confidential data, see Data Confidentiality in Chapter 2, "Message Protection Patterns."
- Digital signatures provide a means of supporting non-repudiation. This is because access to the private key is usually restricted to the owner of the key, which makes it easier to verify proof-of-ownership. For more information about non-repudiation, see Data Origin Authentication in Chapter 2, "Message Protection Patterns."
- Authentication does not require a direct relationship between every client and service.

Liabilities

The liabilities associated with the Brokered Authentication: X.509 PKI pattern include the following:

- Private keys need to be stored securely (such as on a smart card or your computer) and are therefore not as portable as passwords. An attacker could use a private key to impersonate the client. Therefore, you must make sure that the private key is not compromised.
- Generating and verifying digital signatures in X.509 is computationally intensive. If the client sends frequent request messages to the service during a normal interaction, you should consider a means to optimize communication between the two parties, such as secure conversation.
- Certificates by themselves are not well suited to provide role-based security, because role assignment tends to change relatively frequently and X.509 certificates typically have a long life time. However, you can supplement X.509 certificate authentication with a role store to provide more fine-grained authorization capabilities. One possible solution is to combine X.509 authentication with a Lightweight Directory Access Protocol (LDAP) directory or Active Directory with certificate mapping enabled.
- Organizations could require additional infrastructure to support an X.509 PKI. The benefits gained from using an X.509 PKI must be compared with the investment required to use it.

Security Considerations

Security considerations associated with the Brokered Authentication: X.509 PKI pattern include the following:

- It is critical to safeguard the private key associated with the X.509 certificate. If the private key is compromised, the integrity of the corresponding X.509 certificate is violated because another entity besides the client is capable of generating digital signatures that represent the client's identity. If a private key is compromised, the CA can revoke the X.509 certificate, which causes it to become unusable for encryption and digital signatures.
- The life time of an X.509 certificate is considerably greater than that of other authentication broker token types. Most tokens from an authentication broker expire minutes or hours from their time of issue, whereas an X.509 certificate can be valid for several months.
- Regardless of whether an X.509 certificate is renewed or revoked and a new X.509 certificate is re-issued, the X.509 certificate should use a newly generated public/private key pair. For existing X.509 certificates that are being renewed, this is known as re-keying the X.509 certificate.
- Only one copy of the client's X.509 certificates private key should exist when it is used to support non-repudiation through digital signatures. This private key should be accessible to the client only.

- If private keys are centrally managed — for example, by using a key escrow — and the centralized store is compromised, you may not be able to use digital signatures to strongly attribute an action to a specific party.
- In some cases, after a service has authenticated a client, it will need to authorize the client based on the client identity. The service must be able to either recognize the client individually or verify that the client belongs to a limited population. The service can accomplish this in one of the following ways:
  - By defining a policy that only allows requests to be processed that are signed by specific X.509 certificates.
  - By requiring verification of X.509 client certificates against a very restricted trust chain. This allows you to closely regulate the population of clients from which the server will accept requests. For more information about X.509 certificate trust chains and trust anchors, see X.509 Technical Supplement in Chapter 7, "Technical Supplements."
- Messages that are signed and encrypted with X.509 certificates are susceptible to *surreptitious forwarding* attacks. In this type of attack, the recipient of a signed and encrypted message decrypts the message, encrypts it using a third-party's public key, and then sends it on to that third party with the original signature still in the message. In this case, the message can appear as though it was sent to the third party from the original sender. To mitigate this type of attack, the original sender can sign some information that binds the message to the intended recipient, such as the WS-Addressing headers that specify the intended recipient of the message.
- If an authentication broker is compromised, the integrity of the trust that the broker provides is also compromised. If a CA is compromised, an attacker could issue certificates to himself/herself to act as a valid client within the CA's trust chain. An attacker could use these certificates to perform malicious actions while posing as a trusted client.
- You should use mutual authentication to be sure that each party using X.509 is who they claim to be. With mutual authentication, the client authenticates the service and the service authenticates the client. For authentication with X.509 certificates, each party must be able to verify a piece of signed data provided by the other party with that party's X.509 certificate. Alternatively, if only one party has an X.509 certificate, shared keys can be combined with X.509 certificates to provide mutual authentication. For an example of such an approach, see Implementing Message Layer Security with X.509 Certificates in WSE 3.0 in Chapter 3, "Implementing Transport and Message Layer Security."

Related Patterns

Four types of patterns are related to this pattern: parent patterns, child patterns, alternate patterns, and patterns that use the Brokered Authentication: X.509 PKI pattern.

The following parent pattern is related to the Brokered Authentication: X.509 PKI pattern:

- **Brokered Authentication**. This pattern describes how to prove a client's identity to an authentication broker so that the broker can issue a security token.

The following child patterns are related to the Brokered Authentication: X.509 PKI pattern.

- **Implementing Message Layer Security with X.509 Certificates in WSE 3.0**. This pattern explains how to implement brokered authentication, authorization, data integrity, and data origin authentication using X.509 certificates in WSE 3.0.
- **Implementing Transport Layer Security Using X.509 Certificates and HTTPS**. This reference provides a concise reference on how to use SSL for data confidentiality and data integrity and how to use SSL client certificates for brokered authentication and data origin authentication.

The following alternate patterns are related to the Brokered Authentication: X.509 PKI pattern:

- **Brokered Authentication: Kerberos**. This pattern provides an alternative to X.509 based on the Kerberos authentication protocol.
- **Brokered Authentication: Security Token Service (STS)**. This pattern provides an alternative to X.509 that is highly interoperable between platforms, security protocols, and credential types.

The following pattern uses the Brokered Authentication: X.509 PKI pattern:

- **Implementing Direct Authentication with UsernameToken in WSE 3.0**. This pattern relies on relies on X.509 certificates, to ensure that sensitive credentials can be propagated securely.