Internet Draft
Jan. 8, 1998
draft-ietf-cat-mpaker-00.txt

Yongnan Xu
University of Missouri - Kansas City
Lein Harn
Racal Datacom Group

The Multiple-Path Authentication of Kerberos (MPAKER)

Status of this Memo

Abstract

This draft proposes an authentication scheme that improves the efficiency and security without losing features of the standard Kerberos and other extension schemes. Instead of completely replacing those schemes, the new scheme can be integrated with them to provide multiple-path authentication for Kerberos.

Table of Contents

1. Introduction

Kerberos [NT94] is a network authentication system. It is designed to provide strong authentication for client/server applications. With the authentication process, it can also encrypt all of their communications to assure privacy and data integrity as a security feature option. Kerberos authentication involves three parties: a client, a server and a Key Distribution Center (KDC). A Kerberos client acts on a user's behalf and is usually modified from a standard client program by adding Kerberos related functions. A Kerberos KDC services both initial ticket and service-special ticket requests. The initial ticket portion is referred to as the Authentication Server (AS), and the service-special ticket portion is referred to as the ticket-granting server (TGS).

The standard Kerberos authentication scheme [RFC1510][NKT97] has three phases. A client starts AS exchange phase by sending a request to the AS for an initial ticket at first time login. This request is sent in the clear and contains no sensitive information. The AS generates and sends back the initial ticket that includes a session key and a ticket-granting ticket. The session key is encrypted by user's secret key and will be shared between the user and the TGS. The ticket-granting ticket contains a copy of the session key and is encrypted under a secret key known only to the KDC. The client asks for the user's password and converted into the user's secret key using a one-way hash. The key is then used to decrypt the session key from the AS. The ticket-granting ticket and decrypted session key are saved as credentials.

The second phase is the TGS request and response. Before the client makes a service connection, it first communicates with the TGS for a server-special ticket. The client uses credential information and makes a request by sending the ticket-granting ticket, and identity information encrypted under the session key shared between the user and the TGS. After verifying the authentication of the client, the TGS sends to the client a server-special ticket encrypted under the server's secret key and a new session key encrypted by the original session key. This new session key will be shared between the client and requested server.

The client then connects the server and presents the server-special ticket from TGS. The server retrieves its own secret key from a secured local file, and uses this key to decrypt the ticket. There is a copy of the new session key in the ticket. If everything goes

smoothly, the server gets the new session key and proves the client's (user's) identity. This ends the application authentication (AP) exchange phase.

The standard Kerberos scheme requires that the client contact the KDC not only at first login but each time for a new server connection.

The link between the client and the KDC takes unfair workload, especially when they have poor connection in between. An extension scheme [SW97] suggests that the client sends all of KDC requests to the server and the server forwards them to the KDC. This scheme is efficient only for the cases in which there has no or poor links between clients and the KDC. This draft proposes a different scheme which keeps the original AS exchange between the client and the KDC and sends the TGS exchanges to the server. This scheme is efficient for most practical network environment and improves security without losing features of the standard and extension schemes. Instead of completely replacing the standard and extension schemes, the new scheme can be integrated with them to provide multiple authentication paths for selection. The new protocol may be used under normal authentication. The standard protocol or the extension protocol may be selected when having very poor connections between the KDC and the server or between the KDC and the client separately. A user will select different authentication paths based on network environments.

2. The basic protocol

The new scheme has the same operation environment as the standard scheme and extension scheme. To simplify discussion and compare with the two schemes, the protocol process is outlined in high level using the similar notation in [RFC1510][SW97]. The protocol detail should be developed based on the standard Kerberos protocol specification.

2.1 The AS Exchange

The new scheme performs exactly the same AS exchange as the standard scheme. When a user logs in first time, an authentication request is issued to the AS and the AS will return a session key and a ticket-granting ticket. The former is encrypted by the user's secret key, and the latter is encrypted by TGS's secret key. The user decrypts the session key and keeps the session key and ticket-granting ticket as credentials for later use.

```
Client            Server          KDC           Messages
------            ------          ---           -------
request -->     verification                    KRB_AS_REQ

verification    <-- response                    KRB_AS_REP or KRB_ERROR
```

```
AS-REQ ::=   [APPLICATION 10] SEQUENCE {
```

```
                pvno[1]             INTEGER,
```

```
                msg-type[2]         INTEGER,
                padata[3]           SEQUENCE OF PA-DATA OPTIONAL,
                req-body[4]         KDC-REQ-BODY
    }

    AS-REP ::=   [APPLICATION 11] SEQUENCE {
                pvno[0]             INTEGER,
                msg-type[1]         INTEGER,
                padata[2]           SEQUENCE OF PA-DATA OPTIONAL,
                crealm[3]           Realm,
                cname[4]            PrincipalName,
                ticket[5]           Ticket,
                enc-part[6]         EncryptedData
    }

    KRB-ERROR ::=   [APPLICATION 30] SEQUENCE {
                pvno[0]             INTEGER,
                msg-type[1]         INTEGER,
                ctime[2]            KerberosTime OPTIONAL,
                cusec[3]            INTEGER OPTIONAL,
                stime[4]            KerberosTime,
                susec[5]            INTEGER,
                error-code[6]       INTEGER,
                crealm[7]           Realm OPTIONAL,
                cname[8]            PrincipalName OPTIONAL,
                realm[9]            Realm, -- Correct realm
                sname[10]           PrincipalName, -- Correct name
                e-text[11]          GeneralString OPTIONAL,
                e-data[12]          OCTET STRING OPTIONAL
    }
```

2.2 The TGS and AP exchanges

When the user invokes a client for a service, the client opens the user's credentials cache and retrieve the user's session key and ticket-granting ticket. In the standard scheme, the user will contact and present the credentials to the TGS directly for requesting a server-specific ticket later and then connect a server. There are two ways to handle the rest of the authentication process in the new scheme. One is to allow the application server forward the TGS requests and responses, and keep AP exchange unchanged like the standard scheme. The other way is to combine the TGS and AP exchanges together.

2.2.1 The Forwarding of TGS Exchanges

Instead of sending a request to the TGS directly for a server-special ticket, the client connects the server and sends a TGS request which is encrypted with the user's session key. After receiving the authentication request, the server simply forwards it to the TGS. The TGS decrypts the ticket-granting ticket in the request using it's own secret key to get the session key, and then uses the session key to decrypt the authentication information which validates the client. Assuming everything is correct, the TGS now believes the client is a right one. The TGS generates a new session key to be shared between the user and the target server. The new session key is encrypted with the secret key for the server. The TGS also encrypts a copy of the new session key with the original session key it already shares with the user. The TGS create a new ticket including the two encrypted copies of the new session key and sends it back to the server.

The server receives the ticket from the TGS and simply forward it to the client. The client treats this reply as if it comes from the TGS directly. It then sends an AP request to the server using the same process in the standard scheme.

```
Client             Server          KDC         Messages
------             ------          ---         -------
request -->    forwarding --> verification     TGS-REQ

verification  <-- forwarding  <-- response      TGS_REP or KRB_ERROR

request -->   verification                     AP_REQ

verification  <-- response                     AP_REP or KRB_ERROR
```

```
TGS-REQ ::=   [APPLICATION 12] SEQUENCE {
              pvno[1]             INTEGER,
              msg-type[2]         INTEGER,
              padata[3]           SEQUENCE OF PA-DATA OPTIONAL,
              req-body[4]         KDC-REQ-BODY
}

TGS-REP ::=   [APPLICATION 13] SEQUENCE {
              pvno[0]             INTEGER,
              msg-type[1]         INTEGER,
              padata[2]           SEQUENCE OF PA-DATA OPTIONAL,
              crealm[3]           Realm,
              cname[4]            PrincipalName,
              ticket[5]           Ticket,
              enc-part[6]         EncryptedData
}
```

```
AP-REQ ::=   [APPLICATION 14] SEQUENCE {
              pvno[0]             INTEGER,
              msg-type[1]         INTEGER,
              ap-options[2]       APOptions,
              ticket[3]           Ticket,
              authenticator[4]    EncryptedData
}

AP-REP ::=   [APPLICATION 15] SEQUENCE {
             pvno[0]                INTEGER,
             msg-type[1]            INTEGER,
             enc-part[2]            EncryptedData
}
```

2.2.2 The Combination of TGS and AP Exchanges

In the standard scheme, the purpose of third phase (AP exchange) is to present a server-special ticket and authenticators. The former will pass a new session key to the server, and the latter will prove the identity of the client. In the new scheme, the TGS and AP Exchanges can be combined to a single modified TGS exchange (NEW_TGS-REQ and NEW_TGS_REP). It is not necessary for a client to present the server-special ticket since the server is the forwarder of TGS responses and can make copies before forwarding messages received from TGS. The authenticators can be sent with the modified TGS request encrypted by the session key from the AS exchange. The TGS will decrypt it, re-encrypt it with the server's secret key, and sends authenticators to the server with the modified TGS response. The server trusts the client if the TGS issues a non-error TGS response. The standard TGS_REQ and TGS_REP should be modified to allow the server easily to add or retrieve information.

```
Client           Server            KDC        Messages
------           ------            ---        --------
request -->   forwarding --> verification     NEW_TGS-REQ

                 verification,
verification <-- copy and    <-- response     NEW_TGS_REP
                 forwarding                   or KRB_ERROR
```

3. Efficiency and security considerations

The standard scheme requires a server-special ticket exchange between a client and the KDC each time before the client connects a new server. In practical network environment, links between servers and

the KDC usually provides faster and securer connections, in comparison with the links between clients and the KDC. It is a good reason to put more exchange load between the KDC and the servers. The extension scheme suggests a solution by removing all connections between the client and the KDC. The new scheme keeps the AS exchange between the client and the KDC and replaces the server-special ticket exchange between the client and the KDC by a message exchange between the server and the KDC. Since the AS exchange only performs once, it has little effect on the overall authentication performance unless there has no connection between the client and the KDC. It is a very rare case in real networks.

It is reasonable for a client to contact a server as early during the authentication process as possible. The standard scheme completes the server-special ticket request before a real connection to a server. It is possible that the services are not available at that time. The server may be down temporarily. A possible reason is that other access controls block the access. A user may be allowed to access FTP service in a server, and so the user's name and secret key are registered in the KDC which controls the realm. When the user tries rlogin connection to the same server, the access may be denied by security policy in the server or the site's firewalls. The KDC has no idea about possible connection fails, so the client wastes time for requesting the server-special ticket. The new scheme may get the same fail results, but it will stop at the right point of the server connection. In the extension scheme, a client contacts a server at very beginning of the authentication process, that is, at the AS exchange phase. A ticket-granting ticket is independent to servers and a client has no idea which server is the best forwarder to pass the AS request. The client will repeat the AS request until it hits a right server which accepts the forward request. In the new scheme, a client has already got a ticket granting ticket during the authentication service phase and is ready to send TGS request when connecting any server.

The new scheme also minimizes the possibility of revealing the secret keys. According to fundamental principles of cryptography, the more ciphertext to collect, the more likely to compromise the secret key. In the standard scheme, the TGS sends to the client a ticket encrypted under the server's secret key when the client requests a server-special ticket each time. The user may accumulate the ciphertext and use them to break the server's secret key. In the extension scheme, the server forwards the AS responses encrypted by the user's secret key. The server may copy the ciphertext before forwarding the responses and use them to break the user's secret key. The new scheme never exchanges any messages encrypted by either the user's or the server's secret key between them. To keep this feature, the AS exchanges should not be forwarded by the server.

4. Conclusion

The new scheme is more efficient over both the standard scheme and the extension scheme in most cases. The new scheme has the same limitations as the standard scheme and the extension scheme [BM90]. The improvement methods for the standard scheme [TRN97][TNW97] may also apply for the new scheme.

The motivation of the new scheme focuses on the efficiency and security improvement without losing features from both of the standard scheme and the extension scheme. Instead of completely replacing the two schemes, the new scheme can be integrated with them to provide multiple-path authentication in Kerberos system. The new protocol may be used under normal authentication. The standard protocol or the extension protocol may be selected when having very poor connections between the KDC and the server or between the KDC and the client separately. A user will select different authentication paths based on network environments.

5. References

[BM90] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos authenication system", Computer Communication Review, 20 (5), pp119-132, October 1990.

[NKT97] Clifford Neuman, John Kohl and Theodore Ts'o, "The Kerberos Network Authentication Service (V5)",RFC 1510 update, Internet Draft, November 1997.

[NT94] Clifford Neuman and Theodore Ts'o, "Kerberos: An Authentication Service for Computer Networks", IEEE Communications Magazine, 32 (9), pp 33-38, September 1994.

[RFC1510] John Kohl and Clifford Neuman, "The Kerberos Network Authentication Service (V5)", RFC 1510, September 1993.

[SW97] Michael M. Swift, "Initial Authentication with Kerberos and the GSS-API (IAKERB)", Internet Draft, November 1997.

[TNW97] Brian Tung, Clifford Neuman, John Wray, et al., "Public Key Cryptography for Initial Authentication in Kerberos", Internet Draft, November 1997.

[TRN97] Brian Tung, Tatyana Ryutov, Clifford Neuman, et al., "Public Key Cryptography for Cross-Realm Authentication in Kerberos",

6. Contact

Yongnan Xu
ynxu@cstp.umkc.edu

Lern Harn
Lein_Harn_at_HP3@usa.racal.com