Beyond Passwords: Extending Kerberos Authentication Mechanisms

Henry B. Hotz
Jet Propulsion Laboratory

Motivation

- The Kerberos protocol for getting service tickets and access to services is very good
  - BUT almost all Kerberos (and AFS) deployments still use passwords to get the initial ticket-granting-ticket.
    - Passwords have have obvious drawbacks, in particular being easily stolen.
- The IETF has a draft RFC on how the initial authentication framework can be extended for other methods.
  - There are other standards (or draft standards) for using X509 certificates (PKINIT), and One Time Passwords as specific alternatives.

Pre-Authentication in the Kerberos Protocol

- The initial authentication in Kerberos is how you acquire the ticket-granting-ticket.
  - The initial authentication response includes a tgt encrypted with something only you know — usually a hash of your password.
  - To avoid giving "bad guys" sample material for off-line attacks, the initial request should prove that the requester can legitimately use it.
- The data in the initial request is called pre-authentication data, or `PA-DATA`.
  - The `PA-DATA` field optionally exists in all Kerberos messages, and is a generic field for all kinds of supplemental data.
  - For traditional passwords, the `PA-DATA` field contains the current time encrypted with a hash of the user's password.

Pre-Auth, Initial Request

- Absent other knowledge, an initial request just says what the client wants (and omits the `PA-DATA` field).
  - Client and Server principal names
  - Client-supported encryption types.

```
▷ Frame 1 (231 bytes on wire, 231 bytes captured)
▷ Ethernet II, Src: AppleCom_d2:d5:2e (00:17:f2:d2:d5:2e), Dst:
▷ Internet Protocol, Src: laphotz.jpl.nasa.gov (128.149.133.44)
▷ User Datagram Protocol, Src Port: 49885 (49885), Dst Port: ke
▽ Kerberos AS-REQ
    Pvno: 5
    MSG Type: AS-REQ (10)
  ▽ KDC_REQ_BODY
      Padding: 0
    ▷ KDCOptions: 50800000 (Forwardable, Proxyable, Renewable)
    ▷ Client Name (Principal): hotz
      Realm: SCKERB.JPL.NASA.GOV
    ▷ Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
      till: 2009-05-14 08:16:21 (UTC)
      rtime: 2009-06-13 02:16:21 (UTC)
      Nonce: 3502332977
    ▷ Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac
```

Pre-Auth, Initial Response

- "Error" message from the server lists the required/ supported kinds of pre-authentication.
  - This is normal, and not really an error, however it's logged.
  - `PA-ENCTYPE-INFO2` lists supported encryption types and salts.

```
Frame 2 (474 bytes on wire, 474 bytes captured)
Ethernet II, Src: Cisco_3e:54:00 (00:1c:0f:3e:54:00), Dst: AppleCom_
Internet Protocol, Src: jplis-fil-krb04.jpl.nasa.gov (128.149.197.17
User Datagram Protocol, Src Port: kerberos (88), Dst Port: 49885 (49
Kerberos KRB-ERROR
    Pvno: 5
    MSG Type: KRB-ERROR (30)
    stime: 2009-05-14 02:16:21 (UTC)
    susec: 648830
    error_code: KRB5KDC_ERR_PREAUTH_REQUIRED (25)
    Client Realm: SCKERB.JPL.NASA.GOV
  ▷ Client Name (Principal): hotz
    Realm: SCKERB.JPL.NASA.GOV
  ▷ Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
    e-text: Need to use PA-ENC-TIMESTAMP/PA-PK-AS-REQ
  ▽ e-data
    ▷ padata: PA-ENC-TIMESTAMP PA-DASS PA-PK-AS-REP PA-ENCTYPE-INFO2
```

National Aeronautics and Space Administration

Pre-Auth, Timestamp Request

- This is the traditional pre-auth type used with passwords.

```
Kerberos AS-REQ
  Pvno: 5
  MSG Type: AS-REQ (10)
padata: PA-ENC-TIMESTAMP
  Type: PA-ENC-TIMESTAMP (2)
    Value: 3041A003020111A23A04384AD87133AE81485700533C058F..
      Encryption type: aes128-cts-hmac-sha1-96 (17)
      enc PA_ENC_TIMESTAMP: 4AD87133AE81485700533C058F20281E8
KDC_REQ_BODY
    Padding: 0
  KDCOptions: 50800000 (Forwardable, Proxyable, Renewable)
  Client Name (Principal): hotz
    Realm: SCKERB.JPL.NASA.GOV
  Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
    till: 2009-05-14 08:16:21 (UTC)
    rtime: 2009-06-13 02:16:21 (UTC)
    Nonce: 3502332977
  Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac-s
```

Pre-Auth, Timestamp Response

- Once the pre-auth data is verified, KDC returns the ticket requested.
  - More `PA-DATA` explaining how to do the password
    - "`SCKERB.JPL.NASA.GOVhotz`" salt value for password conversion.
    - This info was also in the `PA-ENCTYPE-INFO2`.

```
Kerberos AS-REP
    Pvno: 5
    MSG Type: AS-REP (11)
  padata: PA-PW-SALT
    Type: PA-PW-SALT (3)
      Value: 53434B4552422E4A504C2E4E4153412E474F56686F7
    Client Realm: SCKERB.JPL.NASA.GOV
  Client Name (Principal): hotz
  Ticket
      Tkt-vno: 5
      Realm: SCKERB.JPL.NASA.GOV
    Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
    enc-part aes128-cts-hmac-sha1-96
  enc-part aes128-cts-hmac-sha1-96
```

Pre-Auth, PKINIT Request

- While other fields stay the same, the padata field contains the appropriate data for the type of pre-authentication to be done.
- padata type 16 is more usually written like PA-PK-AS-REQ

```
Kerberos AS-REQ
  Record Mark: 3530 bytes
    Pvno: 5
    MSG Type: AS-REQ (10)
  padata: PA-DASS
    Type: PA-DASS (16)
        Value: 30820CEE80820C5E30820C5A06092A864886F70D010702A0...
  KDC_REQ_BODY
      Padding: 0
    KDCOptions: 50800000 (Forwardable, Proxyable, Renewable)
    Client Name (Principal): hotz
      Realm: SCKERB.JPL.NASA.GOV
    Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
      till: 2009-05-14 08:31:36 (UTC)
      rtime: 2009-06-13 02:31:36 (UTC)
      Nonce: 3480286936
    Encryption Types: aes256-cts-hmac-sha1-96 aes128-cts-hmac-sha1
```

Pre-Auth, PKINIT Response

- The contents of the padata field can similarly contain data specific to the type of pre-authentication being done.
  - In this case it's got both PKINIT and password responses.
  - padata type 17 is more usually written like PA-PK-AS-REP

```
▽ Kerberos AS-REP
  ▷ Record Mark: 2524 bytes
    Pvno: 5
    MSG Type: AS-REP (11)
  ▽ padata: Unknown:17 PA-PW-SALT
    ▽ Type: Unknown (17)
        Value: A08206E9308206E5808206E1308206DD06092A864886F7
    ▽ Type: PA-PW-SALT (3)
      ▷ Value: 53434B4552422E4A504C2E4E4153412E474F56686F747A
    Client Realm: SCKERB.JPL.NASA.GOV
  ▷ Client Name (Principal): hotz
  ▽ Ticket
      Tkt-vno: 5
      Realm: SCKERB.JPL.NASA.GOV
    ▷ Server Name (Principal): krbtgt/SCKERB.JPL.NASA.GOV
    ▷ enc-part aes128-cts-hmac-sha1-96
  ▷ enc-part aes256-cts-hmac-sha1-96
```

Pre-Authentication Framework

- Possible to invent new pre-auth methods to support local needs.
  - Too expensive in most cases.
- FAST (Flexible Authentication, Secure Tunneling) is a framework for supporting new pre-auth mechanisms.
  - Supports functions commonly needed by new mechanisms
    - Session encryption key derivation
    - PA data packaging and common options
  - Current implementation supports:
    - encrypted challenge (in MIT 1.8)
    - OTP (draft specification, experimental implementation internal to RSA)

Security Advantages of FAST

- FAST solves the problems needed to chain multiple exchanges and pre-auth mechanisms.
  - Multiple-exchange capability not fully implemented.
  - Functional in MIT 1.8.
- FAST does *not* solve the chicken-and-egg problem of needing a shared key to set up a client/server exchange.
- FAST *does* allow use of a high-entropy "host" key to set up the exchange instead of direct use of a password.
  - Presumes initial AS-exchanges done in a context that allows access to those keys. Not usable for command-line `kinit` without a keytab.
  - Envisioned to also support anonymous PKINIT for setup.

OTP-Specific Instantiation

Client

Generate AS-REQ

KDC

Generate KRB5_ERROR
- "Conventional" PA types (if any)
- PA-ETYPE-FAST-REPLY
- PA-OTP-CHALLENGE
- must-encrypt flag
- nonce
- optional stuff

Generate client key and reply key
Generate AS-REQ
- PA-FX-FAST-REQUEST
- PA-OTP-REQUEST
- nonce
- encData
- optional stuff

Generate client key and reply key
Verify Client
(Compare decrypted nonce in encData
with PA-OTP-CHALLENGE.)
Generate AS-REP
- PA-ETYPE-FAST-REPLY
- KrbFastFinished data
- PA-OTP-CONFIRM
- encData

Verify KDC
(Compare returned nonce in encData
with PA-OTP-REQUEST)

OTP Exchange Comments

- The FAST envelope includes an armor key, probably based on an existing kerberos ticket.
- The OTP message client and reply keys are derived from the FAST armor key and the OTP value.
  - Unless the OTP value is included in the request, then the keys are the same as the FAST armor keys.

MIT Support

- MIT Kerberos provides a plugin interface to allow add-on code to add support for new pre-authentication data types.
  - Initial availability in 1.6.2
  - Provides the supporting functionality defined to aid new PA types.
- PKINIT and encrypted-challenge are standard pre-auth plugins in current distributions.
- FAST support more mature and complete than Heimdal.
- Can be built with an open-source `ASN1` compiler
  - May be needed to encode/decode new PA types.

MIT Plug-In Interface

- Note that the Heimdal pre-auth plug-in interface is simpler and less mature.

- Globally-visible table of interface values.
- Data
  - `pa_type_list` — list of pre-auth types to advertise
  - `enctype_list` — list of encryption types supported

MIT Plug-In Interface Continued

- Functions
  - Usual sort of entry points for a plug-in.
  - `init()` — at plug-in load
    - `request_init()` — at start of request
      - `process()` — do the work
      - `tryagain()` — process error data to support another attempt
    - `request_fini()` — at end of request
  - `fini()` — before unloading plug-in
  - Helper functions
    - `gic_opts()`
    - `flags()`

Heimdal Support

- Heimdal is growing a similar pre-auth plugin type.
  - Based on generic plugin interface support.
  - Partial support for client-side pre-auth plugins.
  - Has basic FAST functions in 1.3.
- Heimdal bundles an `ASN1` compiler.
- PKINIT support may be more mature.
  - Fewer configuration options.

Practical Issues

- Timeout Issues
  - A standard request to a KDC expects a response within 1 second.
    - If the KDC calls an external service for verification which in turn does other external verification then you can't avoid a client timeout.
      - Does a premature client retry cause another error?
      - Applies to both OTP and PKINIT (with OCSP).
  - Past attempts to support RSA SecurID tokens required tuning of RSA service timeout values to work.
    - RSA believes they should address some of these issues before they release a product based on the OTP draft RFC.
- Multi-pass mechanisms must maintain server context
  - FAST provides a state "cookie" (not yet implemented)

References

- Kerberos – `RFC 4120`
- `draft-ietf-krb-wg-preauth-framework (-16)`
- `draft-ietf-krb-wg-otp-preauth (-12)`