Kerberos WORKING GROUP	S. Hartman, Ed.
Internet-Draft	Painless Security
Updates: 4120 (if approved)	K. Raeburn
Intended status: Standards Track	MIT
Expires: May 03, 2012	L. Zhu
	Microsoft Corporation
	October 31, 2011

Kerberos Principal Name Canonicalization and KDC-Generated Cross-Realm Referrals
draft-ietf-krb-wg-kerberos-referrals-13

Abstract

The memo documents a method for a Kerberos Key Distribution Center (KDC) to respond to client requests for Kerberos tickets when the client does not have detailed configuration information on the realms of users or services. The KDC will handle requests for principals in other realms by returning either a referral error or a cross-realm TGT to another realm on the referral path. The clients will use this referral information to reach the realm of the target principal and then receive the ticket. This memo also provides a mechanism for verifying that a request has not been tampered with in transit.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Current implementations of the Kerberos AS and TGS protocols, as defined in [RFC4120], use principal names constructed from a known user or service name and realm. A service name is typically constructed from a name of the service and the DNS host name of the computer that is providing the service. Many existing deployments of Kerberos use a single Kerberos realm where all users and services would be using the same realm. However in an environment where there are multiple Kerberos realms, the client needs to be able to determine what realm a particular user or service is in before making an AS or TGS request. Traditionally this requires client configuration to make this possible. When having to deal with multiple realms, users are forced to know what realm they are in before they can obtain a ticket granting ticket (TGT) with an AS request. However, in many cases the user would like to use a more familiar name that is not directly related to the realm of their Kerberos principal name. A good example of this is an RFC 822 style email name. This document describes a mechanism that would allow a user to specify a user principal name that is an alias for the user's Kerberos principal name. In practice this would be the name that the user specifies to obtain a TGT from a Kerberos KDC. The user principal name no longer has a direct relationship with the Kerberos principal or realm. Thus the administrator is able to move the user's principal to other realms without the user having to know that it happened.
Once a user has a TGT, they would like to be able to access services in any Kerberos realm for which there is an authentication path from the realm of their principal. To do this requires that the client be able to determine what realm the target service principal is in before making the TGS request. Current implementations of Kerberos typically have a table that maps DNS host names to corresponding Kerberos realms. The user-supplied host name or its domain component is looked up in this table (often using the result of some form of host name lookup performed with insecure DNS queries, in violation of [RFC4120]). The corresponding realm is then used to complete the target service principal name. Even if insecure DNS queries were not used, managing this table is problematic.

This traditional mechanism requires that each client have very detailed configuration information about the hosts that are providing services and their corresponding realms. Having client side configuration information can be very costly from an administration point of view—especially if there are many realms and computers in the environment. This memo proposes a solution for these problems and simplifies administration by minimizing the configuration information needed on each computer using Kerberos. Specifically it describes a mechanism to allow the KDC to handle canonicalization of names, provide for principal aliases for users and services and allow the KDC to determine the trusted realm authentication path by being able to generate referrals to other realms in order to locate principals.
Two kinds of KDC referrals are introduced in this memo:

1. Client referrals, in which the client doesn't know which realm contains a user account.

2. Server referrals, in which the client doesn't know which realm contains a server account.

These two types of referrals introduce new opportunities for an attacker. In order to avoid these attacks, a mechanism is provided to protect the integrity of the request between the client and KDC. This mechanism compliments the Flexible Authentication through Secure Tunnels (FAST) facility provided in [RFC6113]. A mechanism is provided to negotiate the availability of FAST. Among other benefits this can be used to protect errors generated by the referral process.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Requesting a Referral

In order to request referrals as defined in later sections, the Kerberos client MUST explicitly request the canonicalize KDC option (bit 15) [RFC4120] for the AS-REQ or TGS-REQ. This flag indicates to the KDC that the client is prepared to receive a reply that contains a principal name other than the one requested.

```
KDCOptions ::= KerberosFlags
         -- canonicalize (15)
         -- other KDCOptions values omitted
```

The client should expect, when sending names with the "canonicalize" KDC option, that names in the KDC's reply MAY be different than the name in the request. A referral TGT is a cross realm TGT that is returned with the server name of the ticket being different from the server name in the request [RFC4120].

4. Realm Organization Model

This memo assumes that the world of principals is arranged on multiple levels: the realm, the enterprise, and the world. A KDC may issue tickets for any principal in its realm or cross-realm tickets for realms with which it has a direct cross-realm relationship. The KDC also has access to a trusted name service that can resolve any name from within its enterprise into a realm closer along the authentication path to the service. This trusted name service removes the need to use an un-trusted DNS lookup for name resolution.

For example, consider the following configuration, where lines indicate cross-realm relationships:

```
                  EXAMPLE.COM
                  /        \
                 /          \
      ADMIN.EXAMPLE.COM  DEV.EXAMPLE.COM
```

In this configuration, all users in the EXAMPLE.COM enterprise could have principal names such as alice@EXAMPLE.COM, with the same realm portion. In addition, servers at EXAMPLE.COM should be able to have DNS host names from any DNS domain independent of what Kerberos realm their principals reside in.

4.1. Trust Assumptions

Two realms participate in any cross-realm relationship: an issuing realm issues a cross-realm ticket and a consuming realm uses this ticket. There is a degree of trust of the issuing realm by the consuming realm implicit in this relationship. Whenever a service in the consuming realm permits an authentication path containing the issuing realm, that service trusts the issuing realm to accurately represent the identity of the authenticated principal and any information about the transited path. If the consuming realm's KDC sets the transited policy checked flag, the KDC is making the same trust assumption a service would.

This trust is transitive across a multi-hop authentication path. The service's realm trusts each hop along the authentication path closer to the client to accurately represent the authenticated identity and to accurately represent transited information. Any KDC along this path could impersonate the client.

KDC signed or issued authorization data often implies additional trust. The implications of such trust from a security and operational standpoint is an ongoing topic of discussion during the development of this specification. As such, such discussion is out of scope for this memo.

Administrators have several tools to limit trust caused by cross-realm relationships. A service or KDC can control what authentication paths are acceptable. For example if a given realm is not permitted on the authentication path for a particular client then that realm cannot affect trust placed in that client principal. Consuming realms can exercise significant control by deciding what principals to place on an access-control list. If no client using a given issuing realm in authentication paths is permitted to access a resource, then that issuing realm is not trusted in access decisions regarding that resource.

Creating a cross-realm relationship implies relatively little inherent trust in the issuing realm. Significant trust only applies as principals dependent on that issuing realm are given access to resources. However, two deployment constraints may imply significantly greater trust is implied by the initial cross-realm relationship. First, a number of realms provide access to any principal to some resources. Access decisions involving these resources involve a degree of trust in all issuing realms in the transited graph. Secondly, many realms do not significantly constrain what principals users of that realm may grant access. In these realms, creating a cross-realm relationship delegates the decision to trust that realm to users of the consuming realm. In this situation, creating the cross-realm relationship is the primary trust decision point under the administrator's control.

5. Enterprise Principal Name Type

The NT-ENTERPRISE type principal name contains one component, a string of realm-defined content, which is intended to be used as an alias for another principal name in some realm in the enterprise. It is used for conveying the alias name, not for the real principal names within the realms, and thus is only useful when name canonicalization is requested.

The intent is to allow unification of email and security principal names. For example, all users at EXAMPLE.COM may have a client principal name of the form "joe@EXAMPLE.COM" even though the principals are contained in multiple realms. This global name is again an alias for the true client principal name, which indicates what realm contains the principal. Thus, accounts "alice" in the realm DEV.EXAMPLE.COM and "bob" in ADMIN.EXAMPLE.COM may log on as "alice@EXAMPLE.COM" and "bob@EXAMPLE.COM".

This utilizes a new principal name type, as the KDC-REQ message only contains a single client realm field, and the realm portion of this name corresponds to the Kerberos realm with which the request is made. Thus, the entire name "alice@EXAMPLE.COM" is transmitted as a single component in the client name field of the AS-REQ message, with a name type of NT-ENTERPRISE [RFC4120] (and the local realm name). The KDC will recognize this name type and then transform the requested name into the true principal name if the client account resides in the local realm. The true principal name can have a name type different from the requested name type. Typically the true principal name will be a NT-PRINCIPAL [RFC4120].

6. Name Canonicalization

A service or account may have multiple principal names. For example, if a host is known by multiple names, host-based services on it may be known by multiple names in order to prevent the client from needing a secure directory service to determine the correct hostname to use. In order that the host should not need to be updated whenever a new alias is created, the KDC may provide the mapping information to the client in the credential acquisition process.

If the "canonicalize" KDC option is set, then the KDC MAY change the client and server principal names and types in the AS response and ticket returned from those in the request. Names MUST NOT be changed in the response to a TGS request, although it is common for KDCs to maintain ta set of aliases for service principals. Regardless of which alias a client requests, the same service key is used. However, in the TGS request, the client receives a ticket for whichever alias is requested. Services MUST NOT make distinctions based on which alias is in the issued ticket because the service name in a ticket is not cryptographically protected and can be changed by parties other than the KDC.

For example the AS request may specify a client name of "bob@EXAMPLE.COM" as an NT-ENTERPRISE name with the "canonicalize" KDC option set and the KDC will return with a client name of "104567" as a NT-UID.

(It is assumed that the client discovers whether the KDC supports the NT-ENTERPRISE name type via out of band mechanisms.)

See Section 11 for a mechanism to detect modification of the request between the client and KDC. However for best protection, Flexible Authentication through Secure Tunneling (FAST) [RFC6113] or another mechanism that protects the entire KDC exchange SHOULD be used. Clients MAY reject responses from a KDC where the client or server name is changed if the KDC does not support such a mechanism. Clients SHOULD reject an AS response that changes the server name unless the response is protected by such a mechanism or the new server name is one explicitly expected by the client. For example, many clients permit the realm name to be changed in an AS response even if the response is not protected. See Section 14 for a discussion of the tradeoffs in allowing unprotected responses.

In order to permit authorization decisions to be made based on aliases as well as the canonicalized form of a principal name, the KDC MAY include the following authorization data element, wrapped in AD-KDC-ISSUED, in the initial credentials and copy it from a ticket-granting ticket into additional credentials:

```
AD-LOGIN-ALIAS ::= SEQUENCE { -- ad-type number TBD --
  login-aliases  [0] SEQUENCE(1..MAX) OF PrincipalName,
}
```

The login-aliases field lists one or more of the aliases the principal is known by.

In addition te permitting authorization based on aliases, this permits user-to-user exchanges where the party receiving the authenticator knows the other party only by an alias. The recipient of such an authenticator SHOULD check the AD-LOGIN-ALIAS names, if present, in addition to the normal client name field, against the identity of the party with which it wishes to authenticate; either should be allowed to match. (Note that this is not backwards compatible with [RFC4120]; if the server side of the user-to-user exchange does not support this extension, and does not know the true principal name, authentication may fail if the alias is sought in the client name field.)

The use of AD-KDC-ISSUED authorization data elements in cross-realm cases has not been well explored at this writing; hence we will only specify the inclusion of this data in the one-realm case. The alias information SHOULD be dropped in the general cross-realm case. However, a realm MAY implement a policy of accepting and re-signing (wrapping in a new AD-KDC-ISSUED element) alias information provided by certain trusted realms, in the cross-realm ticket-granting service.

The canonical principal name for an alias MUST not be in the form of a ticket-granting service name, as (in a case of server name canonicalization) that would be construed as a case of cross-realm referral, described below.

7. Client Referrals

The simplest form of ticket referral is for a user requesting a ticket using an AS-REQ. In this case, the client machine will send the AS-REQ to a convenient realm trusted to map principals, for example the realm of the client machine. In the case of the name alice@EXAMPLE.COM, the client MAY optimistically choose to send the request to EXAMPLE.COM. The realm in the AS-REQ is always the name of the realm that the request is for as specified in [RFC4120].

The KDC will try to lookup the name in its local account database. If the account is present in the realm of the request, it SHOULD return a KDC reply structure with the appropriate ticket.

If the account is not present in the realm specified in the request and the "canonicalize" KDC option is set, the KDC may look up the client principal name using some kind of name service or directory service. If this lookup is unsuccessful, it MUST return the error KDC_ERR_C_PRINCIPAL_UNKNOWN [RFC4120]. If the lookup is successful, it MUST return an error KDC_ERR_WRONG_REALM [RFC4120] and in the error message the crealm field will contain either the true realm of the client or another realm that MAY have better information about the client's true realm. The client MUST NOT use the cname returned in this error message.

If the client receives a KDC_ERR_WRONG_REALM error, it will issue a new AS request with the same client principal name used to generate the first referral to the realm specified by the realm field of the Kerberos error message corresponding to the first request. (The client realm name will be updated in the new request to refer to this new realm.) The client SHOULD repeat these steps until it finds the true realm of the client. To avoid infinite referral loops, an implementation should limit the number of referrals. A suggested limit is 5 referrals before giving up.

Since the same client name is sent to the referring and referred-to realms, both realms must recognize the same client names. In particular, the referring realm cannot (usefully) define principal name aliases that the referred-to realm will not know.

The true principal name of the client, returned in AS-REQ, can be validated in a subsequent TGS message exchange where its value is communicated back to the KDC via the authenticator in the PA-TGS-REQ padata [RFC4120]. However, this requires trusting the referred-to realm's KDCs. Clients should limit the referral mappings they will accept to realms trusted via some local policy. Some possible factors that might be taken into consideration for such a policy might include:

- Any realm indicated by the local KDC, if the returned KRB-ERROR message is protected by some additional means, for example FAST

- A list of realms configured by an administrator

- Any realm accepted by the user when explicitly prompted

There is currently no provision for changing the client name in a client referral response.

8. Server Referrals

The primary difference in server referrals is that the KDC returns a referral TGT rather than an error message as is done in the client referrals.

If the "canonicalize" flag in the KDC options is set and the KDC doesn't find the principal locally, either as a regular principal or as an alias for another local principal, the KDC MAY return a cross-realm ticket granting ticket to the next hop on the trust path towards a realm that may be able to resolve the principal name.

The client will use this referral information to request a chain of cross-realm ticket granting tickets until it reaches the realm of the server, and can then expect to receive a valid service ticket.

However an implementation should limit the number of referrals that it processes to avoid infinite referral loops. A suggested limit is 5 referrals before giving up.

The client may cache the mapping of the requested name to the name of the next realm to use and the principal name to ask for. (See Section 10.)

Here is an example of a client requesting a service ticket for a service in realm DEV.EXAMPLE.COM where the client is in ADMIN.EXAMPLE.COM.

```
   +NC = Canonicalize KDCOption set
   C: TGS-REQ sname=http/foo.dev.example.com +NC to ADMIN.EXAMPLE.COM
   S: TGS-REP sname=krbtgt/EXAMPLE.COM@ADMIN.EXAMPLE.COM
   C: TGS-REQ sname=http/foo.dev.example.com +NC to EXAMPLE.COM
   S: TGS-REP sname=krbtgt/DEV.EXAMPLE.COM@EXAMPLE.COM
   C: TGS-REQ sname=http/foo.dev.example.com +NC to DEV.EXAMPLE.COM
   S: TGS-REP sname=http/foo.dev.example.com@DEV.EXAMPLE.COM
```

Note that any referral or alias processing of the server name in user-to-user authentication should use the same data as client name canonicalization or referral. Otherwise, the name used by one user to log in may not be useable by another for user-to-user authentication to the first.

9. Cross Realm Routing

RFC 4120 permits a KDC to return a closer referral ticket when a cross-realm TGT is requested. This specification extends this behavior when the canonicalize flag is set. When this flag is set, a KDC MAY return a TGT for a realm closer to the service for any service as discussed in the previous section. When a client follows such a referral, it including the realm of the referred-to realm in the generated request.

10. Caching Information

It is possible that the client may wish to get additional credentials for the same service principal, perhaps with different authorization-data restrictions or other changed attributes. The return of a server referral from a KDC can be taken as an indication that the requested principal does not currently exist in the local realm. Clearly, it would reduce network traffic if the clients could cache that information and use it when acquiring the second set of credentials for a service, rather than always having to re-check with the local KDC to see if the name has been created locally.

When the TGT expires, the previously returned referral from the local KDC should be considered invalid, and the local KDC must be asked again for information for the desired service principal name. (Note that the client may get back multiple referral TGTs from the local KDC to the same remote realm, with different lifetimes. The lifetime information SHOULD be properly associated with the requested service principal names. Simply having another TGT for the same remote realm does not extend the validity of previously acquired information about one service principal name.)

Accordingly, KDC authors and maintainers should consider what factors (e.g., DNS alias lifetimes) they may or may not wish to incorporate into credential expiration times in cases of referrals.

11. Negotiation of FAST and Detecting Modified Requests

Implementations of this specification MUST support the FAST negotiation mechanism described in this section. This mechanism provides detection of KDC requests modified by an attacker when those requests result in a reply instead of an error. In addition, this mechanism provides a secure way to detect if a KDC supports FAST.

Clients conforming to this specification MUST send a new pre-authentication data of type PA-REQ-ENC-PA-REP (TBD1) in all AS requests and MAY send this padata type in TGS requests. The value of this padata item SHOULD be empty and MUST be ignored by a receiving KDC. Sending this padata item indicates support for this negotiation mechanism. KDCs conforming to this specification must always set the ticket flag enc-pa-rep(15) in all the issued tickets. This ticket flag indicates KDC support for the mechanism.

```
       EncKDCRepPart   ::= SEQUENCE {
              key                 [0] EncryptionKey,
              last-req            [1] LastReq,
              nonce               [2] UInt32,
              key-expiration      [3] KerberosTime OPTIONAL,
              flags               [4] TicketFlags,
              authtime            [5] KerberosTime,
              starttime           [6] KerberosTime OPTIONAL,
              endtime             [7] KerberosTime,
              renew-till          [8] KerberosTime OPTIONAL,
              srealm              [9] Realm,
              sname              [10] PrincipalName,
              caddr              [11] HostAddresses OPTIONAL,
              encrypted-pa-data [12] SEQUENCE OF PA-DATA OPTIONAL
      }
```

The KDC response is extended to support an additional field containing encrypted pre-authentication data.

The The encrypted-pa-data element MUST be absent unless either the canonicalize KDC option is set or the PA-REQ-ENC-PA-REP padata item is sent.

If the PA-REQ-ENC-PA-REP padata item is sent in the request, then the KDC MUST include a PA-REQ-ENC-PA-REP padata item in the encrypted-pa-data item of any generated KDC reply. The PA-REQ-ENC-PA-REP pa-data value contains the checksum computed over the type AS-REQ or TGS-REQ in the request. The checksum key is the reply key and the checksum type is the required checksum type for the encryption type of the reply key, and the key usage number is KEY_USAGE_AS_REQ (56). If the KDC supports FAST, then the KDC MUST include a padata of type PA-FX-FAST in any encrypted-pa-data sequence it generates. The value for this padata item should be empty.

A client MUST reject a response for which it sent PA-REQ-ENC-PA-REP if the ENC-PA-REP ticket flag is set and the PA-REQ-ENC-PA-REP padata item is absent or the checksum is not successfully verified.

12. Number Assignments

Most number registries in the Kerberos protocol have not been turned over to IANA for management at the time of this writing, hence this is not an "IANA Considerations" section.

Various values do need assigning for this draft:

- AD-LOGIN-ALIAS

13. IANA Considerations

In the Kerberos pre-authentication and typed data registry at http://www.iana.org/assignments/kerberos-parameters/kerberos-parameters.xhtml#pre-authentication, the PA-REQ-ENC-PA-REP pa-data item should be registered. Because of existing implementations the value 149 is strongly preferred.

14. Security Considerations

For the AS exchange case, it is important that the logon mechanism not trust a name that has not been used to authenticate the user. For example, the name that the user enters as part of a logon exchange may not be the name that the user authenticates as, given that the KDC_ERR_WRONG_REALM error may have been returned. The relevant Kerberos naming information for logon (if any), is the client name and client realm in the service ticket targeted at the workstation that was obtained using the user's initial TGT. That is, rather than trusting the client name in the AS response, a workstation SHOULD perform an AP-REQ authentication against itself as a service and use the client name in the ticket issued for its service by the KDC.

How the client name and client realm is mapped into a local account for logon is a local matter, but the client logon mechanism MUST use additional information such as the client realm and/or authorization attributes from the service ticket presented to the workstation by the user, when mapping the logon credentials to a local account on the workstation.

Not all fields in an RFC 4120 KDC reply are protected. None of the fields in an RFC 4120 AS request are protected and some information in a TGS request may not be protected. The referrals mechanism creates several opportunities for attack because of these unprotected fields.

FAST [RFC6113] can be used to completely mitigate these issues by protecting both the KDC request and response. However, FAST requires that a client obtain an armor ticket before authenticating. Not all realms permit all clients to obtain armor tickets. Also, while it is expected to be uncommon, a client might wish to use name canonicalization while obtaining an armor ticket. The mechanism in Section 11 detects modification of the request between the KDC and client, mitigating some attacks.

There is a wide deployed base of implementations that use name canonicalization or server referrals that uses neither the negotiation mechanism nor FAST. So, implementations may be faced with only the limited protection afforded by RFC 4120, by the negotiation mechanism discussed in this document, or by FAST. All three situations are important to consider from a security standpoint.

An attacker cannot mount a downgrade attack against a client. The negotiation mechanism described in this document is securely indicated by the presence of a ticket flag. So, a client will detect if the facility was available but not used. It is possible for an attacker to strip the indication that a client supports the negotiation facility. The client will learn from the response that this happened, but the KDC will not learn that the client is attacked. So, for a single round-trip Kerberos exchange, the KDC may believe the exchange was successful when the client detects an attack. Packet loss or client failure can produce a similar result; this is not a significant vulnerability. The negotiation facility described in this document securely indicates the presence of FAST, so if a client wishes to use FAST when it is available, an attacker cannot force the client to downgrade away from FAST. An attacker MAY be able to prevent a client from obtaining an armor ticket, for example by responding to a request for anonymous PKINIT with an error response.

If FAST is used, then the communications between the client and KDC are protected. However name canonicalization places a new responsibility for mapping principals onto the KDC. This can increase the number of KDCs involved in an authentication which adds additional trusted third parties to the exchange.

If only the negotiation mechanism is used, then the request from the client to the KDC is protected, but not all of the response is protected. In particular, the client name is not protected; the ticket is also not protected. An attacker can potentially modify these fields. Modification of the client name will result in a denial of service. When the client attempts to authenticate to a service (including the TGS), it constructs an AP-REQ message. This message includes a client name which MUST match the client name in the ticket according to RFC 4120. Thus if the client name is changed, the resulting ticket will fail when used. This is undesirable because the authentication is separated from the later failure, which may confuse problem determination. If the ticket is replaced with another ticket, then later authentication to a service will fail because the client will not know the session key for the other ticket. If the ticket is simply modified, then authentication to a service will fail as with RFC 4120. More significant attacks are possible if a KDC violates the requirements of RFC 4120 and issues two tickets with the same session key or if a service violates the requirements of RFC 4120 and does not check the client name against that in the ticket.

There is an additional attack possible when FAST is not used against KDC_ERR_WRONG_REALM. Since this is an error response not an AS response, it is not protected by the negotiation mechanism. Thus, an attacker may be able to convince a client to authenticate to a realm other than the one intended. If an attacker is off-path this may give the attacker an advantage in attacking the client's credentials. Also, see the discussion of shared passwords below.

More serious attacks are possible if no protection beyond RFC 4120 is used. In this case, neither the client name nor the service name is protected between the client and KDC. In the general case, if an attacker changes the client name, then authentication will fail because the client will not have the right credentials (password, certificate , or other) to authenticate as the user selected by the attacker. However, see the discussion of shared passwords below. Changing the server name can be a very significant attack. For example if a user is authenticating in order to send some confidential information, then the attacker could gain this information by directing the user to a server under the attacker's control. The server name in the response is protected by RFC 4120, but not the one in the request. Fortunately, users are typically authenticating to the "krbtgt" service in an AS exchange. Clients that permit changes to the server name when no protection beyond RFC 4120 is in use SHOULD carefully restrict what service names are acceptable. One critical case to consider is the password changing service. When a user authenticates to change their password they use an AS authentication directly to the password changing service. Clients MUST restrict service name changes sufficiently that the client ends up talking to the correct password changing service.

14.1. Shared-password case

A special case to examine is when the user is known (or correctly suspected) to use the same password for multiple accounts. A man-in-the-middle attacker can either alter the request on its way to the KDC, changing the client principal name, or reply to the client with a response previously send by the KDC in response to a request from the attacker. The response received by the client can then be decrypted by the user, though if the default "salt" generated from the principal name is used to produce the user's key, a PA-ETYPE-INFO or PA-ETYPE-INFO2 preauth record may need to be added or altered by the attacker to cause the client software to generate the key needed for the message it will receive. None of this requires the attacker to know the user's password, and without further checking, could cause the user to unknowingly use the wrong credentials.

In normal [RFC4120] operation, a generated AP-REQ message includes in the Authenticator field a copy of the client's idea of its own principal name. If this differs from the name in the KDC-generated Ticket, the application server will reject the message.
With client name canonicalization as described in this document, the client may get its principal name from the response from the KDC. Using the wrong credentials may provide an advantage to an attacker. For example if a client uses one principal for administrative operations and one for less privileged operation, an attacker may coerce a client into using the wrong privilege to either cause some later operation to succeed or fail.

14.2. Preauthentication data

In cases of credential renewal, forwarding, or validation, if credentials are sent to the KDC that are not an initial ticket-granting ticket for the client's home realm, the encryption key used to protect the TGS exchange is one known to a third party (namely, the service for which the credential was issued). Consequently, in such an exchange, the protection described earlier may be compromised by the service. This is not generally believed to be a problem. If it is, some form of explicit TGS armor could be added to FAST.

15. Acknowledgments

John Brezak, Mike Swift, and Jonathan Trostle wrote the initial version of this document.
Karthik Jaganathan contributed to earlier versions.
Sam Hartman's work on this document was funded by the MIT Kerberos Consortium.

16. References

16.1. Normative References

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
[RFC4120]	Neuman, C., Yu, T., Hartman, S. and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.
[RFC6113]	Hartman, S. and L. Zhu, "A Generalized Framework for Kerberos Pre-Authentication", RFC 6113, April 2011.

16.2. Informative References

[RFC5280]	Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008.

[RFC4556]	Zhu, L. and B. Tung, "Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)", RFC 4556, June 2006.
[XPR]	Trostle, J, Kosinovsky, I and M Swift, "Implementation of Crossrealm Referral Handling in the MIT Kerberos Client", Network and Distributed System Security Symposium, February 2001, .

Appendix A. Compatibility with Earlier Implementations of Name Canonicalization

The Microsoft Windows 2000 and Windows 2003 releases included an earlier form of name-canonicalization [XPR]. Here are the differences:

```
       PA-SVR-REFERRAL-INFO       20

       PA-SVR-REFERRAL-DATA ::= SEQUENCE {
              referred-name   [1] PrincipalName OPTIONAL,
              referred-realm  [0] Realm
       }}
```

1) Windows include an additional encrypted padata element. The preauth data type definition in the encrypted preauth data is as follows:

> The referred-principal is never sent. The referred-realm is included in TGS replies and includes the realm name of the realm to which the client is referred. This information is redundant with the realm in the second component of the returned TGT.

2) When PKINIT ([RFC4556]) is used, the NT-ENTERPRISE client name is encoded as a Subject Alternative Name (SAN) extension [RFC5280] in the client's X.509 certificate. The type of the otherName field for this SAN extension is AnotherName [RFC5280]. The type-id field of the type AnotherName is id-ms-sc-logon-upn (1.3.6.1.4.1.311.20.2.3) and the value field of the type AnotherName is a KerberosString [RFC4120]. The value of this KerberosString type is the single component in the name-string [RFC4120] sequence for the corresponding NT-ENTERPRISE name type.

In Microsoft's current implementation through the use of global catalogs any domain in one forest is reachable from any other domain in the same forest or another trusted forest with 3 or less referrals. A forest is a collection of realms with hierarchical trust relationships: there can be multiple trust trees in a forest; each child and parent realm pair and each root realm pair have bidirectional transitive direct rusts between them.
While we might want to permit multiple aliases to exist and even be reported in AD-LOGIN-ALIAS, the Microsoft implementation permits only

one NT-ENTERPRISE alias to exist, so this question had not previously arisen.

Appendix B. Document history [REMOVE BEFORE PUBLICATION]

13 Better reflect that we are not solving the gnuftp.raeburn.org use case. Clean up other references to information in padata. Fix the Microsoft appendix based on discussions with them

12 Refactor to take advantage of FAST and new protected negotiation mechanism instead of providing our own. Simplify significantly based on this. Remove the true principal name support for now pending discussion in the WG. Add the new protected negotiation mechanism.

11 Changed title. Better protection on server referral preauth data. Support server name canonicalization. Rename ReferralInfo to ClientReferralInfo. Disallow alias mapping to a TGT principal. Explain why no name change in client referrals. Add empty IANA Considerations. Add some notes on preauth data protection during renewal etc.

10 Separate enterprise principal names into a separate section. Add a little wording to suggest server principal name canonicalization might be allowed; not fleshed out. Advise against AD-KDC-ISSUED in cronn-realm cases. Advise policy checks on returned client referral info, since there's no security. List number assignments. Add security analysis of shared-password case. No longer plan to remove Microsoft appendix. Add referral-valid-until field.

09 Changed to EXAMPLE.COM instead of using Morgan Stanley's domain. Rewrote description of existing practice. (Don't name the lookup table consulted. Mention that DNS "canonicalization" is contrary to [RFC4120].) Noted Microsoft behavior should be moved out into a separate document. Changed some second-person references in the introduction to identify the proper parties. Changed PA-CLIENT-CANONICALIZED to use a separate type for the actual referral data, add an extension marker to that type, and change the checksum key from the "returned session key" to the "AS reply key". Changed AD-LOGIN-ALIAS to contain a sequence of names, to be contained in AD-KDC-ISSUED instead of AD-IF-RELEVANT, and to drop the no longer needed separate checksum. Attempt to clarify the cache lifetime of referral information.

08 Moved Microsoft implementation info to appendix. Clarify lack of local server name canonicalization. Added optional authz-data for login alias, to support user-to-user case. Added requested-principal-name to ServerReferralData. Added discussion of caching information, and referral TGT lifetime.

07 Re-issued with new editor. Fixed up some references. Started history.

Authors' Addresses

Sam hartman editor Hartman Painless Security EMail: hartmans-ietf@mit.edu

Kenneth Raeburn Raeburn Massachusetts Institute of Technology EMail: raeburn@mit.edu

Larry Zhu Zhu Microsoft Corporation One Microsoft Way Redmond, WA 98052 US EMail: lzhu@microsoft.com