Microsoft Authorization Data Specification v. 1.0 for Microsoft Windows 2000 Operating Systems

April, 2000

Abstract

Microsoft Windows 2000 includes OS specific data in the Kerberos V5 authorization data field that is used for authorization as described in the Kerberos revisions Internet Draft [**1**]. This data is used for user logon and to create an access token. The access token is used by the system to enforce access checking when attempting to reference objects. This document describes the structure of the Windows 2000 specific authorization data that is carried in that field.

Top-Level PAC Structure

The PAC is generated by the KDC under the following conditions:

- during an AS request that has been validated with pre-authentication
- during a TGS request when the client has no PAC and the target is a service in the domain or a ticket granting service (referral ticket).

The PAC itself is included in the IF-RELEVANT (ID 1) portion of the authorization data in a ticket. Within the IF-RELEVANT portion, it is encoded as a KERB_AUTH_DATA_PAC with ID 128.

The PAC is defined as a C data type, with integers encoded in little-endian order. The PAC itself is made up of several layers. The outer structure, contained directly in the authorization data, is as follows. The top-level structure is the PACTYPE structure:

```
typedef unsigned long ULONG;
typedef unsigned short USHORT;
typedef unsigned long64 ULONG64;
typedef unsigned char UCHAR;

typedef struct _PACTYPE {
    ULONG cBuffers;
    ULONG Version;
    PAC_INFO_BUFFER Buffers[1];
} PACTYPE;
```

The fields are defined as follows:

cBuffers - contains the number of entries in the array Buffers
Version - this is version zero
Buffers - contains a conformant array of PAC_INFO_BUFFER structures

The PAC_INFO_BUFFER structure contains information about each piece of the PAC:

```
typedef struct _PAC_INFO_BUFFER {
    ULONG ulType;
    ULONG cbBufferSize;
    ULONG64 Offset;
} PAC_INFO_BUFFER;
```

Type fields are defined as follows:

ulType - contains the type of data contained in this buffer. For Windows 2000, it may be one of the following, which are explained further below:

```
#define PAC_LOGON_INFO                  1
#define PAC_CREDENTIAL_TYPE             2
#define PAC_SERVER_CHECKSUM             6
#define PAC_PRIVSVR_CHECKSUM            7
#define PAC_CLIENT_INFO_TYPE            10
```

Offset - contains the offset to the beginning of the data, in bytes, from the beginning of the PACTYPE structure. The data offset must by a multiple of 8. If the data pointed to by this field is complex, the data is typically NDR encoded. If the data is simple (indicating it includes no pointer types or complex structures) it is a little-endian format data structure.

PAC Credential Information

PAC_INFO_BUFFERs of type PAC_LOGON_INFO contain the credential information for the client of the Kerberos ticket. The data itself is contained in a KERB_VALIDATION_INFO structure, which is NDR encoded. The output of the NDR encoding is placed in the PAC_INFO_BUFFER structure of type PAC_LOGON_INFO.

```
typedef struct _KERB_VALIDATION_INFO {
    FILETIME LogonTime;
    FILETIME LogoffTime;
    FILETIME KickOffTime;
    FILETIME PasswordLastSet;
    FILETIME PasswordCanChange;
    FILETIME PasswordMustChange;
    UNICODE_STRING EffectiveName;
    UNICODE_STRING FullName;
    UNICODE_STRING LogonScript;
    UNICODE_STRING ProfilePath;
    UNICODE_STRING HomeDirectory;
    UNICODE_STRING HomeDirectoryDrive;
    USHORT LogonCount;
    USHORT BadPasswordCount;
    ULONG UserId;
    ULONG PrimaryGroupId;
    ULONG GroupCount;
    [size_is(GroupCount)] PGROUP_MEMBERSHIP GroupIds;
    ULONG UserFlags;
    ULONG Reserved[4];
    UNICODE_STRING LogonServer;
    UNICODE_STRING LogonDomainName;
    PSID LogonDomainId;
    ULONG Reserved1[2];
    ULONG UserAccountControl;
    ULONG Reserved3[7];
    ULONG SidCount;
    [size_is(SidCount)] PKERB_SID_AND_ATTRIBUTES ExtraSids;
    PSID ResourceGroupDomainSid;
    ULONG ResourceGroupCount;
    [size_is(ResourceGroupCount)] PGROUP_MEMBERSHIP ResourceGroupIds;
} KERB_VALIDATION_INFO;
```

The fields are defined as follows:

LogonTime - the time the client last logged on.

LogoffTime - the time at which the client's logon session should expire. If the logon session should not expire, this field should be set to (0x7fffffff,0xffffffff).

KickOffTime - the time at which the server should forcibly logoff the client. If the client should not be forced off, this field should be set to (0x7fffffff,0xffffffff). The ticket end time is a replacement for the KickOffTime. The service ticket lifetime will never be longer than the KickOffTime for a user.

PasswordLastSet - the time the client's password was last set. If it was never set, this field is zero.

PasswordCanChange - the time at which the client's password is allowed to change. If there is no restriction on when the client may change its password, this field should be set to the time of the logon.

PasswordMustChange - the time at which the client's password expires. If it doesn't expire, this field is set to (0x7fffffff,0xffffffff).

EffectiveName - This field contains the client's Windows 2000 UserName, stored in the Active Directory in the SamAccountName property. This field is optional. If left blank the length, maxlength and buffer are all zero.

FullName - this field contains the friendly name of the client, which is used only for display purpose and not security purposes. This field is optional. If left blank the length, maxlength and buffer are all zero.

LogonScript - This field contains the path to the client's logon script. This field is optional. If left blank the length, maxlength and buffer are all zero.

ProfilePath - This field contains the path to the client's profile. This field is optional. If left blank the length, maxlength and buffer are all zero.

HomeDirectory - This field contains the path to the client's home directory. It may be either a local path name or a UNC path name. This field is optional. If left blank the length, maxlength and buffer are all zero.

HomeDirectoryDrive - This field is only used if the client's home directory is a UNC path name. In that case, the share on the remote file server is mapped to the local drive letter specified by this field. This field is optional. If left blank the length, maxlength and buffer are all zero.

LogonCount - This field contains the count of how many times the client is currently logged on. This statistic is not accurately maintained by Windows 2000 and should not be used.

BadPasswordCount - This field contains the number of logon or password change attempts with bad passwords, since the last successful attempt.

* UserId - This field contains the relative Id for the client.

PrimaryGroupId - This field contains the relative ID for this client's primary group.

* GroupCount - This field contains the number of groups, within the client's domain, to which the client is a member.

* GroupIds - This field contains an array of the relative Ids and attributes of the groups in the client's domain of which the client is a member.

* UserFlags - This field contains information about which fields in this structure are valid. The two bits that may be set are indicated below. Having these flags set indicates that the corresponding fields in the KERB_VALIDATION_INFO structure are present and valid.

```
#define LOGON_EXTRA_SIDS                0x0020
#define LOGON_RESOURCE_GROUPS           0x0200
```

LogonServer - This field contains the NETBIOS name of the KDC which performed the AS ticket request.

LogonDomainName - This field contains the NETBIOS name of the client's domain.

* LogonDomainId - This field contains the SID of the client's domain. This field is used in conjunction with the UserId, PrimaryGroupId,and GroupIds fields to create the user and group SIDs for the client.

UserAccountControl - This fields contains a bitfield of information about the client's account. Valid values are:

```
#define USER_ACCOUNT_DISABLED                   (0x00000001)
#define USER_HOME_DIRECTORY_REQUIRED            (0x00000002)
#define USER_PASSWORD_NOT_REQUIRED              (0x00000004)
#define USER_TEMP_DUPLICATE_ACCOUNT             (0x00000008)
#define USER_NORMAL_ACCOUNT                     (0x00000010)
#define USER_MNS_LOGON_ACCOUNT                  (0x00000020)
#define USER_INTERDOMAIN_TRUST_ACCOUNT          (0x00000040)
#define USER_WORKSTATION_TRUST_ACCOUNT          (0x00000080)
#define USER_SERVER_TRUST_ACCOUNT               (0x00000100)
#define USER_DONT_EXPIRE_PASSWORD               (0x00000200)
#define USER_ACCOUNT_AUTO_LOCKED                (0x00000400)
#define USER_ENCRYPTED_TEXT_PASSWORD_ALLOWED    (0x00000800)
#define USER_SMARTCARD_REQUIRED                 (0x00001000)
#define USER_TRUSTED_FOR_DELEGATION             (0x00002000)
#define USER_NOT_DELEGATED                      (0x00004000)
#define USER_USE_DES_KEY_ONLY                   (0x00008000)
#define USER_DONT_REQUIRE_PREAUTH               (0x00010000)
```

* SidCount - This field contains the number of SIDs present in the ExtraSids field. This field is only valid if the LOGON_EXTRA_SIDS flag has been set in the UserFlags field.

* ExtraSids - This field contains a list of SIDs for groups to which the user is a member. This field is only valid if the LOGON_EXTRA_SIDS flag has been set in the UserFlags field.

* ResouceGroupCount - This field contains the number of resource groups in the ResourceGroupIds field. This field is only valid if the LOGON RESOURCE_GROUPS flag has been set in the UserFlags field._

* ResourceGroupDomainSid - This field contains the SID of the resource domain. This field is used in conjunction with the ResourceGroupIds field to create the group SIDs for the client.

* ResourceGroupIds - This field contains an array of the relative Ids and attributes of the groups in the resource domain of which the resource is a member.

Fields marked with a '*' are used in the NT token.

When used in the KERB_VALIDATION_INFO, this is NDR encoded. The FILETIME type is defined as follows:

```
typedef unsigned int DWORD;

typedef struct _FILETIME {
    DWORD dwLowDateTime;
    DWORD dwHighDateTime;
} FILETIME;
```

Times are encoded as the number of 100 nanosecond increments since January 1, 1601, in UTC time.

When used in the KERB_VALIDATION_INFO, this is NDR encoded. The UNICODE_STRING structure is defined as:

```
typedef struct _UNICODE_STRING
    USHORT Length;
    USHORT MaximumLength;
    [size_is(MaximumLength / 2), length_is((Length) / 2) ] USHORT *  Buffer;
} UNICODE_STRING;
```

The Length field contains the number of bytes in the string, not including the null terminator, and the MaximumLength field contains the total number of bytes in the buffer containing the string.

The GROUP_MEMBERSHIP structure contains the relative ID of a group and the corresponding attributes for the group.

```
typedef struct _GROUP_MEMBERSHIP {
    ULONG RelativeId;
    ULONG Attributes;
} *PGROUP_MEMBERSHIP;
```

The group attributes must be:

```
#define SE_GROUP_MANDATORY              (0x00000001L)
#define SE_GROUP_ENABLED_BY_DEFAULT     (0x00000002L)
#define SE_GROUP_ENABLED                (0x00000004L)
```

The SID structure is defined as follows:

```
typedef struct _SID_IDENTIFIER_AUTHORITY {
    UCHAR Value[6];
} SID_IDENTIFIER_AUTHORITY, *PSID_IDENTIFIER_AUTHORITY;
```

The constant value for the NT Authority is:

```
#define SECURITY_NT_AUTHORITY           {0,0,0,0,0,5}

typedef struct _SID {
   UCHAR Revision;
   UCHAR SubAuthorityCount;
```

```
    SID_IDENTIFIER_AUTHORITY IdentifierAuthority;
    [size_is(SubAuthorityCount)] ULONG SubAuthority[*];
} SID, *PSID;
```

The SubAuthorityCount field contains the number of elements in the actual SubAuthority conformant array. The maximum number of subauthorities allowed is 15.

The KERB_SID_AND_ATTRIBUTES structure contains entire group SIDs and their corresponding attributes:

```
typedef struct _KERB_SID_AND_ATTRIBUTES {
    PSID Sid;
    ULONG Attributes;
} KERB_SID_AND_ATTRIBUTES, *PKERB_SID_AND_ATTRIBUTES;
```

The attributes are the same as the group attributes defined above.

Client Information

The client information is included in the PAC to allow a server to verify that the PAC in a ticket is applicable to the client of the ticket, which prevents splicing of PACs between tickets. The PAC_CLIENT_INFO structure is included in a PAC_INFO_BUFFER of type PAC_CLIENT_INFO_TYPE.

```
typedef struct _PAC_CLIENT_INFO {
    FILETIME ClientId;
    USHORT NameLength;
    WCHAR Name[1];
} PAC_CLIENT_INFO, *PPAC_CLIENT_INFO;
```

The fields are defined as follows:

ClientId - This field contains a conversion of the AuthTime field of the ticket into a FILETIME structure.

NameLength - This field contains the length, in bytes, of the Name field.

Name - This field contains the client name from the ticket, converted to Unicode and encoded using "/" to separate parts of the client principal name with an "@" separating the client principal name from the realm name. The string is not null terminated.

Supplemental Credentials

The KDC may return supplemental credentials in the PAC as well. Supplemental credentials are data associated with a security package that is private to that package. They can be used to return an appropriate user key that is specific to that package for the purposes of authentication. Supplemental creds are only used in conjunction with PKINIT[**2**]. Supplemental credentials are always encrypted using the client key. The PAC_CREDENTIAL_DATA structure is NDR encoded and

then encrypted with the key used to encrypt the KDC's reply to the client. The PAC_CREDENTIAL_INFO structure is included in PAC_INFO_BUFFER of type PAC_CREDENTIAL_TYPE.

Supplemental credentials for a single package are NDR encoded as follows:

```
typedef struct _SECPKG_SUPPLEMENTAL_CRED {
    UNICODE_STRING PackageName;
    ULONG CredentialSize;
    [size_is(CredentialSize)]PUCHAR Credentials;
} SECPKG_SUPPLEMENTAL_CRED, *PSECPKG_SUPPLEMENTAL_CRED;
```

The fields in this structure are defined as follows:

PackageName - This field contains the name of the package for which credentials are presented.
CredentialSize - This field contains the length, in bytes, of the presented credentials.
Credentials - This field contains a pointer to the credential data.

The set of all supplemental credentials is NDR encoded in a PAC_CREDENTIAL_DATA structure:

```
typedef struct _PAC_CREDENTIAL_DATA {
    ULONG CredentialCount;
    [size_is(CredentialCount)] SECPKG_SUPPLEMENTAL_CRED Credentials[*];
} PAC_CREDENTIAL_DATA, *PPAC_CREDENTIAL_DATA;
```

The fields are defined as follows:
CredentialCount - This field contains the number of credential present in the Credentials array.
Credentials - This field contains an array of the presented supplemental credentials.

The PAC_CREDENTIAL_DATA structure is NDR encoded and then encrypted with the key used to encrypt the KDC reply. The resulting buffer is returned in the following structure:

```
typedef struct _PAC_CREDENTIAL_INFO {
    ULONG Version;
    ULONG EncryptionType;
    UCHAR Data[1];
} PAC_CREDENTIAL_INFO, *PPAC_CREDENTIAL_INFO;
```

The fields are defined as follows:

Version - This field contains the version field of the key used to encrypt the data, or zero if the field is not present.

EncryptType - This field contains the encryption type used to encrypt the data. The encryption type uses the same values as the defined encryptions types for Kerberos [**1**].

Data - This field contains an array of bytes containing the encrypted supplemental credential data.

Signatures

The PAC contains two digital signatures: one using the key of the server, and one using the key of the KDC. The signatures are present for two reasons. First, the signature with the server's key is present to prevent a client from generating their own PAC and sending it to the KDC as encrypted authorization data to be included in tickets. Second, the signature with the KDC's key is present to prevent an untrusted service from forging a ticket to itself with an invalid PAC. The two signatures are sent in PAC_INFO_BUFFERs of type PAC_SERVER_CHECKSUM and PAC_KDC_CHECKSUM respectively.

The signatures are contained in the following structure:

```
typedef struct _PAC_SIGNATURE_DATA {
    ULONG SignatureType;
    UCHAR Signature[1];
} PAC_SIGNATURE_DATA, *PPAC_SIGNATURE_DATA;
```

The fields are defined as follows:

SignatureType - This field contains the type of checksum used to create a signature. The checksum must be a keyed checksum.

Signature - This field consists of an array of bytes containing the checksum data. The length of bytes may be determined by the wrapping PAC_INFO_BUFFER structure.

For the server's checksum, the key used to generate the signature should be the same key used to encrypt the ticket. Thus, if the enc_tkt_in_skey option is used, the session key from the server's TGT should be used. The Key used to encrypt ticket-granting tickets is used to generate the KDC's checksum.

The checksums are computed as follows:

1. The complete PAC is built, including space for both checksums
2. The data portion of both checksums is zeroed.
3. The entire PAC structure is checksummed with the server's key, and the result is stored in the server's checksum structure.
4. The server's checksum is then checksummed with the KDC's key.
5. The checksum with the KDC key is stored in the KDC's checksum structure.

PAC Request Pre-Auth Data

Normally, the PAC is included in every pre-authenticated ticket received from an AS request. However, a client may also explicitly request either to include or to not include the PAC. This is done by sending the PAC-REQUEST preauth data.

```
KERB-PA-PAC-REQUEST ::= SEQUENCE {
    include-pac[0] BOOLEAN -- if TRUE, and no PAC present,
                           -- include PAC.
                           ---If FALSE, and PAC
                           -- present, remove PAC
}
```

The fields are defined as follows:

include-pac - This field indicates whether a PAC should be included or not. If the value is TRUE, a PAC will be included independent of other preauth data. If the value is FALSE, then no PAC will be included, even if other preauth data is present.

The preauth ID is:

```
#define KRB5_PADATA_PAC_REQUEST          128
```

References

1 Neuman, C., Kohl, J., Ts'o, T., "The Kerberos Network Authentication Service (V5)", draft-ietf-cat-kerberos-revisions-05.txt, March 10, 2000

2 Tung, B., Hur, M., Medvinsky, A., Medvinsky, S., Wray, J., Trostle, J., " Public Key Cryptography for Initial Authentication in Kerberos", draft-ietf-cat-kerberos-pk-init-11.txt, March 15, 2000

Legal Notice

This Specification is provided to you pursuant to the terms and conditions of the Agreement for Microsoft Authorization Data Specification v. 1.0 for Microsoft Windows 2000 Operating Systems (the "Agreement") for the sole purpose of allowing you to review the Specification for security analysis, as further specified in the Agreement. If you have not downloaded the Specification from Microsoft's website and agreed to the terms and conditions of the Agreement, you are not an authorized licensee of the Specification.

For your reference, the Agreement is reproduced below.

Agreement for Microsoft Authorization Data Specification v. 1.0 for Microsoft Windows 2000 Operating Systems

IMPORTANT—READ CAREFULLY: This Microsoft Agreement ("**Agreement**") is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation ("**Microsoft**") for the version of the Microsoft specification identified above which you are about to download ("**Specification**"). **BY DOWNLOADING, COPYING OR OTHERWISE USING THE SPECIFICATION, YOU AGREE TO BE BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, DO NOT DOWNLOAD, COPY, OR USE THE SPECIFICATION.**

The Specification is owned by Microsoft or its suppliers and is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties.

1. **LICENSE**.

(a) Provided that you comply with all terms and conditions of this Agreement, including without limitation subsections (b)-(d) below, Microsoft grants to you the following non-exclusive, worldwide, royalty-free, non-transferable, non-sublicenseable license, under any copyrights or trade secrets owned or licensable by Microsoft without payment of consideration to unaffiliated third parties, to reproduce and use a reasonable number of copies of the Specification in its entirety for the sole purpose of reviewing the Specification for security analysis. By way of clarification of the foregoing, the Specification is provided to you solely for your informational purposes (for review as specified above) and, pursuant to this Agreement, Microsoft does not grant you any right to implement this Specification.

(b) The Specification is confidential information and a trade secret of Microsoft. Therefore, you may not disclose the Specification to anyone else (except as specifically allowed below), and you must take reasonable security precautions, at least as great as the precautions you take to protect your own confidential information, to keep the Specification confidential. If you are an entity, you may disclose the Specification to your full-time employees on a need to know basis, provided that you have executed appropriate written agreements with your employees sufficient to enable you to comply with the terms of this Agreement. You are also permitted to discuss the Specification with anyone else who has downloaded the Specification and agreed to these terms and conditions.

(c) You may not remove any of the copyright notices or other legends from any copy of the Specification.

(d) Microsoft reserves all other rights it may have in the Specification and any intellectual property therein. Microsoft may have patents or pending patent applications, trademarks, copyrights, trade secrets or other intellectual property rights covering subject matter in the Specification. The furnishing of this Specification does not give you any license to these patents, trademarks, trade secrets, copyrights, or other intellectual property rights, except as specifically set forth in subsection (a) above with respect to certain copyrights and trade secrets.

2. **ADDITIONAL LIMITATIONS**.

(a) The foregoing license is applicable only to the version of the Specification which you are about to download. It does not apply to any additional versions of or extensions to the Specification.

(b) Without prejudice to any other rights, Microsoft may terminate this Agreement if you fail to comply with its terms and conditions. In such event you must destroy all copies of the Specification in your possession or under your control.

3. **INTELLECTUAL PROPERTY RIGHTS**. All ownership, title and intellectual property rights in and to the Specification are owned by Microsoft or its suppliers.

4. **DISCLAIMER OF WARRANTIES. To the maximum extent permitted by applicable law, Microsoft and its suppliers provide the Specification (and all intellectual property therein) AS IS AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties or conditions of merchantability, of fitness for a particular purpose, and of accuracy or completeness, all with regard to the Specification and any intellectual property therein. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION AND ANY INTELLECTUAL PROPERTY THEREIN.**

5. **EXCLUSION OF DIRECT, INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY DIRECT, SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR FOR BUSINESS INTERRUPTION) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SPECIFICATION, ANY INTELLECTUAL PROPERTY THEREIN, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS AGREEMENT, EVEN IF MICROSOFT OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.**

6. **LIMITATION OF LIABILITY AND REMEDIES. Notwithstanding any damages that you might incur for any reason whatsoever, the entire liability of Microsoft and any of its suppliers under any provision of this Agreement and your exclusive remedy for all of the foregoing shall be limited to the greater of the amount actually paid by you for the Specification or U.S.$5.00. The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails its essential purpose.**

7. **APPLICABLE LAW**. This Agreement is governed by the laws of the State of Washington.

8. **ENTIRE AGREEMENT.** This Agreement is the entire agreement between you and Microsoft relating to the Specification and it supersedes all prior or contemporaneous oral or written communications, proposals and representations with respect to the Specification.