



Intel® Active Management Technology
Integration with Microsoft Windows*
Active Directory

Version 3.0.2, January 2007

Information in this document is provided in connection with Intel products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

The API and software may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document and the software described in it are furnished under license and may only be used or copied in accordance with the terms of the license. The information in this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Corporation. Intel Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document. Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Corporation.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an ordering number and are referenced in this document or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at <http://www.intel.com>.

Copyright © 2006, 2007 Intel Corporation.

* Third party other names and brands may be claimed as the property of others.

1 Introduction

Intel® Active Management Technology, starting with Release 2.0 provides for a standard and single-sign-on style of authentication by integrating its authentication framework with Microsoft Windows* Active Directory. Active Directory manages domain authentication based on the Kerberos protocol.

Authentication to Intel AMT integrated with Microsoft Windows domain authentication eliminates the need for ISV applications (including setup and configuration services) to manage unique and strong username/password pairs for all Intel AMT systems. Authentication to Intel AMT is as strong and as secure as authentication to the Windows domain; and administrators wanting to manage Intel AMT systems need only to login to the Windows domain to gain access to Intel AMT devices.

This document provides an overview of Kerberos, a description of the salient features of Active Directory, and the steps required to integrate Intel AMT with Active Directory. It includes an example of setting up Active Directory and the Sample Configuration Application to configure an Intel AMT device to work with Kerberos.

2 Glossary

Term	Definition
Access Control List	In Intel AMT, a list of users and their access privileges.
Active Directory (AD)	A widely-used Microsoft server directory service provides the means to manage the identities and relationships that make up network environments.
Authentication	A security measure designed to establish the validity of a transmission, message, or originator.
Authentication Server (AS)	A Kerberos element in a KDC that recognizes a client at log-on time based on information in its trusted database.
Authenticator	An authentication protocol string created each time authentication occurs and sent with the ticket to the server. It contains a time-stamp encrypted in the session key that can reliably show that the authentication request actually came from the client identified in the ticket.
Authorization	The process of determining what types of activities are permitted. Usually, authorization is in the context of authentication: once you have authenticated a user, the user may be authorized for different types of access or activity.
Domain	Part of the DNS (domain naming system) name that specifies details about a host. A domain is the main subdivision of Internet addresses, the last three letters after the final dot, and it tells you what kind of organization you are dealing with. In the context of Active Directory, every host is a member of a domain. A user logs in to the domain of which he is a member.
FQDN	Fully qualified domain name: the human-readable name corresponding to the TCP/IP address of a network interface, as found on a computer, router or other networked device. It includes both its

Term	Definition
	host name and its domain name.
Group	In Active Directory, a collection of users and objects that share properties and permissions. A group may have another group as a member. The second group is then a sub-group of the first group.
GSS-API	Generic Security Services Application Programming Interface. The generic API for performing client-server authentication.
ISV	Independent Software Vendors that develop applications that use the Intel AMT capabilities.
Kerberized application	A client or server application that supports the Kerberos protocol.
Kerberos	An authentication protocol that depends on a trusted third party (named after a mythological creature).
Key Distribution Center (KDC)	In the Kerberos protocol, a trusted third party that has secret information (passwords) for all clients and services under its supervision.
proxy	A firewall mechanism that replaces the IP address of a host on the internal (protected) network with its own IP address for all traffic passing through it. A software agent that acts on behalf of a user, typical proxies accept a connection from a user, make a decision as to whether or not the user or client IP address is permitted to use the proxy, perhaps does additional authentication, and then completes a connection on behalf of the user to a remote destination.
RC4-HMAC	An encryption type based on the RC4 encryption algorithm that uses an MD5 HMAC for checksum. It is included in the Windows implementation of Kerberos.
Realm	<p>In Kerberos, a realm is the same as an Active Directory domain. Kerberos V5 expects realms to have all capital letters.</p> <p>Intel AMT functionality is divided among different realms, for example, the Storage Realm and the Storage Administration Realm. ACLs associate a user or an SID with one or more realms.</p>
Schema	The Microsoft Active Directory schema contains formal definitions of every object class that can be created. One of these objects is the computer object. The Intel AMT object is based on the computer object.
Security Identifier (SID)	A numeric value that identifies a logged-on user who has been authenticated by Active Directory or a user group.
SOAP	Simple Object Access Protocol defines the use of XML and HTTP to access services, objects, and servers in a platform-independent manner.
SOL/IDE-R	Serial-over-LAN/IDE-Redirect: The proprietary protocols defined for Intel AMT for redirecting keyboard/text or floppy disk/CD transfers from a local host to a remote workstation.
SPEGNO	S imple and P rotected GSS-API N egotiation Mechanism. SPNEGO is a standard GSS-API pseudo-mechanism for peers to determine which GSS-API mechanisms are shared, select one and then establish a security context with it.

Term	Definition
SPN	A service principal name - the name by which a client uniquely identifies an instance of a service.
TCP/IP	Transmission Control Protocol/Internet Protocol: A protocol for communication between computers, used as a standard for transmitting data over networks and as the basis for standard Internet protocols.
Ticket Granting Server (TGS)	A Kerberos element in a KDC that creates tickets used to by clients to access servers.
TLS	Transport Layer Security. A protocol intended to secure and authenticate communications across a public networks by using data encryption.
Token	In Kerberos, a fixed length element that contains a user's SID and includes the user's rights and group memberships.

3 Introduction to Kerberos Authentication

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It has the following characteristics:

- It is secure: it never sends a password unless it is encrypted.
- Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.
- The concept depends on a trusted third party – a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.
- It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client.

Kerberos introduces the concept of a Ticket-Granting Server (TGS). A client that wishes to use a service has to receive a ticket – a time-limited cryptographic message – giving it access to the server. Kerberos also requires an Authentication Server (AS) to verify clients. The two servers combined make up a KDC. Active Directory performs the functions of the KDC. Figure 1 below shows the sequence of events required for a client to gain access to a service using Kerberos authentication. Each step is shown with the Kerberos message associated with it, as defined in RFC 4120 “The Kerberos Network Authorization Service (V5)”.

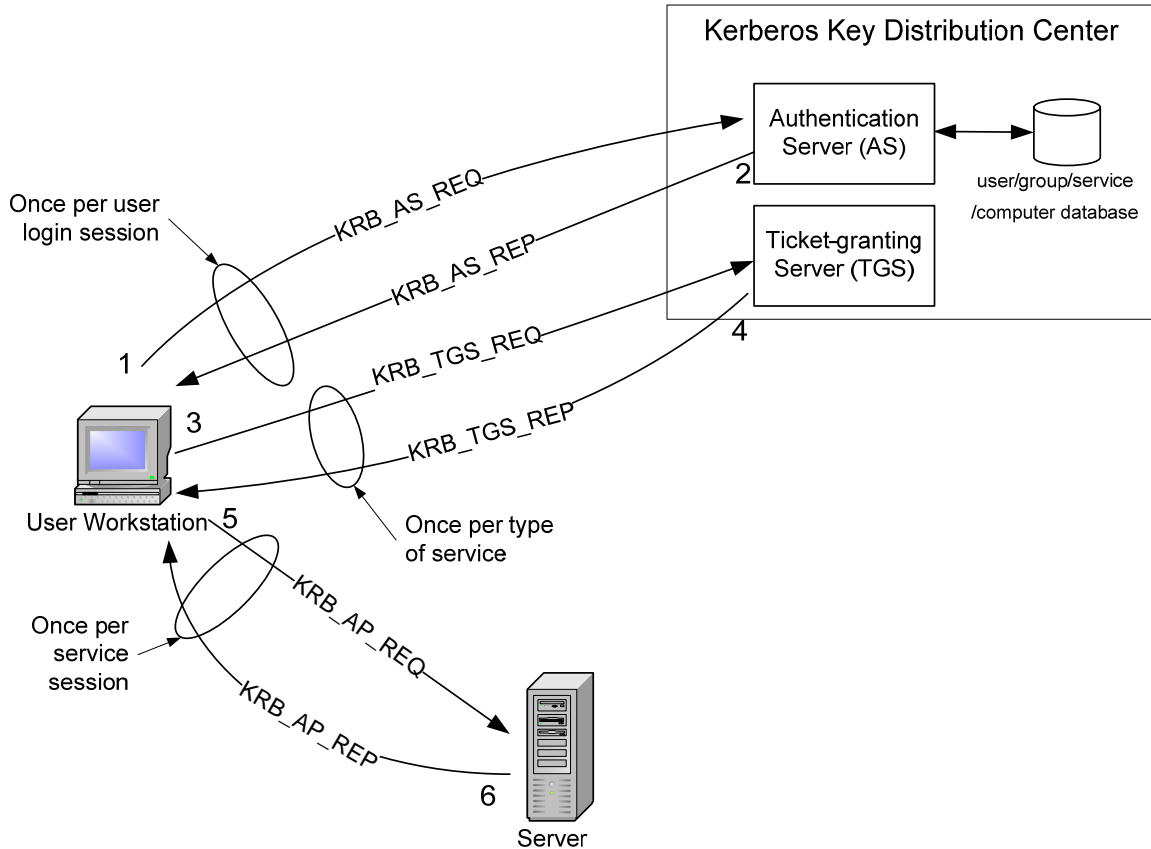


Figure 1. Kerberos Authentication Sequence

Step 1: The user logs on to the workstation and requests service on the host. The workstation sends a message to the Authorization Server requesting a ticket granting ticket (TGT).

Step 2: The Authorization Server verifies the user's access rights in the user database and creates a TGT and session key. The Authorization Server encrypts the results using a key derived from the user's password and sends a message back to the user workstation.

The workstation prompts the user for a password and uses the password to decrypt the incoming message. When decryption succeeds, the user will be able to use the TGT to request a service ticket.

Step 3: When the user wants access to a service, the workstation client application sends a request to the Ticket Granting Service containing the client name and realm name and a timestamp. The user proves his identity by sending an authenticator encrypted with the session key received in Step 2.

Step 4: The TGS decrypts the ticket and authenticator, verifies the request, and creates a ticket for the requested server. The ticket contains the client name and optionally the client IP address. It also contains the realm name and ticket lifespan. The TGS returns the ticket to the user workstation. The returned message contains two copies of a server session key – one encrypted with the client password, and one encrypted by the service password.

Step 5: The client application now sends a service request to the server containing the ticket received in Step 4 and an authenticator. The service authenticates the request by decrypting the session key. The server verifies that the ticket and authenticator match, and then grants access to the service. This step as described does not include the authorization performed by the Intel AMT device, as described later.

Step 6: If mutual authentication is required, then the server will reply with a server authentication message.

The Kerberos server knows "secrets" (encrypted passwords) for all clients and servers under its control, or it is in contact with other secure servers that have this information. These "secrets" are used to encrypt all of the messages shown in the figure above.

In order to prevent "replay attacks," Kerberos uses timestamps as part of its protocol definition. For timestamps to work properly, the clocks of the client and the server need to be in synch as much as possible. In other words, both computers need to be set to the same time and date. Because the clocks of two computers are often out of synch, administrators can establish a policy to establish the maximum acceptable difference to Kerberos between a client's clock and server's clock. If the difference between a client's clock and the server's clock is less than the maximum time difference specified in this policy, any timestamp used in a session between the two computers will be considered authentic. The maximum difference is usually set to five minutes.

Note that if a client application wishes to use a service that is "Kerberized" (the service is configured to perform Kerberos authentication), the client must also be Kerberized so that it expects to support the necessary message responses.

For more information about Kerberos, see <http://web.mit.edu/kerberos/www/>.

4 Microsoft Active Directory and Kerberos Support

Microsoft Active Directory provides Kerberos-based mutual authentication, along with other security services that ease implementation of a network that supports Kerberized clients and servers.

These include:

- Single Sign On
- Windows authorization
- A database of users, computers and network devices
- Multiple domains

Active Directory allows a user to enter a password once per session and uses the Windows group-based authorization approach to define user privileges.

4.1 User Identification within Active Directory

Users are added to Active Directory by an administrator. Users are added to Groups, which, in turn, have privileges in the form of access to services. A group can be a sub-group of another group. Each user, group, sub-group, and service has a unique Security Identifier (SID). SIDs are never reused, so uniqueness is guaranteed indefinitely.

For example, an administrator group defines a group called PlatformMgt. All users who are allowed to manage Intel AMT platforms are added to this group. Application software (.e.g., an ISV Management Console application) is used to add the SID of PlatformMgt to the relevant Access Control List in all Intel AMT devices to be managed by this group. The Intel AMT device saves the ACL in non-volatile memory. When a user attempts to access a platform, Intel AMT verifies that the user's SID is in its ACL before granting the user access. Intel AMT does not use Active Directory directly for authentication or authorization.

4.2 Kerberos Authentication with Active Directory

Once Active Directory "knows" each user, each platform (client and server platforms), and each service, then the following sequence is followed to connect a client to a service:

- Each platform authenticates with Active Directory after it is booted.

- Each user authenticates with Active Directory by logging on only once with a user ID and password. The user receives a “Ticket Granting Ticket” that contains
 - The user’s identity
 - SIDs that the user is allowed to access based on the groups and sub-groups that the user is in.
- The ticket is good for the duration of the logon session or typically up to eight hours.
- The user authenticates to a service by supplying the Ticket Granting Ticket to the Ticket Granting Server.
 - The Ticket Granting Server validates the ticket, and, if the user has the privilege of accessing the requested service, according to Active Directory criteria, the Ticket Granting Server returns a ticket with the user’s SID and all the SIDs to which he is a member.
 - The user’s client application sends the ticket to the service.
 - The service validates the ticket and sends back an authenticator where mutual authentication is required.

The validation and authorization sequence performed between a client and an Intel AMT server is summarized in Figure 2, below.

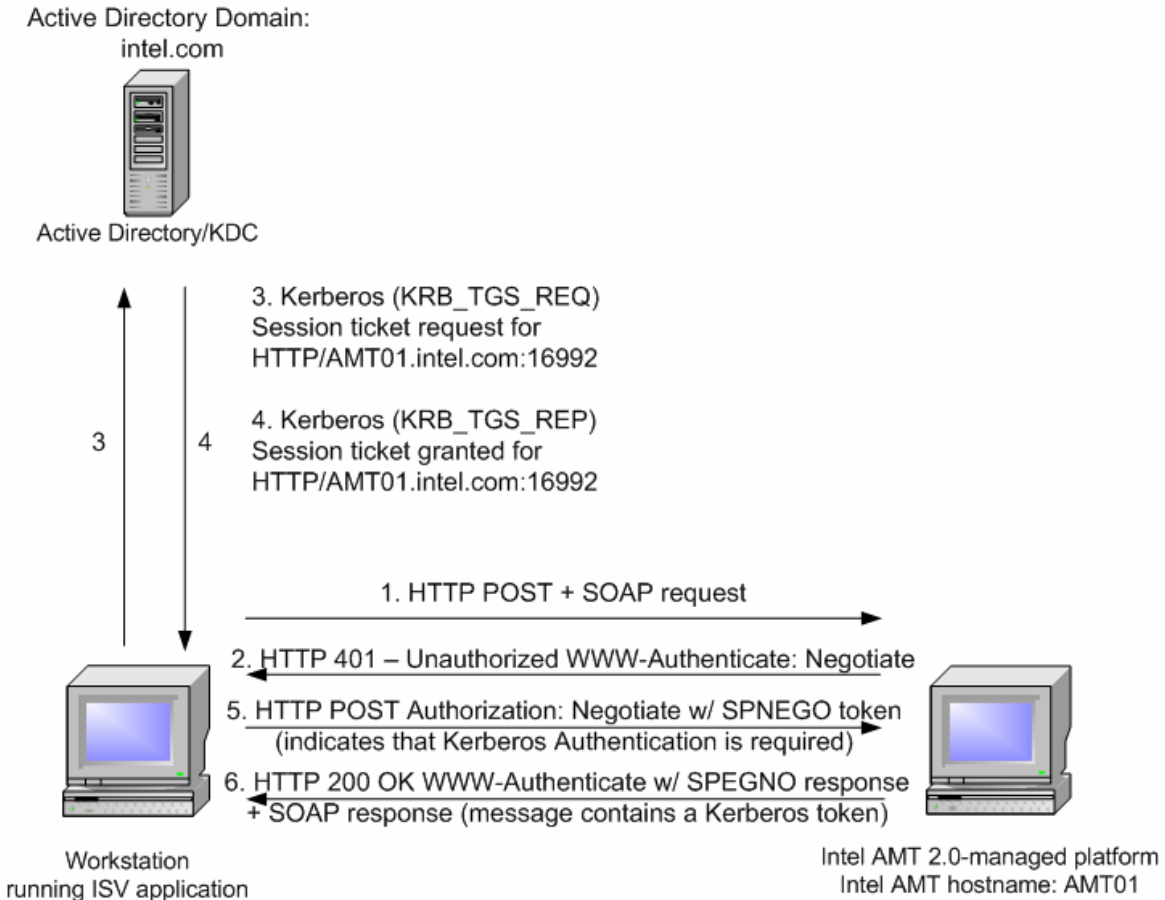


Figure 2. Negotiation Message Sequence

1. The ISV application sends an HTTP POST SOAP request message to an Intel AMT device named “AMT01”.
2. The device responds with a HTTP 401 “Unauthorized” Authenticate/negotiate message.

3. The application sends Kerberos request to access AMT01 with the SPN embedded in it. The SPN in this case is HTTP/AMT01.intel.com:16992 (the SOAP over HTTP service to the selected device).
4. The Active Directory KDC responds a ticket containing the application user's accesses (SIDs).
5. The application workstation sends a POST message with a Negotiate response, indicating that a Kerberos authentication is required. At this point, AMT01 can authenticate the application user and authorize access if the user's SID is in AMT01's ACL for the Realm requested.
6. AMT01 returns an HTTP 200 OK to authenticate with the application platform.

5 Using Active Directory to Manage Intel AMT Devices

Intel AMT supports the Kerberos option based on the following standards:

- Kerberos V5 (RFC 1510)
- GSS-API (RFC 1964)
- SPNEGO (RFC 2578)

Intel AMT supports the RC4-HMAC cipher suite.

Intel AMT is a Kerberized UNIX service from the point of view of Active Directory. Each device registers with Active Directory and provides four Service Principal Names (SPNs) for the four services it provides:

SPN	Service
HTTP/FQDN:16992	SOAP over HTTP
HTTP/FQDN:16993	SOAP over HTTPS
HTTP/FQDN:16994	Redirection over TCP
HTTP/FQDN:16995	Redirection over TLS

The SOAP SPNs support all of the Intel AMT functionality that uses SOAP over HTTP or HTTPS for remote communications. However, see [Notes and Limitations](#), below.

The Intel AMT redirection functionality communicates using TCP/IP with or without TLS.

Each Intel AMT device is recorded in the Active Directory database as an Intel AMT object, which is defined as an Active Directory computer object with the version of Intel AMT linked to it. The Intel AMT device hostname makes the entry unique. Active Directory uses the Intel AMT device password to create the device secret.

The Sample Configuration Application performs this function in a simplified way by registering each Intel AMT device as a user with the associated SPNs. The Intel AMT Setup and Configuration Server provides scripts for extending the Active Directory schema, and creates AMT objects for all configured Intel AMT devices.

The Intel AMT device maintains an Access Control List (ACL) of those users that can access Realms within the device. When a Management Console client application manages the device directly and uses Digest authentication, the ACL contains an entry per user. Each entry contains a user ID, a password, a list of the Intel AMT realms to which the user has access, and whether the user has local access, remote access, or both.

Username	Password	Realms	Access
User01	*****	Admin; Storage	Remote
User02	*****	Agent Presence	Remote; Local

When the Intel AMT device is configured to work with Active Directory, an ACL entry contains an SID, a list of realms, and local/ remote access permissions. An SID can be for an individual user or it can be an Active Directory Group and represent multiple users.

SID	Realms	Access
01050000374FF6...	Admin; Storage	Remote
0105000013AC81...	Agent Presence	Remote; Local

An Intel AMT device can operate with both forms of ACL simultaneously, so that a Management Console application that is Kerberized can access an Intel AMT device using Kerberos, while another application can contact the same device using Digest authentication. Note that the MEBx SOL/IDER settings can limit redirection applications to Kerberos-only ACL entries.

Although local applications running on the host processor on the same platform as the Intel AMT device can be in a group with an SID registered in the local Intel AMT device, it is recommended that local applications not look at the device as a remote server and instead should use Digest authentication.

The Setup and Configuration Server adds Intel AMT objects to the Active Directory database in a simplified way, and configures the Intel AMT device for Kerberos as detailed below.

6 Configuring an Intel AMT Device for Kerberos Authentication

The following tasks must be performed to prepare an Intel AMT device for Active Directory:

- Create an Intel AMT object in the Active Directory database.
- Create the SPNs for the Intel AMT object in Active Directory.
- Compute the Kerberos master key from the Intel AMT object password.
- Configure the Intel AMT Kerberos parameters.
- Configure the Intel AMT ACL.

The following maintenance functions must also be performed:

- Change the Intel AMT object password.
- Update the Intel AMT master key.

The task that adds or modifies Active Directory database entries requires privileges to:

- Create Intel AMT objects.
- Change an Intel AMT object password.
- Update the link attribute in the Active Directory computer object.

These functions are performed either via Active Directory API functions or via Intel AMT API functions.

The Sample Configuration Server in the Intel AMT SDK performs a simplified version of several of the above functions. It creates a user entry (not an Intel AMT object entry) in the Active Directory database and creates the SPNs for the entry. When the Sample Configuration Server detects the Kerberos option in the .CONF.xml file, it creates the directory entry and the associated SPNs using functions in the file **KerberosUtil.cpp**, which can be found in the SDK directory **Windows\Intel AMT SDK\Samples\Configuration\ConfigurationServer\Src**. These functions call Microsoft-supplied sample code that calls Active Directory API functions. The application must run (that is, be executed by a user) with sufficient Active Directory privileges to create a user and add to or modify its properties.

The Sample Configuration Server sets Kerberos options in the Intel AMT device by calling the **SetKerberosOptions** function. The *Network Interface Guide* describes this function and its parameters.

The Sample Configuration Server also updates the time on the Intel AMT device using the Network Time Interface. See the *Network Interface Guide* description of the Network Time Interface for the algorithm used to synchronize Intel AMT time. This approach is used to minimize differences between client and server clocks necessary for timestamp validation.

Use the command **AddUserAclEntryEx** to create an ACL for the Intel AMT device that supports the Active Directory environment. The data structure allows creation of either Digest or Kerberos entries. This function is also described in the *Network Interface Guide*.

7 Redirection Library Kerberos Support

The Intel AMT Redirection Library supports Kerberos authentication. For a user of the library, the only difference between using Kerberos for authentication and using the library without Kerberos is that use without Kerberos requires a valid username and password. Calling the library with Kerberos does not require the username and password.

An Intel AMT device redirection setting is established using the Intel AMT BIOS extension. The SOL/IDE-R BIOS option has a choice of authentication or no authentication. When the authentication option is selected, Intel AMT will authenticate with Kerberos or Digest.

A client application initiates an SOL or IDE-R session with an Intel AMT device by calling either **IMR_IDEROpenTCPSession** or **IMR_SOLOpenTCPSession** – both are functions in the redirection library. These functions have as input parameters the username and password of the client. The library opens a socket with the Intel AMT device and negotiates the protocol to be used between them. If the SOL/IDE-R authentication was enabled, then the library will attempt to contact a ticket granting server for establishing a Kerberos connection. If this fails, the library will attempt to connect using the username/password combination. Note that if the normal mode of operations is to connect using Kerberos, the username/password parameters to the library functions will be ignored.

8 Kerberos Authentication Using .NET

The Intel AMT SDK examples that have Kerberos authentication capability are all implemented using WinHTTP. The following example shows how to perform Kerberos authentication using .NET 2. The example connects with the Intel AMT Remote Control service.

```
//-----  
//  
// Copyright (C) Intel Corporation, 2003 - 2006.  
//  
// File: KerberosClient.cs  
//  
// Contents: Sample code for a Intel(R) AMT Release 2.0 network  
// client with Kerberos authentication.  
//
```

```
//-----
using System;
using System.Net;

/*
 * This is a short example of invoking a WSDL on Intel AMT using
 * Kerberos authentication on the clear
 *
 * Compile Requirements:
 * - Visual Studio 2005
 * - The RemoteControlService must be generated from
 *   RemoteControlInterface.wsdl with wsdl.exe
 *
 * Runtime Requirements:
 * - Client host must have .Net 2.0 framework.
 * - Client host must be a member of the domain forest.
 * - Intel AMT must be configured for Kerberos authentication (this can
 *   be done with the sample configuration server.
 * - The user whose credentials are provided below must be a member
 *   of a group which has a Kerberos ACL installed on Intel AMT that
 *   includes the RemoteControl security realm.
 * The sample Configuration Server has a set of Kerberos settings
 * that includes the creation of a Kerberos ACL for the group
 * "Domain Users" (any authenticated user)
 * and includes the RemoteControlRealm (this realm has the number 5).
 */
namespace SpnProblem
{
    class KerberosClient
    {
        // The assignments below must be replaced with meaningful values.
        // *In a production environment the password must be stored securely*
        const string hostname = "hostname";
        const string domain = "domain";
        const string user = "user";
        const string password = "password";

        [STAThread]
        static void Main(string[] args)
        {
            try
            {
                string fqdn = hostname + "." + domain;

                // set target URI
                Uri serviceUri = new Uri("http://" + fqdn +
":16992/RemoteControlService");
                RemoteControlService r = new RemoteControlService();
                r.Url = serviceUri.ToString();

                // set SPN for target uri
                AuthenticationManager.CustomTargetNameDictionary.Add(
                    serviceUri.ToString(),
                    "HTTP/" + fqdn + ":16992"
                );
            }
        }
    }
}

```

```

        // set Kerberos credentials
        CredentialCache myCache = new CredentialCache();
        myCache.Add(
            serviceUri,
            "Negotiate",
            new NetworkCredential(user, password, domain)
        );
        r.Credentials = myCache;

        // invoke RemoteControl
        uint state;
        r.GetSystemPowerState(out state);
        System.Console.WriteLine("state = {0}", state);
    }
    catch (WebException we)
    {
        Console.WriteLine("Error: could not connect to remote
host");
        Console.WriteLine(we.Message);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
    }
    finally
    {
    }
}
}

```

9 Security Considerations

- The SDK samples based on WinHTTP do not implement a check for Kerberos Mutual Authentication. Although the WinHTTP library always queries Intel AMT for Kerberos Mutual Authentication support, it does not check the result. Intel recommends using TLS Mutual Authentication in combination with Kerberos to compensate for this limitation.
- The replay cache mechanism implemented in the firmware stores the content of the decrypted client authenticator of each new SOAP request to prevent a possible replay attack of a Kerberos token. In some cases, (for example, during stress testing), a SOAP request containing a valid Kerberos Token will be rejected if the replay cache is full or if the cache was lost. The replay cache is purged of authenticators that are beyond the window of the maximum clock tolerance (typically 5 min) each time a new authenticator is received. The clock tolerance can be configured in Intel AMT.

When a token is rejected, the SOAP request will return an error code within an “HTTP 401 Unauthorized” error:

KRB_ERR_GENERIC – “Replay cache is full; Try later”.

- Some Active Directory configurations enable an optional client IP address field in the ticket that allows a server to guarantee that the ticket is used from a specific address only. The Intel AMT firmware does not check this optional field when validating the correctness of the ticket.

This was a design decision, since checking the IP would likely cause issues in an environment that uses Network Address Translation (NAT), and provides only minimal additional security, since IP addresses are easily spoofed.

- Intel AMT does not check the Kerberos key version number against the version number in a Kerberos ticket. The version number is incremented by Active Directory when the AMT object password is updated. The management application that requested the password update then updates the Intel AMT platform with a call to **SetKerberosOptions**. Two of the parameters to this function are the master key, which is based on the AMT object password, and the key version. Clients attempting to connect with an Intel AMT platform using a Kerberos ticket based on the previous password will now fail as the Intel AMT firmware does not check for previous versions of the password and will only authenticate connections based on the new password. Client applications should request a new ticket when a connection fails as it may be due to an unexpired ticket not matching an updated password.

10 Notes and Limitations

- All systems performing SOAP HTTP communications to Intel AMT devices using Kerberos authentication require a Microsoft Hotfix to WinHTTP so that they will execute correctly. This applies both to remote applications and local agents, although using Kerberos authentication from the local platform is not recommended. It also applies to the platform performing modifications to the Active Directory schema. Hotfixes for Windows 2003 and for Windows XP are available from Microsoft. The Hotfix number is KB899900.
(See <http://support.microsoft.com/?id=899900> for detailed information.)
- Internet Explorer 6 requires a Hotfix to use the Kerberos authentication protocol to connect to the Intel AMT WEBUI. The Hotfix number is KB908209.
(See <http://support.microsoft.com/?id=908209> for detailed information.)
- A client application that attempts to use Kerberos authentication must be on an intranet in common with the Intel AMT device. This is a requirement of the WinHTTP library. It is necessary to identify the domain of the intranet to WinHTTP by setting up a proxy bypass. If this is not done, the SDK sample will not work with Kerberos authentication. The command used to set the proxy bypass domain is a function of the operating system in use:

In the following example, applicable to **Windows 2003** and **Windows XP**, the command **proxycfg** configures the intel.com domain to be on the proxy bypass list. “_” is set up as a dummy “proxy”.

```
Z:\>proxycfg
Microsoft (R) WinHTTP Default Proxy Configuration Tool
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
Current WinHTTP proxy settings under:
  HKEY_LOCAL_MACHINE\
    SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
  Settings\Connections\
    WinHttpSettings :
```

```
    Direct access (no proxy server).
```

```
Z:\>proxycfg -p "_" *.intel.com
Microsoft (R) WinHTTP Default Proxy Configuration Tool
Copyright (c) Microsoft Corporation. All rights reserved.
```

```
Updated proxy settings
Current WinHTTP proxy settings under:
  HKEY_LOCAL_MACHINE\
    SOFTWARE\Microsoft\Windows\CurrentVersion\Internet
Settings\Connections\
  WinHttpSettings :

  Proxy Server(s) : _
  Bypass List     : *.intel.com
```

See

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winhttp/http/proxycfg_exe_a_proxy_configuration_tool.asp

for a description of the **proxycfg** command and its parameters.

In **Vista***, the command **proxycfg** has been dropped and WinHTTP proxy setting has been incorporated into the **netsh** command.

From the command line, perform the following sequence to place intel.com on the bypass list:

```
Z:\>netsh
netsh>winhttp
netsh winhttp>set proxy
netsh winhttp proxy>set proxy proxy-server="xyz" bypass-list="<local>, *.intel.com"
```

Current WinHTTP proxy settings:

```
Proxy Server(s) : xyz
Bypass List: <local>, *.intel.com
```

Note that the proxy server is a dummy name.

11 Configuration Example

The following example shows how to setup and configure Intel AMT to work with Active Directory and Kerberos using the Sample Setup and Configuration Application. See the *Intel AMT Setup and Configuration Server Installation and User Manual* to configure for Kerberos using the Intel AMT SCS. The document describes:

- Setting up a Domain Server
- Setting up the Intel AMT host processor
- Setting up the sample Setup and Configuration Application (SCA)
- Preparing Intel AMT to be configured
- Configuring a remote platform or local host to work with Intel AMT
- Demonstrating that the configuration was successful

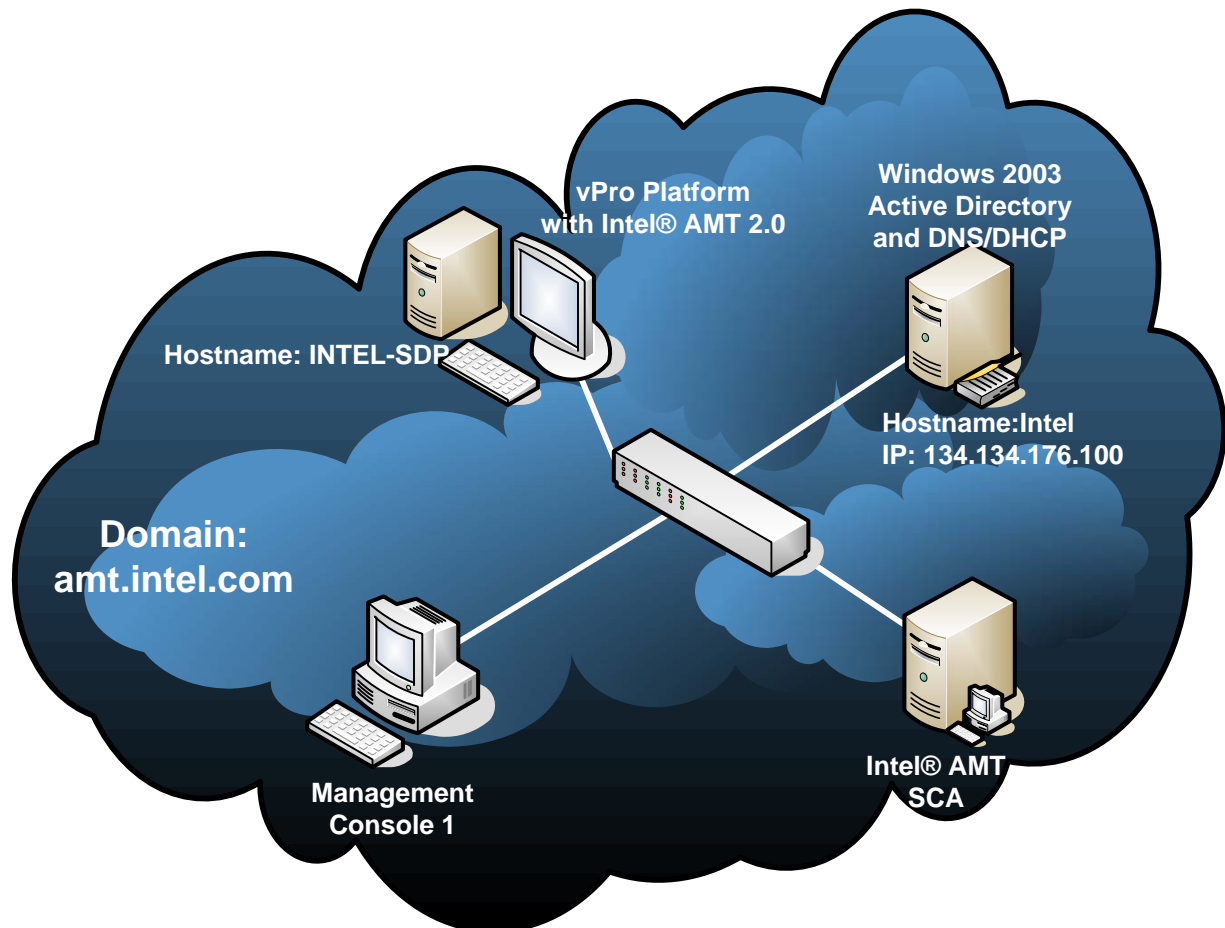
11.1 Assumptions

The example assumes that the reader has a basic knowledge of networking, Active Directory, and Kerberos.

Be sure to install all necessary Hotfixes, as described [above](#).

The Active Directory installation is performed on a platform running Windows 2003 Server with Service Pack 1. A DHCP server should be installed as well. DNS server configuration occurs during the Active directory installation.

A typical configuration would have multiple platforms performing different roles, as in the diagram below, where the domain server, setup and configuration server, and management console are all on separate machines. The example described below uses only one platform for all of these functions.



11.2 Configure the Domain Server

Install all the necessary applications:

- Active Directory
The following Microsoft site has instructions for installing Active Directory.
<http://support.microsoft.com/kb/324753>

See the following site also for step-by-step help in installing Active Directory.
http://www.petri.co.il/how_to_install_active_directory_on_windows_2003.htm

The example discussed below configures a domain named "amt.intel.com" on a Domain Server named Intel.

The tool **adsiedit.msc**, available on the Windows Server 2003 CD, is useful for managing Active Directory and user and group entries.

- DHCP and DNS servers
DHCP and DNS servers are not required for Intel AMT, but a DNS server must be present for Kerberos operation and DHCP servers are typically used in enterprise installations. By default, DHCP will enable the 006 DNS and 015 DNS Domain Name options.
In the example, the domain is set to **amt.intel.com** and the domain server IP (and therefore the DNS IP) is set to **134.134.176.100**.
The name of the domain server in the example is **intel**.
- In Active Directory, Create a Group for AMT users in the domain Users list named AMT. This group will be used later for defining the Intel AMT ACL entry. A user must be in this group to communicate with an Intel AMT device.

11.3 Configure the Intel AMT host processor

Set the hostname and domain of the host processor. In the example, the hostname of the vPro platform is set to **Intel-SDP**, and the domain is set to **amt.intel.com**, so the **FQDN** of the host platform is **Intel-SDP.amt.intel.com**.

The platform should now be able to identify itself to the domain server.

Create a user for the host platform, first adding it to the list of domain users on the server.

11.4 Prepare the SCA for Kerberos

Configure the default.config.xml Kerberos parameters. The example parameters are:

```
<host_name>intel-sdp</host_name>
<domain_name>amt.intel.com</domain_name>

<kerberos>
<containerDN>CN=users,DC=amt,DC=intel,DC=com</containerDN>
<password>MY$root1</password>
<clock_tolerance>5</clock_tolerance>
<acls>
<acl>
<access>any</access>
<user_group_dn>CN=AMT,CN=users,DC=amt,DC=intel,DC=com</user_group_dn>
<realms>
<realm>3</realm>
</realms>
</acl>
</acls>
</kerberos>
```

The containerDN is set to the container name Users for the domain amt.intel.com.

The password entry is the “secret” held by Active Directory for the User object associated with the Intel AMT platform. It is *not* the password used to access the Intel AMT device when using non-Kerberos information. Rather, it is the password used by Intel AMT only for decrypting Kerberos tickets.

Access is set to “any”, meaning that both remote and local applications can authenticate using Kerberos.

The clock tolerance is set to five minutes. This is the length of time that entries are maintained in the replay cache.

In default.config.xml, also enable setting the system time so the Intel AMT device is synchronized to the server clock:

```
<set_network_time>true</set_network_time>
```

The example sets up one ACL entry. The Kerberos SID in the entry corresponds to the Group named AMT defined in the Users in the amt.intel.com domain. Any user who is a member of this group can access this Intel AMT platform. Realm 3 is the PTAdministrationRealm realm, so any user in the AMT group has sufficient privileges to access any Realm.

TLS was not enabled in the example.

If the SCA process was run previously, there may be an existing User entry for the Intel AMT instance. This Active Directory user entry must be deleted; otherwise the SCA will generate an error indicating an attempt to create a duplicate user.

11.5 Configure Intel AMT using the MEBx sub-menu

Because the platform running the SCA was not registered in the DNS as "Provisionserver", The IP address of the Domain server was entered as the IP of the setup and configuration (provisioning) server.

A valid PID/PPS pair must be entered. The first set in the default psk.repository.xml was used.

All other parameters were left to their default values.

The SCA was activated and configuration completed normally, with the results shown in the screenshot below:

```

C:\Documents and Settings\Administrator\Desktop\sdk 2.0_124363\Configuration\ConfigurationServ...
Waiting for incoming connection...
[2006-07-18 14:32:25] Incoming Connection from 134.134.176.1:16994
Incoming data is:
  Version: 2
  Count : 1
  UUID  : 03020100-0504-0706-0809-0A0B0C0D0E0F
  PID   : XXXX-XXXX

reading configuration from default.conf.xml

>> starting soap call sequence <<
soap call: GetCoreVersion ok
AMT version: 2.0.2
soap call: GetPkiCapabilities ok
[PKI capabilities]
CrlStoreSize = 1424
RootCertMaxSize = 1500
RootCertMaxInstances = 4
FqdnSuffixMaxEntries = 4
FqdnSuffixMaxEntryLength = 50
CertChainMaxSize = 4100
SupportedKeyLengths = 1024,1536,2048
[PKI capabilities]

soap call: SetHostName ok
soap call: SetTcpIpParameters ok
soap call: SetDomainName ok
soap call: SetPingResponse ok
soap call: SetProvisioningMode ok
soap call: SetRngKey ok
soap call: SetTlsKeyAndCertificate ok
soap call: GetLowAccuracyTimeSynch ok
soap call: SetHighAccuracyTimeSynch ok
soap call: EnumerateTrustedRootCertificates ok
- No trusted root certificates to delete
soap call: AddTrustedRootCertificate ok
soap call: SetCRL ok
soap call: SetTrustedFqdnCN ok
soap call: SetTlsOptions ok
soap call: EnumerateUserAclEntries ok
- No existing ACLs to remove
soap call: AddUserAclEntryEx ok
creating user in domain under CN=users,DC=amt,DC=intel,DC=com
user name: intel-sdp
account name: intel-sdp
registering SPNs to user "intel-sdp":
  HTTP/INTEL-SDP.amt.intel.com:16992
  HTTP/INTEL-SDP.amt.intel.com:16993
  HTTP/INTEL-SDP.amt.intel.com:16994
  HTTP/INTEL-SDP.amt.intel.com:16995
soap call: setKerberos ok
soap call: SetEnabledInterfaces ok
soap call: GetDigestRealm ok
soap call: SetAdminAclEntryEx ok
soap call: CommitChanges ok
Configuration process completed successfully

Waiting for incoming connection...

```

SDK Samples can be run from a management console platform or from the local host. There are setup steps required for this to work in a Kerberos/Active Directory environment.

11.6 Configure a Remote Platform or Local Host to work with Intel AMT using Kerberos

- The platform needs to be recognized by the domain.
The platform is recognized in the domain by setting the domain and hostname and attempting to login. Then any user on the platform that activates a management console program or an

SDK sample must be in the Active Directory group that was defined in default.config.xml (in the example, this is the AMT User group).

- Microsoft Hotfix must be installed.
The SDK samples and any other program authenticating with an Intel AMT instance with Kerberos using the WinHTTP library need Microsoft hotfix KB899900 installed.
- Run the proxyconfig command described [above](#).
- If Microsoft Internet Explorer (IE) will be used to connect to the WebUI feature of Intel AMT, then the domains containing Intel AMT platforms must be included in IE's list of trusted sites. Add the domain by selecting the IE **Tools>Options** menu, selecting the **Security** tab and selecting either **Local intranet** or **Trusted sites**. Click **Sites...**
Under **Trusted sites**, add the domain (for example, *.intel.com). Under Local internet, select **Advanced...** and add the domain explicitly.
Other applications may have similar configuration requirements.
- All calls to Intel AMT must address the device using the FQDN. The Active Directory entry for the Intel AMT platform is defined with the FQDN, so trying to connect using only the host name will fail. In the example, INTEL-SDP will not work, but intel-sdp.amt.intel.com will. This is also the case when using TLS as the certificates are based on the fqdn and not the hostname.

11.7 Demonstrate that the Setup was done Correctly

Run one of the remote samples from the SDK to demonstrate the Kerberos mechanisms are functioning. For example, run the GeneralInfo Sample. This can be from the domain server or another platform in the same domain. The user must be a member of the AMT group, as described above.

Run one of the local samples (such as the Agent Presence sample) to show host-to-Intel AMT Kerberos authentication.