Chapter 5

Intel® Active Management Technology Overview

Management is doing things right; leadership is doing the right things.
—Peter Drucker (1909–2005)

As we discussed in the last chapter, Intel® Active Management Technology (Intel AMT) is a major component of the Intel vPro™ platform that differentiates this platform from other platforms. Let's look more closely at the details of the capabilities that Intel AMT provides. In this chapter we look at different end-user solutions that are built using the Intel AMT capabilities.

Key Characteristics

Before we examine the functional capabilities of Intel AMT, let's briefly look at the key characteristics that make it different from traditional management based on OS agents.

Out of Band Access

Intel AMT is powered by a separate hardware engine: the Manageability Engine in Intel chipsets. This hardware engine has its own processor, its own

portion of the memory, its own portion of Flash, its own sideband access to the networking controllers, and so on. It therefore provides the out of band access benefits that we looked at in Chapter 1. We will go into details of these Intel AMT components in Chapter 7. At this point, it is enough to say that Intel AMT can operate independent of the main CPU and independent of the operating system state. Intel AMT is always available and always accessible.

Low Power Operation

The Manageability Engine from which Intel AMT operates can function in lower power states. The system can be in the low-power standby or hibernate mode or even turned off, and Intel AMT will continue to function.

Operation in Various System States

Advanced Configuration and Power Interface (ACPI) is an industry specification that defines the system sleep states (S-states). S0 is the state when system is fully powered on. S3 state is called system standby state, where the OS image is intact in the system memory. System memory contents are maintained in self-refresh mode. The processor, disk and all the peripherals are turned off. S4 state is when the memory contents are transferred to disk, and memory is powered off. This is also called hibernate. S5 is system off state. Intel AMT is available in all these system S-States.

When the system is in S0 (fully powered) state, the system may still be in various states of OS or BIOS operations. These states are not defined by ACPI, but we make this further distinction based on accessibility to the platform from a remote management console.

When the OS is fully operational and networking access is functional, agent-based methods can usually be used to manage the system. Intel AMT works in this state and coexists with these mechanisms.

When the OS is not functioning normally, an agent or a device driver has crashed, the networking software is running slow, some rogue application is hogging the CPU, or the OS is hung or has crashed, OS-based agents are not accessible by management consoles. Intel AMT is functional in these states and can be used to bring the system back to its normal functioning state.

In the pre-OS environment, when the OS has not yet started or the system does not yet even have an OS installed, or in certain cases when the BIOS encounters failures, Intel AMT continues to work and provides critical remediation services to allow the system to be fixed.

OS-Independent Agent-less Solution

Intel AMT does not require any agent in the OS for its core operations; this makes it OS independent. Intel AMT capabilities are available whether the system has Windows, Linux, or any other operating system installed, or for that matter even if no operating system is installed.

Tamper-Resistant Solution

Intel AMT components execute in an isolated environment and are not accessible by any software from either the main system operating system or applications. Malware cannot change or modify any code module in Intel AMT. Intel AMT is protected at the hardware level. The only access to Intel AMT is through a well defined API, which requires authenticated credentials to access. No mechanisms are provided to download a software component into Intel AMT, which prevents any malware from getting into the Intel AMT execution environment.

Discover, Heal, and Protect

Having looked at the key characteristics of Intel AMT, let's look further into its various capabilities.

Key Intel AMT usage models are centered around the three pillars of Discover, Heal and Protect.

A management console can *discover* all Intel AMT systems on the network and collect hardware and software inventory information from those systems. Intel AMT is able to do all this regardless of system power or health state, and regardless of OS state.

A management console can get alerts from the Intel AMT system when things are failing or about to fail. Once these alerts are received the system administrator can remotely diagnose and *heal* the system with tools such as remote login, remote boot, text console redirection and remote media redirection.

Intel AMT provides hardware-based capabilities to watch the operation of the critical security agents and can alert the system administrators when these events are accidentally or purposely stopped by malware. Some versions of Intel AMT also provide a mechanism to watch outgoing network traffic and detect anomalous patterns that are indicative of a spreading worm. Thus, Intel AMT provides capabilities to *protect* the systems and the network from malware.

Key Capabilities

In this section, we walk through the key capabilities of Intel AMT. In the next chapter, we look at how we can build various solutions using these capabilities.

Hardware Inventory

Intel AMT keeps all the hardware inventory data in a nonvolatile RAM (NVRAM). In some cases, the manageability engine is directly connected to various components on the platforms, and Intel AMT collects that information directly. In other cases the BIOS scans the hardware, collects the asset information and relays it to Intel AMT to store in the NVRAM. The information includes the make, model number, asset tag, and so on. This information is made available through a management console.

Software Inventory

Intel AMT has the capability to keep software asset information in NVRAM as well. OS services, drivers, and applications can relay this information to Intel AMT, which stores it in Third Party Data Store (described later in this chapter). This information is then made available to management consoles even when the OS is unavailable. This is particularly useful when the OS and applications fail and the system administrator needs to know the system configuration to be able to fix it.

Hardware Health and Platform Sensors

Intel AMT monitors various hardware components such as processor, memory, chipset, networking controller, and disk controllers on the platform and proactively reports any failures or impending failures to the management console. Several sensors on the platforms collect readings from various physical places on the platform and contribute to determining the overall health status. These sensors include CPU presence and error sensors, temperature sensors, voltage sensors, chassis integrity sensors, and fan sensors.

Remote Power Control

Intel AMT allows a remote system administrator to reset or power off a system even when the normal access methods to reach the operating system may not work. For example, the OS is hung or crashed, or an OS is not installed. Furthermore, a system administrator can even reach a system that is powered off and send it a command to turn it on.

Boot Control

This capability allows a system administrator to control how the system is booted. A specific boot device can be selected. For example, the system can be instructed to boot from a hard disk, a CD-ROM, a network controller (PXE boot) or a remote IDE device (explained below). Additionally, options can be provided to change the display of BIOS messages on the local console, or redirect the messages to the remote console.

Text Console Redirection

This feature is also often known as serial redirection or serial over LAN (SOL). Intel AMT hardware emulates a serial COM port for the system. This allows all text output to this COM port, to be packaged by Intel AMT into network messages and sent to remote console over the network. Conversely, the keyboard input is captured at the remote console and sent through the network packets to the Intel AMT system. Intel AMT takes that keyboard sequence and sends it to the COM port as if it were the local keyboard input.

This capability allows all the pre-boot BIOS text output and input to be controlled by a remote management console. Pressing a function key to change BIOS settings can be achieved using this functionality.

Disk Redirection

This feature is also known as IDE Redirection (IDE-R). This allows a remote system administrator to mount a storage media, which can be a local disk on system administrator's console, a disk volume, a CD-ROM, and ISO image, a USB flash drive, or any other storage device to be accessed by the Intel AMT system as if it is a local IDE drive.

Intel AMT hardware emulates a local IDE disk drive interface for the system. All the writes are captured by Intel AMT and packaged into network packets and send to the console. Console software takes those packets and writes it to the mounted media. Similarly any read requests from an Intel AMT system go through the remote console and content is returned via network packets. To the local system, this all appears as if it is communicating with a local IDE drive, except for the somewhat slower performance compared to local access.

Persistent NVRAM Log

Intel AMT maintains a log of all critical platform events in NVRAM. This log is a persistent historical record of the events that happened on the platform. It is embedded in the platform and is more reliable than the traditional disk-based logs that are prone to be lost if the disk fails or is replaced. Like the rest of the Intel AMT capabilities, this log is accessible in all platform states, thereby providing an important diagnostic tool.

Alerts

Intel AMT can be configured to send directed alerts to one or more configured IP addresses. These alerts can be for configured critical or informational events. The alerts can go out as SNMP traps, or can be delivered using a more sophisticated WS-Management event registration and delivery mechanisms.

Third Party Data Store (3PDS)

Intel AMT provides the capability for a third party to have a programmatic and secure access to a fixed size reserved block in the NVRAM. This allows a software application to keep a critical portion of the data in this persistent storage and not get affected by disk wipes or by malware access. Another reason to keep the data in NVRAM is that it is accessible all the time

independent of the system state. One example of this is a virus definition file for the virus scanner software. A system administrator can update all the systems on the network with the version number of the latest virus definition files without worrying about the fact that some systems may not be up and running. When the systems are booted or brought up from sleep, the virus scanner software agent can check if it needs to pull the latest virus definition files from the server.

Agent Presence

Intel AMT provides a watchdog capability that allows a local application on Intel AMT system to be configured to periodically send a heartbeat message to the Intel AMT hardware. If the application fails to send the heartbeat within a pre-defined time interval, the Intel AMT hardware can generate an alert and notify the remote console, as well as send a local notification to display on the console. This allows critical software agents to be watched for presence. This is very useful to watch critical security agents on the platform.

System Defense

The system defense feature allows a remote system management console to define and enforce network security policies. A system defense policy contains a set of filters that are applied to incoming or outgoing packets, along with a set of actions to perform when the filter is matched, or not-matched. Once these policies are loaded into Intel AMT and activated, Intel AMT constantly monitors the incoming and outgoing network packets and takes specified actions. The actions can include sending an alert to the system management console, logging an event in event log, taking autonomous action to throttle incoming or outgoing traffic, or even completely stopping the network traffic. While the network traffic to host is stopped, Intel AMT is still accessible over the network for remediation actions.

Endpoint Access Control

Intel AMT helps secure network endpoints by validating their compliance with network policies. The Endpoint Access Control (EAC) feature allows the system administrators to implement differentiated policy enforcement and configuration based on the security state of the endpoint. At every

connection, or on demand, a client system's profile is securely surveyed in a trusted manner. The system posture (including credentials, configuration, and system data) along with Intel AMT configuration parameters (Firmware Version, TLS enabled, SOL enabled, and so on) is compared to current requirements. For systems not meeting the minimum standards, network access is restricted, and a user notification is displayed to convey to the end user that normal network operation will be delayed until remediation is complete.

Full authentication and posture checking before allowing network access can greatly reduce the potential for malware to propagate onto the network, and this allows the IT administrator to maintain all systems in compliance with current policies and limits rogue or visitor systems from gaining network access. Intel AMT enables the acquisition of accurate endpoint state and attributes information for network admission control, via "always-available" communication, regardless of the PC's power state, the state of the OS, or the absence of management agents. Accurate identification of machines in a pre-boot environment results in improved automation and enforcement of secure network policies.

Interfaces and Protocols

Intel AMT capabilities are accessed in a secure manner from a management console or from an application running locally on the Intel AMT platform. Several mechanisms for Intel AMT access exist and are listed below.

Network Access

All communications with Intel AMT system from a remote management console happen over the standard networking protocols. This communication is supported over wired (Ethernet) or wireless (Wi-Fi†) connections.

Intel AMT can have its own IP address, or can share the IP address with the host. Shared IP address is the most commonly used configuration. When the IP address is shared, only certain packets destined to Intel AMT ports are filtered and sent by the networking controller to Intel AMT. This communication happens without any OS assistance via a sideband connection between the networking controller and the Intel AMT manageability engine hardware.

Intel AMT supports communication over fully encrypted secure channel provided by TLS (HTTPS). With the secure communication channels, all network traffic is protected against.

All Intel AMT communications require authentication of the user credentials. Intel AMT manages the access controls based on the permissions set for the user. Intel AMT uses passwords and can also integrate with Kerberos and Active Directory security mechanisms.

External Operations Interface

Intel AMT has traditionally supported a SOAP-based network interface called External Operations Interface (EOI). EOI has been the interface since the first release of Intel AMT. EOI works over HTTP and HTTPS transport protocols.

WS-Management Interface

Intel AMT today supports the standard WS-Management interface. This is the preferred interface for the future generations of Intel AMT. Intel AMT supports WS-Management over HTTP and HTTPS as well.

Platform Event Traps

Intel AMT alerts can be sent via Platform Event Traps (PET), which are defined using SNMP packet format, and are delivered using SNMP protocol. The PET format is a standard defined by DMTF.

In addition to PET alerts, Intel AMT can also send events using WS-Management event formats.

Local Access

Intel AMT allows local applications to access some of the Intel AMT functionality through a device driver and higher-level software layers.

Intel® Management Engine Interface

Intel Management Engine Interface (Intel MEI) is a hardware interface that is used to communicate to the Intel AMT subsystem. Intel MEI is essentially a PCI device interface. Intel MEI is bidirectional, as either the host OS or Intel AMT firmware can initiate transactions.

Communication between the local host OS and Intel AMT is accomplished by means of the Intel MEI driver.

The BIOS typically sends messages directly through the Intel MEI interface. Most OS-based applications go through another layer of software that sits on top of Intel MEI and provides the higher level of abstraction.

Network-Compatible Interface

Local Manageability Service (LMS) is a service that runs locally in the user space of the host OS. LMS exposes Intel AMT functionality through standard network-compatible interfaces (EOI and WS-Management as described above). LMS listens for the request directed to the Intel AMT local host and when an application sends a SOAP/HTTP message addressed to the local host, LMS intercepts the request and routes the request to Intel AMT via the Intel MEI driver.

Intel® AMT SDK

Intel AMT SDK provides the documentation for interfacing with Intel AMT along with necessary libraries and also provides sample source code.

For example, if you want to read/write data to the third party data store (3PDS) then your application uses the storage library API to accomplish the task. The storage library talks to the LMS interface and that in turn talks to the Intel MEI driver to access the 3PDS. As far as the client application is concerned, it simply talks to the LMS or Storage Library (Intel AMT SDK) and the message gets routed to the Intel Management Engine via the Intel MEI driver.

Intel® AMT and Enterprise Infrastructure

Intel AMT needs to work with existing enterprise infrastructure, as most of the Intel vPro platforms are deployed within a standard IT environment. In order to facilitate this integration, Intel AMT provides a number of capabilities to make the deployment in enterprise environments straightforward. Most of these components are not required in smaller environments.

Active Directory Integration

For the IT environments that deploy Active Directory for authentication, authorization, and access control, Intel AMT can be configured to use Active Directory. Intel AMT is also capable of using Kerberos Authentication and Active Directory–based authorization and access controls.

Setup and Configuration Server

In order to make large-scale Intel AMT deployments easier, a setup and configuration service can be deployed. This service can be integrated into management consoles. The Setup and Configuration service discovers the new Intel AMT systems on the network and can send them the configurations parameters in a secure manner.

Management Consoles

Management consoles are a key part of providing overall manageability solutions for Intel vPro platforms. Management consoles can use the standard WS-Management interfaces or legacy EOI interfaces to communicate with Intel AMT platforms. The Intel AMT SDK provides another higher level of API abstraction to make this integration even simpler.

If the enterprise is not setup with Active Directory integration, then the management console has to setup the passwords on the system with for password/digest authentication, deploy Intel AMT access control realms on the system with Intel AMT, and manage that over time.

Management consoles also have the responsibility to maintain the up-to-date time in the Intel AMT subsystem. The periodic time synchronization is part of the management console responsibility.

Certificate Server

If the enterprise needs to support TLS and HTTPS, then it needs to provide services from a Certificate Server to generate certificates for Intel AMT. These certificates are deployed in systems with Intel AMT.

BIOS

Although the BIOS is technically a component on the platform itself, we include it in this list because changes are required for Intel AMT to work properly. On Intel AMT systems, the BIOS must contain an Intel Manageability Engine BIOS Extensions (Intel MEBX) module to be able to enable and configure Intel AMT. This has to be considered when performing BIOS upgrades. Typically, the vendor that supplies Intel MEBX–enabled BIOS with the platform will include it in the updates, so this generally should not be a problem.

Intel AMT also provides the capability to check the firmware and measure its different components as they are initialized. This measurement can be matched against a pre-stored value. The BIOS typically does this and makes a policy-based decision if the measurements do not match.

Routers, Access Points, and Servers

As described earlier regarding Endpoint Access Control capability, Intel AMT needs to operate in an 802.1x network to provide the posture information to the access points or backend network. The policies have to be configured for Intel AMT to get access to the network if the posture information is as expected.

DHCP and DNS

Intel AMT integrates with DHCP to obtain an IP address. Then it uses DHCP options to provide its discovery information and update DNS servers, which is later used to discover the Intel AMT systems on the network.

Wi-Fi Access Points

Intel AMT can be accessed over a Wi-Fi network. Intel AMT can work with access points that do not support encryption. It also supports WAP and WPA keys. Profiles with these keys need to be provisioned in systems with Intel AMT. Intel AMT can also synchronize the profiles with the OS.

Security Compliance Suites

Several security compliance suites need to audit the activities on a platform. Intel AMT provides a well secured audit log that records all actions that are performed with Intel AMT. Compliance suites can retrieved this audit log and analyze it as part of the overall platform audit log.

Summary

Intel AMT provides a number of capabilities that allow discovery, healing, and protection of the platform and resources. These capabilities can be accessed using local or network interfaces in a secure manner. In the next chapter, we see how these capabilities are used to solve real world problems.