Security Evaluation of Intel's Active Management Technology

VASSILIOS VERVERIS

KTH Information and
Communication Technology

Master of Science Thesis
Stockholm, Sweden 2010

TRITA-ICT-EX-2010:37

Security Evaluation of Intel's Active Management Technology

Master thesis

Vassilios Ververis
ververis{at}kth.se

Supervisor and Examiner: Prof. Gerald Q. Maguire Jr. KTH
Industrial Supervisor: Prof. Jean-Pierre Seifert, Technische Universtität Berlin

Abstract

Intel's Active Management Technology (AMT) is, a hardware-based platform for remotely managing and securing personal computers out of band. AMT is available in most desktop and notebooks PCs equipped with an Intel Core 2, Centrino, or Centrino 2 processors with support for vPro technology. AMT operates independently of the platform processor and operating system. Remote platform management applications can access AMT securely, even when the platform is turned off, as long as the platform is connected to power supply and to a network. Developers can build applications that utilize AMT using the application programming interface (API) provided by Intel. While this might seem to enable creation of a powerful management tool, a secure infrastructure that is secure against insider and outsider attacks on an enterprise network is difficult. Unfortunately this technology can also potentially be used to create a powerful backdoor that is easily deployed and offers numerous features due to its almost unlimited permissions since the platform can be managed even though it is powered off.

Sammanfattning

Intel Active Management Technology (AMT) är en hårdvarubaserad plattform för avlägset att hantera och säkra datorer utanför bandet. AMT är tillgänglig de flesta stationära och bärbara dator utrustad med en Intel Core 2, Centrino, eller Centrino 2 processorer med stöd för vPro-teknik. AMT driver oberoende av plattform processor och operativsystem. Remote optimera hanteringen ansökningar kan komma åt AMT säkert, även om Plattformen är avstängd, så länge som plattform är ansluten till linjen makt och till ett nätverk. Utvecklare kan bygga applikationer som utnyttjar AMT använder Application Programming Interface från Intel. Även detta kan verkar för att möjliggöra skapandet av ett kraftfullt verktyg i förvaltningen, faktiskt skapar en säker infrastruktur som är säkert mot insider och outsider angrepp på företagets nätverk är svårt. Tyvärr har denna teknik kan komma i används för att skapa en kraftfull rootkit som är lätt att iordningställas och erbjuder flera egenskaper på grund av dess nästan obegränsade tillstånd eftersom plattformen kan lyckades även om den är avstängd.

Contents

List of Figures

List of Tables

List of Listings

List of Abbreviations

ACL	Access Control List
AD	Active Directory
AMT	Active Management Technology
API	Application Programming Interface
ARP	Address Resolution Protocol
ASF	Alert Standard Format
ATM	Automatic Teller Machines
BIOS	Basic Input/Output System
BMC	Baseboard Management Controller
CA	Certificate Authority
CIM	Common Information Model
CN	Common Name
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSR	Certificate Signing Request
DASH	Desktop and mobile Architecture for System Hardware
DHCP	Dynamic Host Configuration Protocol
DMTF	Distributed Management Task Force
DoS	Denial of Service
DSSS	Direct Sequence Spread Spectrum
EAP	Extensible Authentication Protocol
FQDN	Fully Qualified Domain Name
GNU	Gnu's Not Unix
GPL	General Public License
GPU	Graphics Processing Unit
HDAA-MD5	HTTP Digest Access Authentication MD5
HECI	Host Embedded Controller Interface
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICH	I/O Controller Hub
ICMP	Internet Control Message Protocol
IDE	Integrated Device Electronics
IDS	Intrusion Detection System

IETF ..... Internet Engineering Task Force
INC ..... Incorporated
IP ..... Internet Protocol
IPMI ..... Intelligent Platform Management Interface
ISV ..... Independent Software Vendor
IT ..... Information Technology
ITU-T ..... Telecommunication Standardization Sector
JtR ..... John the Ripper
LAN ..... Local Area Network
LMS ..... Local Manageability Service
MAC ..... Media Access Control
MD5 ..... Message Digest 5
ME ..... Management Engine
MEBx ..... Management Engine BIOS extension
MITM ..... Man In The Middle
MPI ..... message passing interface
NIC ..... Network Interface Card
O ..... Organization
OEM ..... Original Equipment Manufacturer
OS ..... Operating System
OU ..... Organization Unit
PC ..... Personal Computer
PEM ..... Privacy Enhanced Mail
POS ..... Point of Sale
QOP ..... Quality Of Protection
RF ..... Radio Frequency
SCS ..... Setup and Configuration Service
SIMD ..... Single Instruction Multiple Data
SIP ..... Session Initiation Protocol
SMB ..... Small Medium Business
SOAP ..... Simple Object Access Protocol
SSE2 ..... Streaming SIMD Extensions 2
SSID ..... Service Set Identifier
SSL ..... Secure Sockets Layer
TCP ..... Transmission Control Protocol
TLS ..... Transport Layer Security
TXT ..... Trusted Execution Technology
UDP ..... User Datagram Protocol
UI ..... User Interface
URI ..... Universal Resource Identifier
URL ..... Uniform Resource Locator
USB ..... Universal Serial Bus
VT ..... Virtualization Technology
WLAN ..... Wireless Local Area Network

WS ..... Web Services
WWW ..... World Wide Web
XML ..... Extensible Markup Language
ZTC ..... Zero Touch Configuration

Preface

Acknowledgments

This thesis work took place at the SECT research group in technical university of Berlin. There are a number of people that they have helped me during my research on this thesis and I would like to acknowledge. First and foremost I would like to thank my supervisor at SECT research group Prof. Jean-Pierre Seifert for providing his valuable support and his excellent guidance during my whole thesis work. Also my supervisor and examiner at KTH Prof. Gerald Q. Maguire Jr, for his overall guidance, the essential comments, and his prompt feedback throughout the different sections of the thesis.

Special thanks to the SECT research group for the extensive help as well as all my friends and my family for their love and support during the work of the thesis.

Chapter 1

Introduction

Managing a computer system remotely is a non-trivial task using access and maintenance tools that are freely-available especially for the GNU/LINUX operating system (OS). A typical scenario for most enterprises is to provide remote management for multiple systems running on a variety of server platforms and heterogeneous operating systems. The requirements for information technology (IT) administration constantly shift. At the same time enterprises are trying to perform this administration within a tight budget, thus generally precluding time consuming operations while still providing reliable and secure system solutions for the enterprise's activities.

1.1 Remote management

One of most used out-of-band[71] (OOB) management infrastructure is intelligent platform management interface (IPMI) which requires a separate serial port connection from each remote machine to a centralized console server. The intelligent platform management interface (IPMI) is a popular approach to remote management. It can be performed without strict server requirements and is supported by numerous hardware vendors[41]. IPMI utilizes a baseboard management controller (BMC); a micro-controller embedded on the motherboard of workstations and servers that operates independently of the operating system OS and the management software of the system – even when the monitored system is turned off[42].

However, IPMI does not cover the complete spectrum of devices and systems that need to be managed; when compared to remote management solutions based on secure and routable protocols using a common data model. IPMI's common information model (CIM) provides a standard definition of management information for systems, networks, applications, and services, while offering support for vendor extensions. CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network[21].

In April 1998, Intel and IBM corporation collaborated on the development of alert-on-LAN technology formally known as the alert standard format (ASF). ASF[20] is a protocol for remote management and control of systems in both OS-present and OS-absent environments, primarily focused on minimizing on-site IT maintenance, maximizing system availability, and maximizing performance of the local user. Since 1998, Intel focused on developing extended features based on Web Services (WS) Management and simple object access protocol (SOAP) technologies, whilst establishing compatibility with standards based implementations and applications.

In 2005 Intel released the vPro technology enclosing Intel AMT, Intel trusted execution technology (TXT), and virtualization technology (VT). In 2007 the distributed management task force (DMTF) published the desktop and mobile architecture for system hardware (DASH) standard[22], a suite of specifications that takes full advantage of the WS-Management (WS-MAN)[23] requirements, providing standards for secure out-of-band and remote management of desktop and mobile systems. In the same year Intel released support on vPro for the DASH initiative specification, implementing hardware based support for remote repair, diagnostics, configuration, tracking assets, secure network defense filters, hardware-assisted anti-virus and rootkits[1] protection.

1.2 Problem to be addressed by this thesis project

Intel AMT is an out-of-band remote management technology making use of a dedicated communication channel which is embedded in Intel AMT enabled chipsets located on the north-bridge part of the motherboard. This technology is mainly designed to assist system administrators and IT managers to remotely maintain, repair, update, monitor, and upgrade networked computers without requiring on-site technical support or user actions. AMT helps organizations and enterprises to reduce costs and minimize time spent on administrative operations without demanding a complex management infrastructure.

Unfortunately this technology can provide a powerful backdoor[75] to the managed platform that is fully operational and accessible, even while the PC is turned off[10]. The problem to be addressed on this thesis is the fundamental security vulnerabilities found in the authentication schema, the remote provisioning mechanism, and the mobile version in terms of the new attacks vectors that they introduce to the Intel AMT platform.

[1]Rootkit is a form of system modification software, defined as an application that fraudulently gains or maintains administrator level access that may also execute in a manner that prevents detection[18]. It can be used for eavesdropping network traffic, capturing user keystrokes, alternating log files and modifying standard OS system tools to circumvent detection. Rootkit's operations are hidden on the system by manipulating OS commands that execute arbitrary code and by crafting the results returned by these commands chosen by the attacker.

1.3 Importance of this thesis

The contributions of this thesis are two fold. First it uncovers fundamental security vulnerabilities of the Intel AMT system and it sketches the implications that these vulnerabilities may have on critical operations; thence on remote network management. To demonstrate the inefficiency of current deployments of Intel AMT to home users or enterprises we orchestrate a potpourri of attacks with respect to the authentication schema, the remote provisioning process, and the mobile version. Unfortunately, these is almost no related work (concerning the problem introduced in section 1.2). At the Black Hat conference in USA, Alexander Tereshkin and Rafal Wojtczuk presented a ring -3 rootkit[10]. They describe ring -3 rootkit as a backdoor that abuses the Intel AMT technology and could potentially bypass the dedicated memory protection of AMT and compromise the AMT code executed on the chipset. They introduced an attack vector that assumes to be performed locally in order to be successful; whereas in this thesis our attack vectors can be accomplished remotely. Detailed information can be found in section 3.

Intel's AMT could be clearly seen as a major privacy concern, as it introduces new attack vectors via a covert channel, since the user is not able to monitor the network traffic, produced by the Intel AMT embedded microprocessor. An attacker or a malicious entity, could misuse this technology for monitoring, controlling, exploiting, or even gaining access to the whole system, since Intel AMT is present on computers used by both home users and corporations. Moreover, Intel AMT operates even when it is disabled in the BIOS configuration; showing that the complexity and visibility of the technology are at issue. Detailed information is present in section x. Being prone to potential attack vectors, Intel AMT provides the basis for a backdoor that silently operates in our PCs.

1.3.1 Corporations adopted Intel AMT

Intel's worldwide computing environment includes more than 100,000 PCs, by the end of 2009 they have deployed and provisioned 50,000 PCs with Intel vPro technology and the AMT platform extension[45]. Additionally there is an extensive list of large corporations that have adopted Intel AMT in their IT infrastructure.

Atos Origin an international IT services company[16]. The company's annual revenues are more than 5 € billion and it employs over 46,000 people in 40 countries.

Nottingham University Hospitals (NHS) Trust is one of the largest hospitals in the UK with an annual budget of more than 555 € million. The hospital has provisioned[36] and uses around 6,000 Intel vPro based desktop PCs embedded with the Intel AMT platform over two sites: Queen's medical center and city hospital.

University of Plymouth the fifth largest university in the United Kingdom, deployed around 4,800 PCs with Intel DQ965GF vPro and AMT enabled motherboards[29].

Bangkok's general hospital one of the largest hospitals in Thailand. They have migrated to the Intel AMT infrastructure with over 1500 PCs[77],[34]

Additionally, the following[33],[34] ISVs have added support for the Intel AMT infrastructure and applications usages, supporting management software for Windows, GNU/Linux and Mac OS: Altiris, BMC Software, Check Point Software, Cisco, Computer Associates, HP, LANDesk Software, Microsoft, Novell, StarSoftComm, Symantec, Trend Micro.

1.4 Related work

Michiel Timmers and Adriaan van der Zee have published an article[57] that describes the limitations and capabilities of Intel AMT targeting the network defense system sub-component of system defense and agent presence security tool-set[32] of the underlying technology. They have conducted a number of experiments aimed at determining the effectiveness of network blocking and rate limiting filters; in terms of throughput and the overall network performance. Their experiments are categorized based on blocking outgoing ICMP traffic, rate limiting a single protocol, the effectiveness of a rate limit on other network based traffic and also on a large number of non-matching filters. They concluded that most of these network filters can be performed via a network switch device, but without the robustness and manageability that AMT architecture offers, combined with the reduced work load on the IT staff. Moreover by using the Intel AMT platform there are no extra software requirements and computing overhead as would be the case with most remote network management suites. Their article covers only a limited fraction of the AMT platform's capabilities and does not evaluate or measure the performance of the other features that underly this technology (as described in section 2.3)

Additionally related work is the presentation at the Black Hat conference in Las Vegas, USA by Alexander Tereshkin and Rafal Wojtczuk introducing $ring - 3$ rootkits[10]. They demonstrated code injection executed into the Intel AMT management engine (ME) environment. Their attack is successfully only in the Intel Q35 chipset and requires physical presence. They implemented a proof of concept that injects arbitrary code into the chipset's AMT/ME memory on an Intel DQ35JO motherboard, using the chipset memory reclaiming mechanism[11]. Their code is based on the ARC4 architecture and repetitively write the "ITL" string to incremental locations in the host memory. This is a proof of concept code injection that imitates a rootkit's behavior. Their findings imply that a malicious person needs local access to the motherboard in order to implement such a rootkit.

1.5 Thesis structure

This thesis is organized as follows:

Chapter 2 gives a detailed overview of the Intel AMT architecture, enumerating the interface types, the use cases, the released versions, registered network ports, the offered setup, and configuration models; as well as utilization of a basic connection to the platform. Additionally we discuss functionalities that extend this platform and the supported configuration methods for the deployment of this technology.

In the third chapter there is an overview of the hypertext transfer protocol (HTTP) authentication schema that is used by AMT and we discuss our implementation (a patch) for successfully brute-forcing Intel AMT credentials giving access to the remote management functions, including some benchmarks for doing so. Moreover we present a way of exploiting the built-in support for provisioning thus exploiting remote provision to avoid the need for any physical access to the targeted computer. We clearly present how easily the AMT protection technology can be circumvented even while the AMT functionality is disabled. Additionally, we present some potential attack vectors on the mobile version of this platform classifying these as confidentiality, integrity, and availability attacks.

In chapter 4 we conclude with a summary of the potential privacy threats and lack of protection, as well as summarizing the security concerns that the AMT technology introduces. Finally, we present some suggestions for future work in order to mitigate the risks that this technology introduces as well as providing recommendations for potential research related to this technology.

Chapter 2

Intel AMT architecture

In this section we present an overview of how the AMT's architecture performs providing characteristics about the platform architecture in terms of hardware such as schema, as well as a detailed explanation of how the overall system perform in the hardware level. Additionally we introduce the available interface types that apply to AMT (local and remote), how the AMT performs in the user space level as well as enumerating the functionalities of remote network management tasks (i.e. use cases). Moreover we present the firmware releases of AMT, their offered capabilities (in respect to hardware) as well as firmware upgrade details. Additionally we cover the network ports registered and used by AMT, the setup and configuration models providing details on the configuration settings and the requirements to initiate a remote connection to manage an AMT platform. Furthermore we outline the configuration methods used by AMT and list the fraction of the systems to the original equipment manufacturer (OEM) that employ the AMT platform. Last we demonstrate a serious vulnerability of the small to medium business configuration model and we denote how the Intel's AMT local access restrictions can be circumvented.

2.1 Platform architecture

Intel AMT hardware is realized as an embedded ARC4 micro-controller[12] located in the chipset's graphics and memory controller hub. The basic system architecture is illustrated in Figure 2.1. The management engine's (ME) persistent code resides in firmware within the same flash memory device as the BIOS, in computer systems containing specific Intel motherboard chipsets as illustrated in Figure 2.2. The ME persistent code is stored outside the context of the OS in a protected, compressed, and non-volatile flash memory section in order to survive power outages and system rebuilds. The ME can access its dedicated memory space even when the system is in S3 power state (see section 2.1.1), and the ME can dynamically change the memory power state to allow the ME access to memory through the graphics and memory controller hub[37].

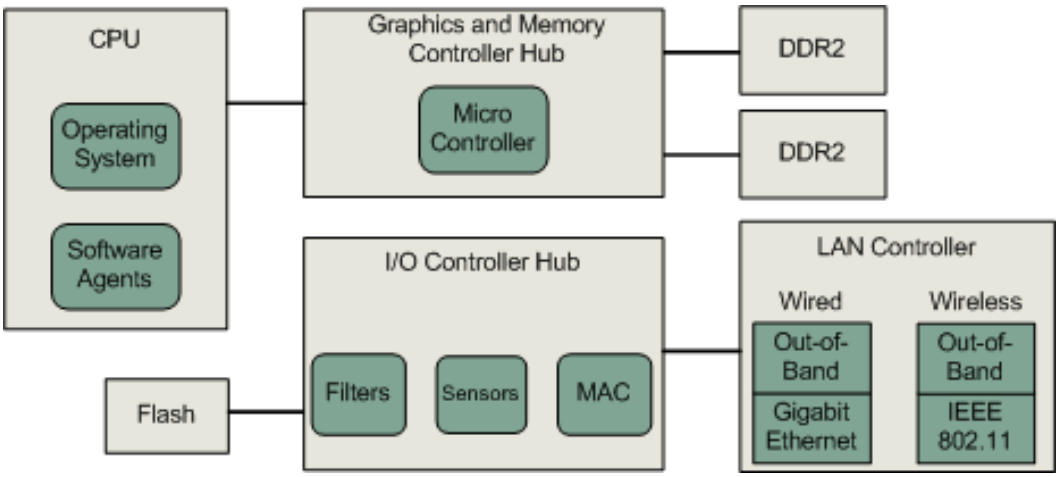

Figure 2.1. Intel AMT silicon architecture (adapted from [37])

The ME is connected via a serial peripheral interface (SPI) the FLASH memory device. The flash memory device contains also a third-party data storage designed to allow communication using a master-slave relationship model.Note that this third-party data store is designed to allow other application to securely store their data in the FLASH memory device. For dynamic memory, the micro-controller uses a small amount of the main system memory, typically less than 1% of the total system memory. A more detailed view is depicted in Figure 2.2

One of the most interesting features that this technology supports is direct access to the network interface card (NIC) via SOAP services as the ME share access to a LAN interface. In practice this means that they use the same MAC address and IP address (especially if using the dynamic host configuration protocol (DHCP)). As a result both the ME and the main processor will appear to be reachable using the same hostname. The OS can use the ME as an embedded network firewall solution (shifting the processing to the ME from the host processor) enabling the ME to handle IP port based filtering, while forwarding and routing address resolution protocol (ARP) and DHCP network packets between the host and the ME micro-controller as shown in Figure 2.3.

Intel AMT can provide manageability over a wired or wireless network environment as well as accessibility outside the context of the OS. Additionally, some operations can be initiated by the host OS of the managed station to the ME for transmission (for example, as of a management service notification). This communication is implemented with the aid of a host embedded controller interface (HECI). Thus the operating system can use a HECI driver to communicate via this HECI to the ME.

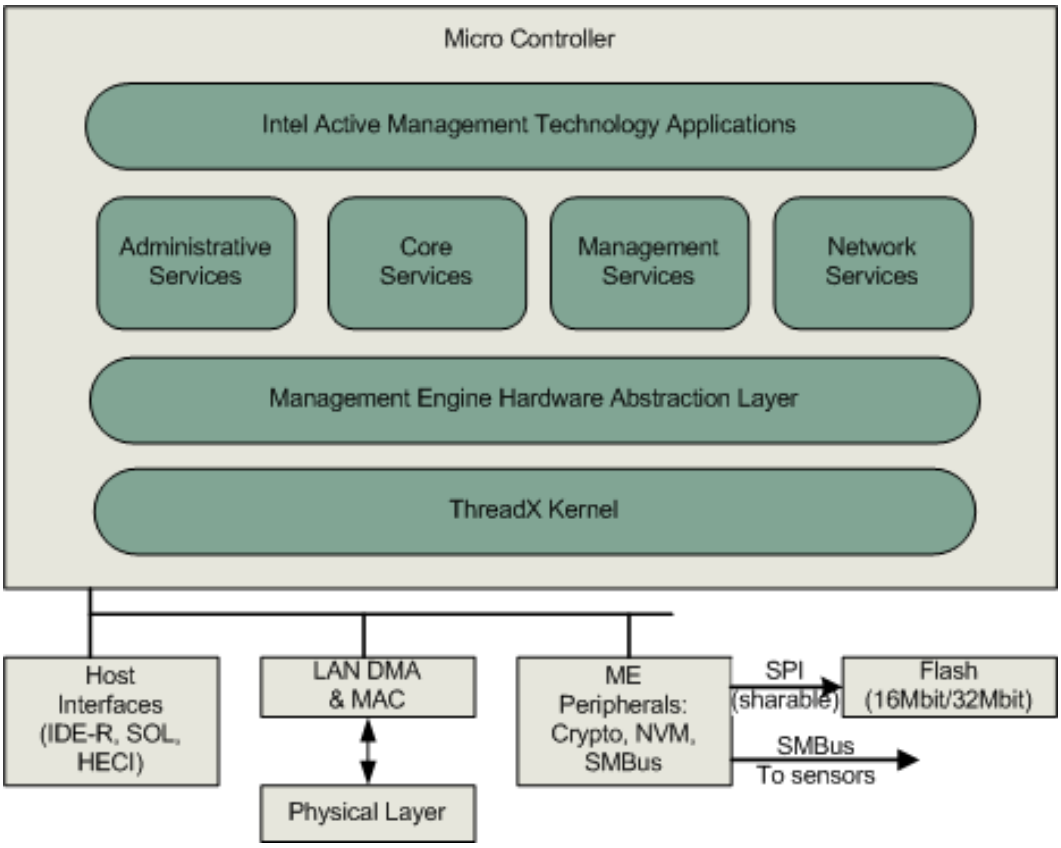

Figure 2.2. Management engine diagram in greater detail (adapted from [37])

2.1.1 System power states

The ME runs independently from the main system's power state, thus it can operate under all six system power states[30]:

- S0 defines a completely powered on and fully operational system.
- S1 to S3 power states are various sleep states.
- S4 is the hibernate state. This state use only a very limited amount of power, hence the system is almost powered off.
- S5 defines a completely powered off system (but attached to a power supply) and is available to the ME and the Wake-on-LAN portion of the network interface(s).

The system power states S0 to S5 can only be controlled over a wired local area network (LAN); since the radio in a wireless network interface card (NIC) is generally not operational in power states other than S0. As a result, the wireless

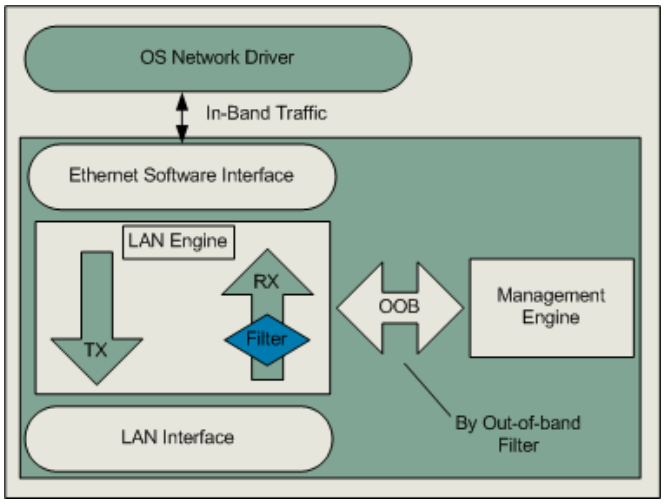

Figure 2.3. Management engine network OOB architecture (adapted from [37])

LAN Intel AMT interface is inaccessible when notebooks are powered down or in low-power modes (sleep, hibernate, or suspend power state modes)[44].

2.2 Interface types

By design AMT supports two types of interface access: local and remote access (via wired and wireless) interfaces. Remote interfaces can send and receive network based traffic via a LAN connection and access the most AMT firmware functions, while the local interface has limited access to the AMT firmware in order to prevent alternations of critical security aspects of the configuration.

2.2.1 Remote access interfaces

Remote access interfaces communicate with Intel AMT via three methods[52]: SOAP messages, Proprietary Redirection Protocol, and WS-Management. Each of these is explained in more detail below.

- SOAP messages based on the XML language the SOAP network protocol are used for transmitting and exchanging AMT related messages with the structure depicted in Figure 2.4:

  An envelope provides the framework for defining what is in the message and how to process it. The header provides a set of encoding rules that denoting instances of application-defined data types. The body contains text that follows a convention for representing procedure calls and responses.

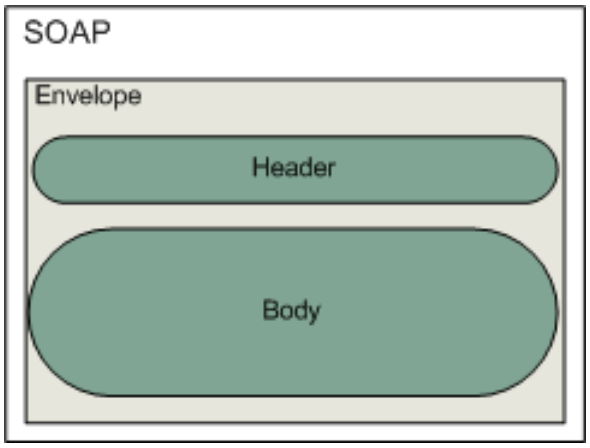

Figure 2.4. SOAP message structure

- The proprietary redirection protocol utilizes serial over LAN and integrated drive electronics (IDE) as well as universal serial bus (USB) redirection capabilities to redirect input and output to the serial interface or the IDE interface to the LAN interface. This allows a remote application to provide serial input (or to receive serial output) and IDE (disk operations) operations. This enables PXE remote booting from a remote CD or disk drive. Both Serial over LAN and IDE redirection use a proprietary protocol and can be implemented using Intel AMT software development kit in independent software vendor (ISV) applications.

- WS-Management[23] is a DMTF open standard that manages, accesses and exchanges information in an object oriented manner across supported devices and systems via the network. WS-Management is based on the CIM as extended by Intel to include support for most AMT functionalities by implementing the DASH specification. An overview of the standard management protocol stack as defined by the WS-Management standard is illustrated in Figure 2.5. The WS standard specifications used to communicate with AMT are WS-Transfer, WS-Enumeration, WS-Eventing, and WS-Addressing. Each of these is described further below.

WS-Transfer Provides simple operations on a single resource (GET, PUT, CREATE, or DELETE)

WS-Enumeration Provides context specific enumeration based on a given filter by utilizing the PULL operation.

WS-Eventing Provides a subscription to events emitted by a resource.

WS-Addressing Provides a common framework for defining references to resources and defines the basic mechanisms for sending and routing these to the appropriate destination.

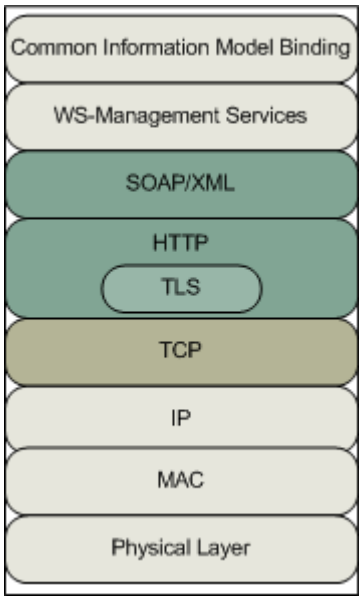

Figure 2.5. Standard management protocol stack

2.2.2 Local access interface

Intel AMT access the local interface by using the WS-Management standard protocol transmitting SOAP messages transmitted over HTTP. This way permitted local applications could communicate with the Intel AMT interface. When a local application transmits a SOAP message destined to the local Intel AMT host name, the local manageability service (LMS) intercepts the request and routes it via the HECI to the management engine for further processing. The LMS acts as a proxy service that transfers TCP requests (open/close connections and TCP packets), between the host management applications and the ME as illustrated in Figure 2.6.

The HECI enables the host OS to control other devices (e.g. an on-board fan controller) and provides a bi-directional channel so that the host OS and the management engine can initiate and complete transactions. The I/O controller hub (ICH), often called the south bridge chipset, is used to interface the processor to various I/O devices. It exists in several versions (shown later in Table 2.4), when referring to a generic version we write ICHx (as in Figure 2.6).

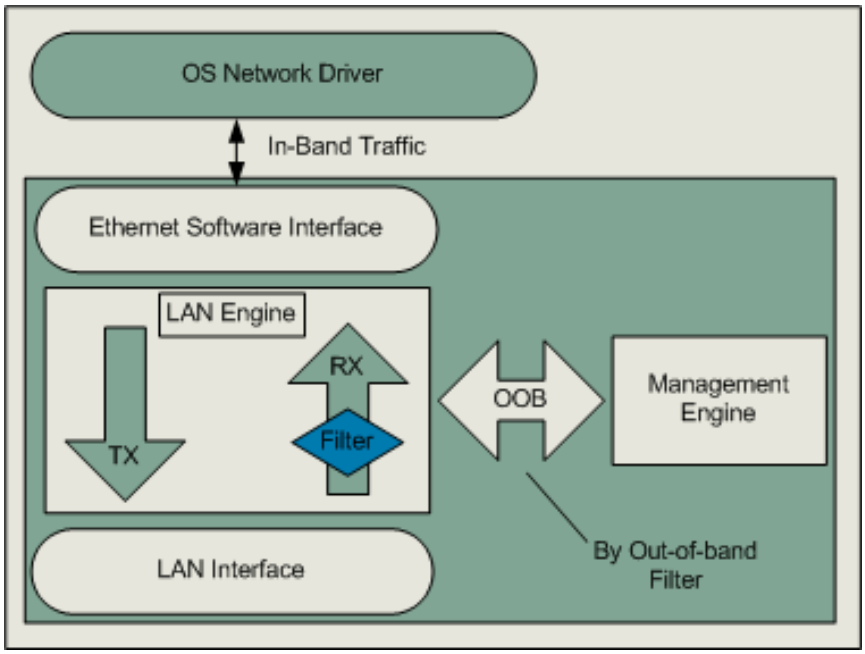

Figure 2.6. Routing local requests to Intel AMT (adapted from [52]

2.3 Architecture features

Intel AMT functionality can be categorized in terms of use cases[52] for a variety of remote network management tasks. The common tasks are discussed in the following section.

2.3.1 Discovering IT assets

Since the ME is able to respond to network traffic even when the host is powered off, it is possible for a network management system to query the AMT enabled device (typically a PC) and get updated information regarding its configuration. This configuration is stored in a local non-volatile data storage area managed by the ME. For instance, a remote application could query the network according to a schedule to find out what computers are attached to the network, what their MAC and IP address is, what is the type and the available number of processors, what is the processor's serial number, what OS that is being run, etc.

Some of this information can be stored in advance by the OS when it is running, but the information remains accessible even if the OS is not running. By using the hardware asset interface which runs locally on the platform specific IT infrastructure data regarding hardware and software inventory can be tracked down and audited with ease. Additionally the underlying information can be stored in the non-volatile

data storage of the platform with the aid of the storage interface and the library components.

2.3.2 Remote repair of systems

Using AMT an administrator can remotely control and manage a PC with the AMT technology. This could be used to install a new OS (even on a system that has no OS installed yet), run diagnostics on the computer, etc. The remote access functionality can be used to examine hardware logs. Specific filters could be set up to automatically, trigger an alert to a remote management system. This behavior establishes a watchdog[1] timer that reports when specific patterns of connections to TCP ports occur. This might be used as part of an intrusion detection system (IDS).

2.3.3 Viruses and rootkit protection

Version control of anti-virus and firewall applications is of great importance; especially in an enterprise since the number of malicious programs as well as viruses and rootkits[18] are increasing daily. By using the storage interface of the AMT platform it is possible to track the version number of the protection software that is being used on a given computer, in order to determine that the software is up to date and has the latest virus signatures. Moreover by using the redirection and remote control interfaces this software can be updated regardless of the system state (i.e., the software can be updated even if the computer is initially powered off). Additionally a managed device can be placed in a quarantine state by using the system defense interface to control its network access, thus limiting the communication of this device.

Furthermore by combining the remote agent presence interface with the local agent presence interface different policies and customized log events can be applied with the aid of the event management interface. This is useful when a malicious threat is detected so that an AMT capable system can be automatically isolated from the rest of the network (using the circuit breaker interface) while triggering an array of commands to help remedy the infected system.

2.3.4 Infrastructure

Part of the AMT platform makes use of the security administration, network administration, and network time interface to configure the access control list (ACL) with the appropriate network and security settings in order to adapt the system to

[1] Watchdog timer is a software or hardware timer that provides protection, by triggering an alarm (such as system reset) to the system when an error occurs, for instance network packet flooding. The watchdog facility ensure that the system is in a working state.

the desired configuration to fit the local IT environment. The use of the ACL manages who has access to which functionalities within the Intel AMT platform.

2.3.5 IDE-R and Serial over LAN features

An interesting hardware based feature of AMT is the remote booting operation via the integrated device electronics redirection (IDE-R)[39]. This feature can be used by the IT technical support personnel or the administrator to remotely boot a PC that has software remediation problems (e.g. booting errors) to a known clean state. This is achieved by using the IDE-R functionality in order to boot the problematic PC from an image or medium (e.g. CD-ROM) on a local or remote storage destination. Moreover by using the console redirection via the Serial over LAN[39] functionality the administrator or the IT department can provide support by controlling a PC outside the context of the OS. This feature allows the remotely use of a keyboard and mouse, as well as video console output in order to perform tasks such as BIOS configuring and pre OS maintenance. The redirection network traffic is transmitted over TCP port number 16994 and over TLS using port number 16995 (see Table 2.5).

2.4 Intel AMT Releases

Table 2.4 lists the Intel AMT firmware releases [40], [2], [19] along with their capabilities and the corresponding hardware. These various releases introduce new features and resolve previous bugs. The functions and features of the Intel AMT platform reside in firmware and are supported via different firmware releases shipped on a variety of systems. Note that Intel AMT version 1.0 is based on Intel's Gigabit Ethernet controller[40].

Table 2.1. Intel AMT firmware releases [40],[2],[19]

Version	South Bridge	Chipset
1.0	ICH7	Intel 82573LM/LC
2.0	ICH8	Intel Q963/Q965
2.5	ICH8M	Intel GM965/PM965
3.0	ICH9	Intel Q35
4.0	ICH9M	Intel GM45 or 47/PM45
4.1	ICH9M	Intel GM45
5.0	ICH10	Intel Q45
6.0	PCHM	Calpella, Piketon

2.4.1 ME firmware upgrade

The revision numbering of the ME firmware uses the following format: W.X.Y.ZZZZ, where W stands for platform version, X for major version, Y for minor version, and ZZZZ for build number[50]. The following requirements must be met in order to successfully upgrading the AMT ME firmware[50]:

1. (W): the platform version needs to be the same as the current firmware
2. (X): the major version is required to be the same or greater than the existing firmware. Note that version 2.0 can only be upgraded to 2.1 or 2.2 and similarly version 2.5 to 2.6.
3. (Y): the minor version must be equal or greater than the existing firmware version when the major version (X) is the same.
4. Downgrading to previous firmware versions is not permitted.

Apparently in order to take advantage of the full-featured and secure capabilities of AMT, one must also upgrade to a newer AMT hardware version for instance it is not supported to upgrade the AMT firmware from version 3.x to version 5.x since the system is based on an AMT 3.x hardware, in fact it can be also dangerous performing such an upgrade. Additionally, all firmware upgrades are provided from the original equipment manufacturer (OEM). Any upgrade is subject to the OEM taking responsibility for supporting the newer firmware. In most of cases a version of AMT is shipped along with the BIOS flash upgrades from the OEM. Table 2.2 list some notebook and desktop OEMs along with the corresponding model, chipset, and the latest released AMT firmware revision for the referenced models[53].

In the same format we have also generated Table 2.3 presenting a listing of OEMs supporting AMT on a variety of embedded platforms such as points of sale (POS), kiosks and automatic teller machines (ATM)[58]. Note that the AMT versions listed on Table 2.3 have the latest supported firmware as of August 2009 and not the standard version that is shipped from factory default.

Table 2.2. Intel AMT enabled desktops and notebooks [53]

Type	Brand Model	Chipset	Ver.
D	Acer Veriton L/S/M 670G, M67WS	Intel Q45	5
N	Acer TravelMate 6592	Mobile Intel GM965 Express	4
N	Acer TravelMate 6593G	Intel PM45	4
N	Acer TravelMate 6493,6593	Intel GM45	4
D	Dell OptiPlex 780,960	Intel Q45 Express	5
N	Dell Latitude E4200,4300	Mobile Intel GS45 Express	4
N	Dell Latitude E6400,6500	Mobile Intel 45 Express	4
D	Dell Esprimo P5925,E5925	Intel Q35 Express	3
D	Dell Esprimo E7935	Intel Q45 Chipset	5
W	Dell Celcius W360	Intel Q35 Express	3
W	CESLIUS W370	Intel Q45 Chipset	5
N	Fujitsu Lifebook E8420/T5010	GM45	4
N	Fujitsu Esprimo U9215,M9415,D9515,X9515,X9525	GM45	4
N	Dell Celcius H265/H270	PM45	4
N	Dell Celcius H250	Mobile Intel PM965 Express	2.5
N	Dell Lifebook E8410	Mobile Intel GM965 Express	2.5
N	Dell Lifebook E8410	Mobile Intel PM965 Express	2.5
D	HP Compaq dc7700p	Intel Q965 Express	2.1
D	HP Compaq dc7800p	Intel Q35 Express	3.2
D	HP Compaq dc7900	Intel Q45 Express	5
N	HP Elitebook 2530p,2730p	Mobile Intel GS45 ICH9M	4.1
N	HP Elitebook 6930p	Mobile IntelPM45 or GM45 ICH9M	4.1
N	HP Elitebook 8530p/8530w/8730w	Mobile Intel PM45 Express ICH9M	4.1
N	HP Compaq 2510p,2710p, 6910p,8510p/w, 8710p/w	Mobile Intel GM965 Express	2.6
D	Lenovo ThinkCentre M55p	Intel Q965 Expres	2.1
D	Lenovo ThinkCentre M57p	Intel Q35 Express	3
D	Lenovo ThinkCentre M58p	Intel Q45 Express	5
N	Lenovo M58pX200/X200s,X301	Mobile Intel GS45 Express	4
N	Lenovo T400,T500	Mobile Intel GM45 Express	4.x
N	Lenovo W700	Mobile Intel PM45 Express	4
N	Lenovo X61, X61S	Mobile Intel GM965 Express	2.x
N	Lenovo T61/T61P	Mobile Intel PM965 Express	2.x
N	Lenovo X61	Mobile Intel GM965 Express	2.x
N	LG S510	Mobile Intel PM965 Express	4
N	Panasonic CF-T8,W8,F8(All Models)	Mobile Intel GS45 Express	4
N	Panasonic CF-30Mk3(All CF-30 K or L models)	Mobile Intel GS45 Express	4
C	Panasonic CF-19Mk3(All CF-19 K or L models)	Mobile Intel GS45 Express	4
N	Panasonic CF-52ELN(B/D/F/H)QAM	Mobile Intel PM45 Express	4
N	Panasonic CF-52FLN(B/D)ZAM	Mobile Intel PM45 Express	4
N	Panasonic CF-52HFN(B/D)ZAM	Mobile Intel GM45 Express	4
N	Panasonic CF-52HUN(B/D)ZAM	Mobile Intel GM45 Express	4
D	Samsung Magic Station DB-P70,Z70	Intel Q35 Express	3.x
N	Samsung SENS P55	Mobile Intel PM965 Express	2.x
D	Intel Desktop Board DQ35JO	Intel Q35 Express	3.2
D	Intel Desktop Board DQ45CB/EK	Intel Q45 Express	5.1

D: Desktop, N: Notebook, W: Workstation, C: Convertible, Ver.: Version

Table 2.3. AMT enabled embedded systems [58]

Type	**Brand/Model**	**Chipset**	**Version**
POS	Fujitsu TeamPoS 3624	Intel Q35 Express	3.2
POS	Fujitsu TeamPoS	Intel Q35 Express	3.2
POS	NCR RealPOS 70XRT	Intel GM45 Express	4.1
Kiosk	NCR SelfServ 60	Intel GM45 Express	4.1
POS	NCR RealPOS 80XRT	Intel Q965 Express	2.2
SC	NCR SelfServ Checkout	Intel Q965 Express	2.2
ATM	NCR SelfServ 22,25,26,32,34,38	Intel Q965 Express	2.2
POS	Radiant P1760	Intel GME965 Express	2.6
POS	Radiant P1560	Intel GME965 Express	2.6
POS	Wincor Nixdorf Beetle /S-II plus	Intel Q35 Express	3.2
POS	Wincor Nixdorf Beetle /M-II plus	Intel Q35 Express	3.2

SC: Self checkout

2.5 AMT assigned network ports

Intel AMT platform transmits data over the network by using registered Internet assigned numbers authority (IANA) network ports registered by David T. Hines and Nimrod Diamant from Intel in February 2005 [47]. These ports are listed in Table 2.5. The type of data that is transmitted to the network by and from the AMT device can be categorized as command and response messages, redirection traffic, and system alerts. The TLS and HTTPS network ports usage is optional and is subject to the provision model used (i.e. the enterprise provision model with TLS support).

Table 2.4. AMT registered IANA ports[47]

Service	Ports	Description
amt-soap-http	TCP, UDP 16992	Intel AMT SOAP/HTTP
amt-soap-https	TCP, UDP 16993	Intel AMT SOAP/HTTPS
amt-redir-tcp	TCP, UDP 16994	Intel AMT Redirection/TCP
amt-redir-tls	TCP, UDP 16995	Intel AMT Redirection/TLS

For other OOB services such as ASF or DASH compliant systems the registered IANA ports[47] assigned by Jim Davis and Carl First as listed on Table 2.5. In addition to the IANA registered network ports, AMT uses by default TCP port number 9971 as the configuration service port. This port is used to initiate the configuration process by transmitting hello packets to the provisioning server. This service is also know as remote provisioning, a detailed discussion about the remote configuration process is given in section 3.7.

Table 2.5. OOB registered IANA ports[47]

Service	Ports	Description
oob-ws-http	TCP 623	DMTF OOB web services management protocol
asf-rmcp	UDP 623	ASF remote management and control protocol
oob-ws-https	TCP 664	DMTF OOB secure web services management protocol
asf-secure-rmcp	UDP 664	ASF secure remote management and control protocol

2.6 Configuration methods

Intel AMT supports different methods for provisioning of a client system, three configuration methods exist for implementing the desired provisioning: manually installing and configuring the client, one-touch by using USB disk booting, and remotely by using a zero-touch configuration. The major differences between

these configuration types is that the two first require physical interaction with the AMT client whereas the last does not. One-touch configuration requires that an administrator enters the credential data into the PC being provisioned by booting using a USB memory stick key or by manually entering the necessary configuration data into the AMT MEBx screen.

2.6.1 Zero touch configuration model

In zero-touch configuration mode the provisioning can be accomplished from a remote location since the MEBx contains at least one or more (usually four) third party trusted certificate fingerprints that are provided by default in any AMT platform at the time of production state (i.e., when the BIOS is flashed). The remote configuration process utilizes a setup and configuration SCS server for successfully provisioning and enabling AMT on a PC without physical attendance, thus allowing administrators to deploy and provision systems without visiting each system individually. A special type of certificate is need for remote configuration to take place, a thorough discussion of the zero-touch remote configuration process is given in section 3.7.

2.7 Setup and configuration models

In order to make a concrete security evaluation of how this technology performs we need to understand the setup and configuration models, also know as provisioning models. Table 2.6 shows a detailed listing of the different security schema supported by the AMT platform. There are three different stages of increased security: small-to-medium business (SMB) mode, enterprise mode with no transport layer security (TLS) support, and the enterprise mode with TLS support. Following we are going

Table 2.6. Provisioning models[52]

Feature	**Basic** (no encryption)	**Standard** (no encryption)	**Advanced** (encryption)
Firmware setting	SMB Mode	Enterprise mode (no TLS)	Enterprise mode (TLS)
Provision model	Manual, One touch	Manual, One touch, Remote	Manual, One touch, Remote
Network infrastructure	DHCP or Static IP	DNS and DHCP	DNS and DHCP, CA, AD (opt.)
Client authentication	HTTP digest	HTTP digest	HTTP digest, Kerberos (opt.)
Management traffic encryption	n/a	n/a	TLS using certificates
Secure network authentication	n/a	802.1X, NAC, NAP (opt.)	802.1X, NAC, NAP (opt.)
Client configuration maintenance	One-to-one	One-to-many	One-to-many

to present and describe how to active the SMB provisioning model on a PC equipped with the Intel AMT technology. We explain the SMB model extensively since it requires less effort to be configured and with no additional requirements such as a DHCP server, a DNS server and a certification authority (CA). Additionally we demonstrate how easily a malicious person could enable the AMT technology, when the physical security of the PC is compromised (i.e an unattended PC).

2.7.1 SMB provision model

The SMB mode is implemented in order for smaller businesses to take advantage of the Intel AMT features without depending on independent software vendors (ISVs) or third party software solutions. It is the most basic form of the setup and configuration models listed in Table 2.6. For this mode there is no requirement for network infrastructure services, but one rather simple configuration task through the MEBx configuration screen[35]. The steps necessary for configuring an Intel AMT capable PC from factory default mode are described below. After successfully connecting the required peripherals and devices (e.g. power supply, keyboard, mouse, and video) and powering up the system, the AMT MEBx will appear on the screen prompting the user to press the CTRL-P keyboard combination as displayed in Figure 2.7.

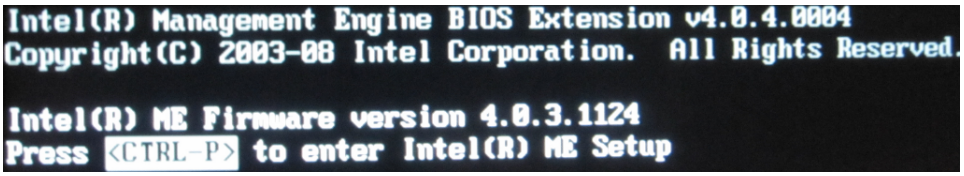

Figure 2.7. AMT MEBx post screen

By entering the keyboard combination CTRL-P the user enters the main menu of the Intel AMT MEBx screen as displayed in Figure 2.8. To successfully configure the setup in SMB mode the following steps must be performed[35]:

1. Entering the default password "admin" to the password prompt.
2. A new screen is displayed that requests you to change the (default) password to a new acceptable value as will be discussed in section 3.5.4.
3. Enter the AMT configuration menu and define an appropriate host name for the configured system; as shown in Figure 2.9.
4. Enter to the TCP/IP sub-menu to select between DHCP (enabled by default) or static IP addressing scheme according to the desired LAN configuration parameters.
5. Enter the provision model sub-menu and select the SMB mode.
6. Finally, exiting the AMT MEBx allows changes to be take place (this requires a system power-cycle)

Figure 2.8. AMT MEBx main menu screen

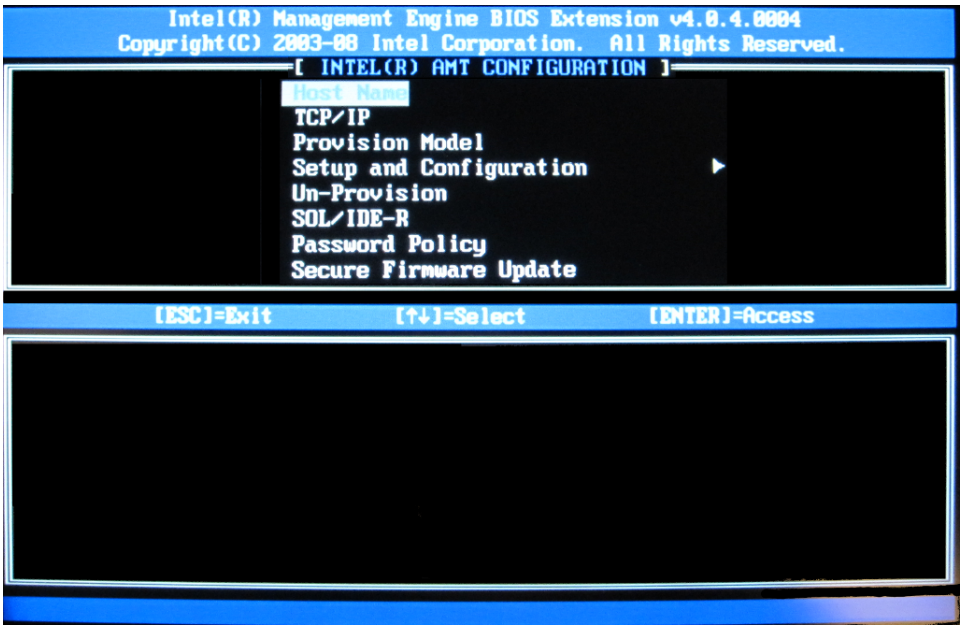

Figure 2.9. AMT MEBx configuration menu screen

2.7.2 Connecting to an Intel AMT device

The built-in web server on Intel AMT can be accessed through the LAN or Internet connection via a web browser that supports the HTTP digest access authentication scheme and JavaScript functionality. To connect with the AMT built-in web server one must enter the following URL listed on 3.3. Figure 2.10 illustrates the login web page of the AMT built-in web server.

```
http://amtdevice:16992
```

Listing 2.1. Built-in AMT web server URL

Note that the "amtdevice" must be replaced with the appropriate configured host name or IP address of the AMT system. It is important to mention that Intel AMT denies any local host access to the built-in web server located on the same machine due to a restrictive policy concerning local services. The applications running within the OS are not considered trusted for AMT, as stated in [13]:

> "by restricting local access to services, we also limit the possibility of rogue applications trying to attack Intel AMT using the local interface." [[13]; page 192]

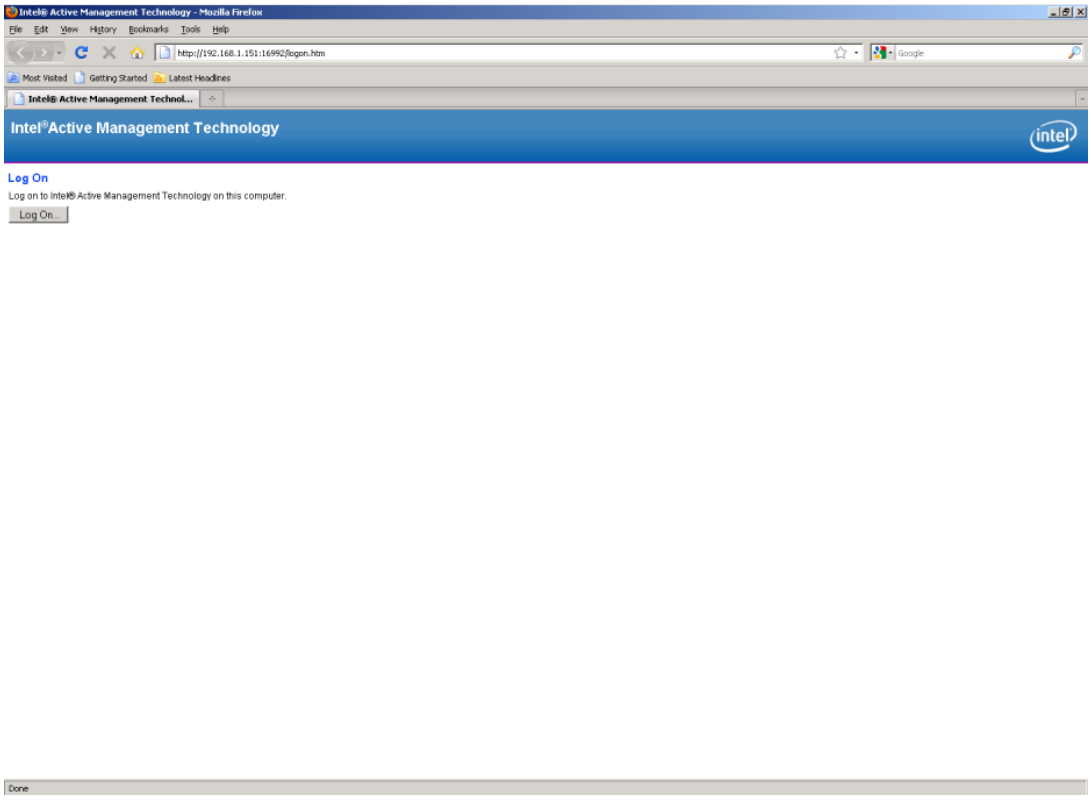

Figure 2.10. AMT web UI login page

Chapter 3

Security analysis of Intel AMT

3.1 Chapter overview

In this chapter we assess and evaluate the security consideration of Intel's AMT taking mostly network approach, since our main research evaluation is based on remote attacks vectors since there is almost not related work (see section 1.2). We start our security analysis by describing a way to bypass Intel's AMT local interface access restrictions. Then we quote statements from Intel's AMT software engineer, Intel's development forum presentation as well as parts from Intel's security configuration guide manual regarding the SMB provisioning model vulnerability.

3.1.1 Lab environment

For our security evaluation we have build a laboratory environment based on our test hardware; a Fujitsu Siemens lifebook T5010 series notebook with AMT version 4 based on the GM45 Express chipset with a revision 9 I/O Controller Hub (ICH). Additionally we have configured several virtual machines in order to successfully evaluate the AMT technology in a variety of OS such as GNU/Linux, FreeBSD, Windows as well as different architectures. Moreover we have chosen open source applications and freely available tools on Intel's website[51] to demonstrate that our attacks vectors could be orchestrated effectively, while keeping a tight budget.

We have implemented the specified security lab in order to experiment with different security software and test a variety of attack methods, while keeping it on a safe environment separate from the production network (i.e. our work IT environment). The specific notebook was chosen based on the network interfaces cards; wired and wireless thus we could evaluate both wired (standard version) and mobile version of AMT. Additionally the cost of the selected hardware is affordable for mainstream users to buy showing that this notebook as well as similar priced PCs on the market (see Table 2.2) could be acquired giving a large magnitude of order.

3.2 Bypassing Intel AMT's local access restrictions

In this section we demonstrate a trivial way of bypassing the Intel AMT protection mechanisms that prohibit access to the AMT built-in web server via the local host. By connecting a universal serial bus (USB) to Ethernet adapter to an available USB port we can add an extra Ethernet port which is **not** managed by Intel AMT LMS, thus once connected to the LAN or Internet one can access the built-in web server from the same local host regardless of the host name. Using a cheap USB to Ethernet adapter we trivially overcome the restrictive policy restricting Intel AMT from access by local applications.

We tested this with a 9.04 Ubuntu Linux release and found that this requires no additional drivers or even administrative (root) privileges to configure or enable the network device (USB to Ethernet adapter) to successfully access the built-in Intel AMT web server. As of December of 2009 the web statistics for Ubuntu's OS share indicate a percentage between forty and fifty percent as stated in[69] and [25]. Nonetheless bypassing Intel AMT's local access restrictions is classified as a vulnerability flaw that the system is exposed to. Usurping privileges on the user's system is defined as allowing unauthorized access to system control functions[65].

3.3 SMB setup mode vulnerability

> A security vulnerability is a flaw in a product that makes it infeasible -even when using the product properly- to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, or assuming ungranted trust. [[14]]

According to Intel's security configuration guide[52] SMB mode is considered the most insecure configuration type for AMT. This guide states:

> (section 3.1) Vulnerability: The Small Business setup, which does not support TLS-based communication, is used when sufficient infrastructure is not available to support the recommended Enterprise setup.[52]

Additionally on February 2008 Ylian Saint-Hilaire a senior software engineer, author of the Intel AMT book[13] and architect at Intel Research Labs in Hillsboro, Oregon wrote on his blog:

> Allow TLS in SMB mode. This is a long time feature request that is somewhat related to the first issue. In my work with Intel AMT, I can do everything I need to setup TLS in SMB mode except enabling it. Allowing administrators to setup server-side authenticated TLS would be very easy to add to Intel AMT and would provide improved security with almost no work. In fact, Intel AMT Commander could just prompt the administrator on first connect if he or she want to enable TLS when

> a non-TLS SMB computer is found. A new root certificate would be generated if none already exist. Strictly speaking, it would not provide "bank level" security, but would go a long way for shops, schools, small business owners that have more to think about than understanding secure manageability.[76]

Furthermore in the Security FAQ for Intel vPro Technology there is a short description of the risk of not using TLS:

> Both authentication credentials and data between configuration/-management console or web client and an Intel AMT machine is traversing network in a clear text and may be eavesdropped. Also, a rogue machine may be put on a network to receive profiles with credentials from the configuration console.[8]

3.3.1 SMB countermeasures

The following countermeasures for this vulnerability are given in Intel's security configuration guide manual:

> Enterprise setup is designed to serve the needs of large organizations. When supported with the proper network infrastructure services, enterprise setup can provide automated one-touch setup and configuration for Intel AMT platforms.[52]

As we have previously quoted from Intel's security configuration guide[52] the SMB model is vulnerable *by design*. However, as presented at the latest (2009) Intel's developer forum[6] the SMB provision model has a highly important role. In particular, we highlight the following from the presentation[6]:

- SMB is second largest client segment and the fastest growing.
- SMB is a large opportunity and should be taken seriously.

3.4 IDE-R and Serial over LAN vulnerability

The IDE-R and Serial over LAN features in AMT functionalities are subject to a serious implementation vulnerability when they are transmitted in the clear (i.e., when not using TLS). Surprisingly this is a known issue for Intel as Ylian Saint-Hilaire wrote on his blog in February of 2008:

> No TLS, Serial-over-LAN/IDE-R password in the clear. As many of you have discovered, when using Intel AMT in small business or enterprise mode without TLS, the login username and password is sent on the network in the clear when the administrator performs a serial-over-LAN or IDE redirect operation. With so many coffee shops, schools,

> Internet cafes playing around with Intel AMT features, this could be a big problem. Imagine a classroom with a few vPro computers with AMT setup in SMB mode by an unsuspecting teacher. A student running a packet sniffer, obtaining the password and rebooting AMT computers remotely.[76]

This weakness of Intel AMT remote IDE and Serial over LAN transmitting administrator credentials in clear text is depicted in Figure 3.1. Along these lines, why does Intel AMT permit use of this architecture *without* a TLS implementation? This seems to be a poor decision since it is a great security vulnerability, especially considering that it is being used for such a crucial operation; remote management. Blindly trusting the end-users and administrators, that would not even reviewed thoroughly the security documentation of Intel AMT.

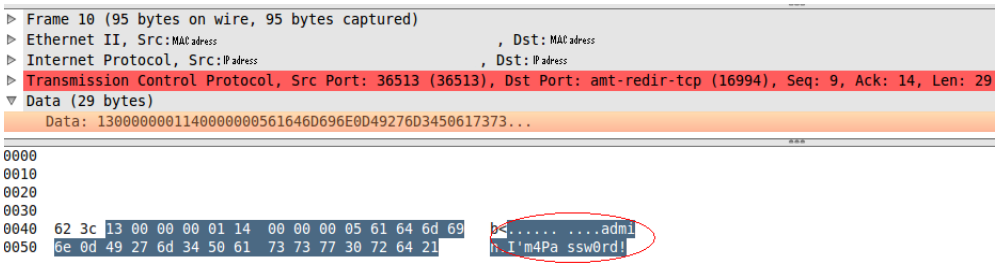

Figure 3.1. Administrator credentials transmitted in clear-text

3.5 HTTP digest authentication scheme

3.5.1 Introduction

HTTP is a request and response protocol[63] widely used on the WWW for exchanging data, normally HTML files (web pages), across different architectures via the Internet, LAN, and WAN. HTTP offers support for two authentication schemes[48]: basic access authentication and digest access authentication. The former is the most popular authentication method since almost all web browsers support it. Digest access authentication was implemented in order to supersede the basic access authentication scheme which uses an insecure way of transmitting credentials by using Base64 content transfer encoding; a simple algorithm to encode and decode text[59].

Digest authentication uses message digest 5 (MD5) cryptographic hashes in addition to nonce values to implement more secure authentication than the basic access authentication method, which transmits credentials over the network in the clear. When a network session is eavesdropped the username and password can be trivially decoded. Below we demonstrate a serious implementation fault illustrating

the poor choice Intel made when selecting the digest access authentication method for the AMT platform. This insecurity applies also to a wide range of web servers on the Internet, as well as session initiation protocol (SIP) devices and services that are using HTTP digest authentication.

3.5.2 How digest access authentication works

HTTP network packets are composed of a header and an entity (i.e., payload) fields. Digest authentication scheme is based on a challenge-response protocol[48], when the client requests a web page that requires authentication without providing any credentials in the clear, the server sends back a reply with a status message (401) "authentication required", providing the authentication realm, the available "quality of protection" (qop), the opaque value that must be returned to the server to match the challenge with the client's response, and a cryptographic nonce. Then the client submits the user's credential in an MD5 hashed response along with the nonce, opaque value, universal resource identifier (URI), authentication realm, a nonce counter, the selected qop, and the client's nonce value.

Finally the server compares the response with it's own computation which is a hash normally stored in a local database and grants or deny access accordingly. Below we show how the digest response value is computed. First the A1 hash is calculated:

```
A1= MD5(usename:realm:password)
```

Listing 3.1. A1 hash calculation

Then the A2 hash is computed:

```
A2= MD5(http method:URI)
```

Listing 3.2. A2 hash calculation

Finally the response is computed:

```
response= MD5(A1:nonce:nonce counter:client nonce:qop
    directive:A2)
```

Listing 3.3. response hash calculation

Note that the colon punctuation mark (:) is part of the hashing computation. An overview of the HTTP digest access authentication scheme is illustrated in Figure 3.2. Additionally an example (taken from[48]) is shown in listing 3.4 where an access protected HTML file is requested from the web server via a HTTP GET request. As previously discussed the first time that the client requests the file no authorization occurs and the server sends the response shown in listing 3.4 in order to instruct the client to use the digest access authentication scheme.

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
realm="testrealm@host.com",
qop="auth,auth-int",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Listing 3.4. Initiation of HTTP digest access authentication – server response[48]

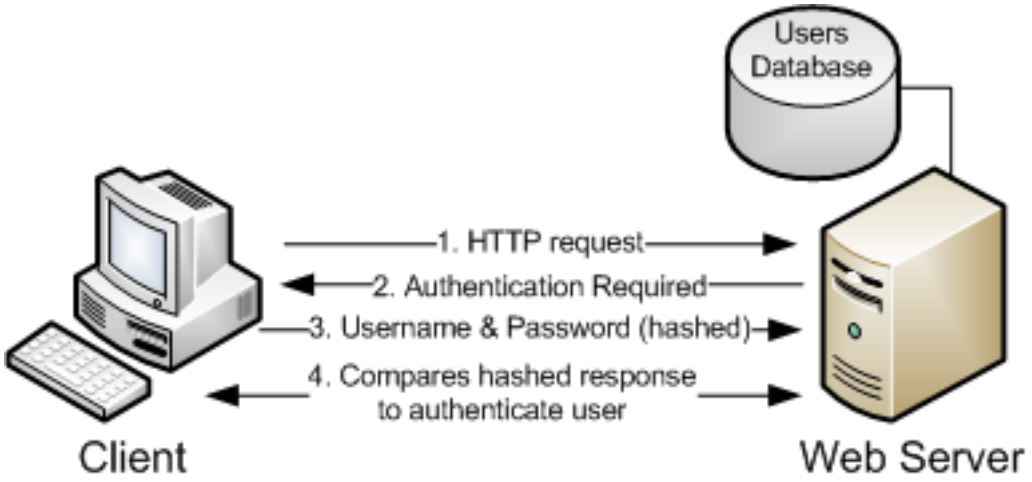

Figure 3.2. Digest access authentication schema

Next the client is to enter the appropriate credential data (username and password) which are valid for the requested resource. Listing 3.5 shows the client request for the HTML file with the authorization header added.

```
Authorization: Digest username="Mufasa",
realm="testrealm@host.com",
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
uri="/dir/index.html",
qop=auth,
nc=00000001,
cnonce="0a4f113b",
response="6629fae49393a05397450978507c4ef1",
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

Listing 3.5. Client request with the authorization header on HTTP digest authentication[48]

3.5.3 How Intel AMT handles HTTP digest access authentication

Intel AMT client authentication[48] is based on the digest access authentication scheme in basic and standard provision models where no standard encryption applies, as explained in section 2.6. Contrary to modern security practices Intel AMT provides only conformance with the default RFC 2617[48] that dates back eleven years for their implementation of authentication for the embedded web and XML server. As stated

> An off-line brute force dictionary based password attack can occur in the case of an eavesdrop attacker (MIMT). Since the user's credentials are transmitted as an one way MD5 hashed key on a HTTP message to the server. The server can mitigate this type of attack by not allowing users to select passwords that are in a dictionary.[48]

Additionally we enumerate the following security considerations (listed along with the section in RFC 2617 where this issue is addressed):

Section 4.2 Digest Authentication does not provide a strong authentication mechanism, when compared to public key based mechanisms. Additionally, it offers no confidentiality protection other than protecting the actual password. All of the rest of the request and response are available in plain text. Digest Authentication offers only limited integrity protection for the messages in either direction.

Section 4.7 An attacker that can eavesdrop can test any overheard nonce/response pairs against a list of common words to perform a dictionary attack.

Section 4.8 Both Basic and Digest authentication are vulnerable to "man in the middle" (MITM) attacks. A MITM attack has all the problems of eavesdropping -while offering some additional opportunities to the attacker.

Section 4.13 Stored passwords: If the password file is compromised, the attacker can access documents on the server within this realm.

Things have changed during last decade when brute force password based attacks where computationally infeasible due to computation power and memory limitations. Technological advances along with more sophisticated techniques have been introduced, such as distributed computing[24], on-line databases of precomputed hashed keys[56], field-programmable gate arrays (FPGA's), pre-computed dictionary attacks, custom hardware, and graphical processing unit (GPU) accelerated attacks for example nVidia's CUDA[60] technology.

3.5.4 Intel AMT password policy

In this section we will examine the password strength and limitations of the Intel AMT password policy. In factory setup mode (in the default configuration) the

default password in Intel's management engine BIOS extension (MEBx) is "admin" when the platform is in a non-provisioned state. Obviously an adversary that gains physical access to a computer with AMT capabilities can compromise the system by entering MEBx (with the CTRL-P key combination) and typing the default password ("admin"). Then a malicious person can configure the AMT as required to implement his attack schema at will. Note that the AMT technology is enabled by default in most PCs BIOS, so people with no knowledge of the AMT technology suffer from this default password policy. The setup mode password policy that MEBx uses is the following:

- Password length can be between 8 and 32 characters long
- The password must have both upper and lower case Latin characters
- It must have at least one numeric character
- It must have at least one ASCII non-alphanumeric character (!, @, *, #, $, %, ^, &,)
- In general 7-bit ASCII characters in the range of 32-126 without the invalid characters (listed below) can be used in the password.

The following characters are considered invalid and not allowed in the password policy of AMT:

- " (double quotations)
- . (period)
- , (comma)
- : (colon)

Note that the underscore character (_) and the space character are not counting to the password complexity according to the password policy of AMT.

3.5.5 Exhaustive password policy

Using this policy the password length is more interesting than the password complexity policy requirement, since it reduces the total number of available passwords (in terms of the maximum entropy). As a result password cracking is a big concern, especially for enterprises. Note that a sophisticated attacker may save time by avoiding passwords that do not meet the policy criteria, thus making cracking process more efficient. Such a policy restriction suffers from a rainbow table cracking attack[62] based upon generating only the potential reduced password set by applying the complexity policy requirements to the initial set of possible strings.

3.5.6 International keyboards on AMT MEBx

AMT MEBx has an implementation fault[50] when non-US keyboards are used. MEBx assumes that everything is typed on a QWERTY type US keyboard. Therefore, different keyboards mappings, for instance QWERTZ or AZERTY, are treated as if they were US keyboards. As a result tremendous problems arise since users have to blindly enter the password and trust what they type on the keyboard. Entering the same sequence of keys will work, but will not when the OS level is running - due to the use of language locales.

For instance if a user is typing in more than two languages, then changing to another language and typing the password for the AMT platform will not work as long when the user enters the password using the local keyboard mapping. Ending with an absurd recommendation from Intel[68] as illustrated in Figure 3.3.

> Passwords using the A,M,Q,W,Y and Z keys can cause problems and are **not** recommended[68].

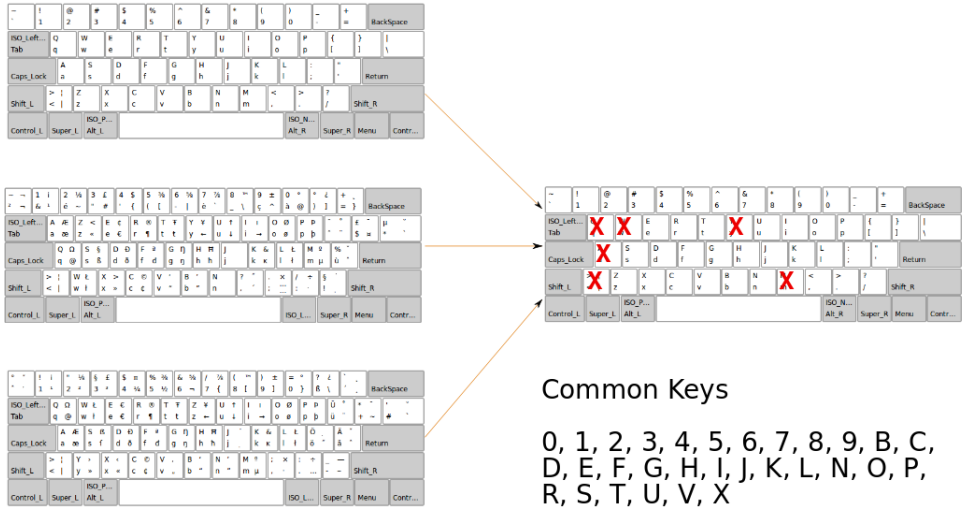

Figure 3.3. MEBx common keys (adapted from [68])

3.5.7 Keyboard mapping implementation fault

The keyboard mapping implementation fault of AMT assumes that only US keyboards are used, this implies that end users and system administrators need

to obtain a US keyboard and stick to the US language locale on the OS and change their typing habits to match the AMT implementation fault. This is a known issue since May of 2008[68], there is still no released fix. Additionally when changing the password on the Intel manageability platform via web access (e.g. web browser) the administrator credentials are not synchronized with the MEBx[50].

This can be an issue since every time that the credentials are changed the Intel AMT platform needs to be re-provisioned for the credential change to take effect. If the password is changed via web access, but the machine is re-provisioned then the Intel AMT is going to use the old credentials that were entered when the MEBx was configured. Thus new credentials should be instantiated inside the MEBx after the last stage of the post BIOS boot phase or on the first remote configuration provisioning, details presented in section 3.9 regarding the AMT's privacy protection mechanisms. Moreover since usernames and passwords are involved a database is required to keep track of all of this information, thus enlarging the insecurity factor as more data needs to be securely managed[54].

3.5.8 Password-based authentication to Intel AMT: attack scenario

For the attack scenario to be successful, the attacker requires to eavesdrop ***only*** a network packet transmitted over the wire. Specifically a transmitted response or request of the HTTP session is needed while the administrator is remote managing the AMT platform. Next the attacker obtains the HTTP transmitted packet (see Figure 3.5), most commonly by sniffing the networked traffic. From this point the attacker can perform a brute-force password attack to recover the administrator's or other privileged user's credentials as discussed in section 3.6. Most important, this attack can be implemented in two out of the three setup and configuration models of Intel's AMT technology (i.e. basic and standard).

3.6 Cracking process

3.6.1 John the ripper patch

John the Ripper (JtR) is an active password cracking tool[9] that is used for different cipher algorithms and offers support for numerous patch scripts and different OS system and processor architectures as well as a variety of brute force attack methods for recovering passwords mostly by providing password hash strings. Licensed under the GNU general public license (GPL) version 2 and can be downloaded from the openwall project website[9]. It is an open source code community driven project where developers and users can implement and contribute additional cipher-text formats, optimization, and implementation tweaks as software patches and scripts. In our case we have used the HTTP digest access authentication (HDAA-MD5) patch contributed by Romain Raboin[67] based on the RFC 2617[48] standard.

As previously discussed AMT uses the RFC 2617[48] standard unmodified as the main and only access authentication method while transmitting data over the network to and from the AMT platform. When TLS is not applied as the security configuration model (i.e. SMB and standard provision models) a network capture of a transmitted packet from an eavesdropped connection is sufficient for utilizing an off-line brute-force attack, to successfully recover the administrator or management account password.

3.6.2 Patch for AMT

The AMT platform includes the ":" colon character as part of the realm directive value on the HTTP transmitted packet as depicted in listing 3.6.

```
realm="Digest:8DC54871523E45648425F8782646E734897428B3"
```

Listing 3.6. HTTP digest realm directive example of AMT

JtR detects automatically the cipher text hash type and splits the user name from the password hash by using the ":" colon character as a delimiter. Using this knowledge we have implemented a patch based on the HDAA-MD5 cipher text plugin[67] (see listing 3.7) to provide an AMT patch. The resulting patch can easily be applied to John the Ripper version 1.7.4.2 along with the jumbo patch that performs many updates on the JtR main engine[1].

[1]Note that only the file HDAA_fmt.c is affected by this patch, thus one can apply the AMT patch after the Romain Raboin[67] HDAA-MD5 plugin has been installed.

```
--- john-1.7.4.2-jumbo-2.orig/src/HDAA_fmt.c     2010-02-05
    16:21:04.000000000 +0000
+++ john-1.7.4.2-jumbo-2/src/HDAA_fmt.c 2010-02-05
    15:46:53.000000000 +0000
@@ -289,6 +289,7 @@ static void                    *hdaa_salt(
    char *ciphertex
         int             nb;
         int             i;
         char            **request;
+        char            *AMTrealm;
         char            *str;
         reqinfo_t       *r;
 #ifdef __MMX__
@@ -308,7 +309,6 @@ static void                    *hdaa_salt(
    char *ciphertex
                         nb++;
                 }
         }
-
         /* calculate h2 (h2 = md5(method:digestURI))*/
         str = malloc(strlen(request[R_METHOD]) + strlen(
            request[R_URI]) + 2);
         sprintf(str, "%s:%s", request[R_METHOD], request[
            R_URI]);
@@ -319,7 +319,10 @@ static void                    *
    hdaa_salt(char *ciphertex
         memset(conv, 0, sizeof(conv));
         bin2ascii(h2);
         free(str);
-
+        /* When the HTTP digest message contains the : (JtR
    field separator) as part of the realm section. Convert _
    to : */
+        if ( NULL != ( AMTrealm = strstr(request[R_REALM], "
    Digest_") ) )
+                        AMTrealm[6] = ':';
+
         /* create a part of h1 (h1tmp = request:realm:)*/
         snprintf(r->h1tmp, HTMP - PLAINTEXT_LENGTH, "%s:%s:"
            , request[R_USER], request[R_REALM]);
```

Listing 3.7. Intel AMT JtR patch

In order for the patch to work successfully one must change the colon (':') character of the AMT realm to the ('_') character. Hence the patch checks for the colon character and changes it appropriately. To apply the patch on the GNU/Linux OS one must use the command as shown in listing 3.8.

```
patch -p1 < ../john-1.7.4.2
```

Listing 3.8. Apply JtR source code patch

The command shown listing 3.8 strips one leading directory name from the pathnames specified in the patch file[61]. Note that one must apply the jumbo patch or the HDAA-MD5 plugin[67] *prior* to the AMT patch. An excellent resource that describes in detail how to apply patches to the JtR source code can be found on the openwall community website[61].

3.6.3 Creating the password string

In order to create the password string for use with the HDAA-MD5 plugin[67] we use a single captured HTTP packet based on Intel AMT network traffic (see listing 3.5). The password string syntax for use with JtR must be defined as shown in listing 3.9.

```
user:$MAGIC$response$user$realm$method$uri$nonce$nonceCount$
     $ClientNonce$qop
```

Listing 3.9. HDAA-MD5 password string syntax

By default the dollar sign ('$') is the field delimiter for the password string; however, it can be changed in the HDAA_fmt.c file. After applying the AMT patch one can use the syntax shown in listing 3.10. In this example the value of the magic string represented as $MAGIC$ in listing 3.9 is $response$.

```
admin:$response$9a54e484f89b2f68521f9e49fbf4b3ae$admin$
Digest_8DC54871523E45648425F8782646E734897428B3
$GET$/dir/index.html
$dcd98b7102dd2f0e8b11d0f600bfb0c093$00000001$0a4f113b$auth
```

Listing 3.10. HDAA-MD5 AMT specific password string syntax

3.6.4 Results

Here we present some performance tests based on the JtR benchmarks of how fast the HDAA-MD5 cipher-text performs. The benchmarks are based on a standard GNU/Linux distribution Ubuntu 9.10 with non-optimized and non-tweaked versions of the GNU project C compiler version 4.4.1 and the message passing interface (MPI)

based on mpich2 version 1.2. The MPI spreads the workload simultaneously to multiple cores. The benchmarks have been measured on the systems shown in Table 3.1. The benchmark are based on JtR version 1.7.4.2 with the jumbo patch. In Table 3.2 we summarize the benchmark reports, both benchmarks were compiled for the x86-64 bit architecture with support for the streaming single instruction, multiple data (SIMD) extensions 2 (SSE2) instruction set. The single instruction/multiple data stream processing mode enables a single instruction to act on multiple data streams at one time. The GNU C compiler (referenced above) will automatically generate vectorized code when the target machine supports SSE2.

Table 3.1. Benchmarked systems

Vendor	Baseboard	CPU	Cores
Gigabyte	GA-MA74GM-S2H	AMD Athlon 64 X2 5200+	2
Intel	MFS5520VI	Intel Xeon E5530 2,40GHz	8(16 HT)

HT: Hyper-threaded

The tested systems are multi-core CPU systems and the password combinations per second (c/s) are shown for one CPU core. The total maximum number of password combinations per second can be achieved with proper CPU parallelization. In our benchmark (see Table 3.2) the crack progress of Intel Xeon E5530 at 2.4 GHz CPU performance is marked significant than AMD Athlon 64 X2 at 2.7Ghz due to the 8 physical hyper-threaded cores, it is almost 16 times faster than a single core, as it takes advantage of otherwise-idle execution units. Additionally the brute-force cracking performance can be boosted tremendously if optimizing compilers are used.

Table 3.2. JtR benchmarks

CPU	Time	Combinations/s	Total c/s
AMD Athlon 64 X2 2.7Ghz	200 s	1,421,000	2,842,000
Intel Xeon E5530 2.4GHz	200 s	1,380,000	22,080,000

3.6.5 GPU cracking scenario

Recently, it appears that GPUs have became efficient in providing cost effective ways, to recover (crack) a key or a password with an enormous speed capacity. A variety of security projects introduced the use of GPUs in order to increase the magnitude of order in password cracking attacks, especially in the case of brute force attacks. A notable project is BarsWF[3] calculates 350 million keys (combinations) per second. The result is based on a nVidia 9600GT/C2D 3Ghz CUDA version on a single machine! Additionally IGHASHGPU release provides a recovery speed on ATI HD4850 that peaks at 955 million keys (combinations) per second.

A likely scenario would be a distributed cross-platform password recovery project based on GPU acceleration for brute forcing password attacks. Since the cost of this implementation for a large corporation that focuses in password recovery of hashes is of relatively high importance such as large scale organizations, governments and multinational companies. The incentive for utilizing a distributed brute force password attack could potentially derive a variety of malicious activities such as gathering intelligence, compromise a resource, access classified databases and assets, etc. The outcome of a distributed GPU accelerated password recovery attack would be of an increasingly high magnitude of order. A rough estimate calculation of 100 machines that could compute around 500 million (combinations) keys per second, based on one or two powerful GPU cards could provide a potential of 50,000 million (combinations) keys per second.

3.7 Remote provisioning

An interesting use case of Intel's AMT was introduced in section 2.6. That use is remote configuration without any physical attendance. This is known as zero touch remote provisioning or bare-metal provisioning. As the name implies the goal is an automated set up and configuration process that applies to almost every non-provisioned or new (factory default) AMT capable computer when connected to a network[32]. Detailed information will be presented in this section.

3.7.1 Introduction

There are many ways to provision AMT capable computers, but the most interesting is the zero touch remote provision method, also know as bare-metal remote provisioning. The important characteristics of this provisioning method is that:

1. It can occur though a wired network connection, typically a LAN.

2. There is no need for physical interaction with the computer, it simply requires that the computer have electrical power and a network connection.

Normally Zero touch configuration (ZTC) process takes place outside the context of the OS, since this feature was created to enable remote provisioning AMT systems prior to OS installation and without any physical presence being needed. A typical example is to deploy one thousand AMT systems by simply taking the machines out of their packages and connect them to a power supply and connecting them with an Ethernet network cable to the LAN, then deploying an OS or other boot image to all of the Intel AMT clients.

Remote provision process

By default Intel AMT uses a timer used to initiate or limit the network interface, allowing the SCS to configure a device without any physical attendance or software

based support. The ZTC process starts at the first boot of the system by requesting an IP address lease from the DHCP server pool along with other local configuration information, such as the host name, the default gateway, and one or more DNS servers on the attached network. An overview of the ZTC provisioning process is illustrated in Figure 3.4. The basic steps in this AMT provisioning process are:

- An AMT enabled system is plugged to power and connected to a wired LAN or WAN.
- The AMT platform transmits a DHCPDISCOVER[1] request packet containing the following information:
  - The source MAC address is set as the client's MAC address (remember that the AMT shares a common MAC address with the PC).
  - The destination MAC address is set as a network link local broadcast address (FFFFFF-FFFFFF).
- The DHCP server listens for a DHCPDISCOVER request and replies with a DHCPOFFER[1] packet containing a potential IP address for this host.
- Finally the client replies back with a DHCPREQUEST[1] packet and requests the appropriate network parameters (IP address, netmask, gateway) that are offered from the DHCP server.
- The DHCP server updates the DNS entries and (if dynamic DNS service is implemented) specifies the DNS domain name (DHCP option 15).
- The AMT device is now in setup mode status and starts transmitting "HELLO" packets to the provisioning server.
- The provisioning server checks for the appropriate fully qualified domain name (FQDN) entries for this AMT host.
- The provisioning server loads the appropriate certificate (the one intended for provisioning) and checks for an appropriate certificate fingerprint of AMT (as was discussed in section 3.7.1).
- The provisioning server configures the AMT device with the desired configuration parameters.

The complete Intel AMT remote provisioning flow for ZTC is illustrated in 3.5.

[1]DHCPDISCOVER, DHCPOFFER and DHCPREQUEST are messages transmitted through out the DHCP configuration process

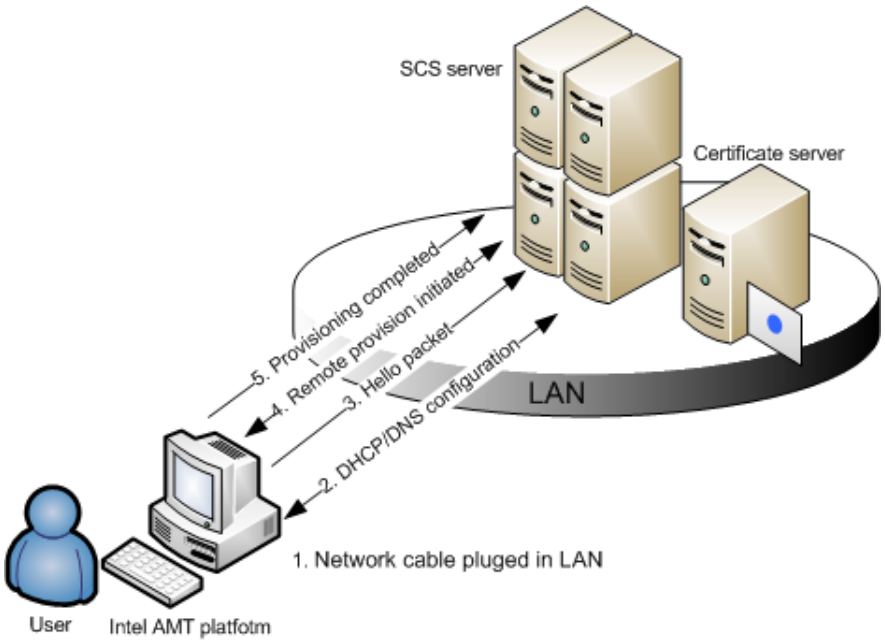

Figure 3.4. ZTC provisioning process

3.7.2 Remote provision certificate fingerprints

During production AMT platforms are equipped with one or more active embedded hashed root certificates (factory default) from various SSL vendors worldwide. These certificates are stored in the firmware image that is stored in the FLASH memory. As per the Telecommunication Standardization Sector (ITU-T) recommendation X.509[17] a root certificate is a public key certificate which identifies the root certificate authority[2] Every signed certificate from a CA vendor includes a hash called a fingerprint. This fingerprint is used to validate that the certificate as originating from the CA vendor and not from a malicious CA. Intel AMT includes a table with SHA1 fingerprints of several different SSL vendors[38] that are trusted CA by default (see Table 3.7.2). Note that the Starfield CA is only available in Intel AMT firmware releases 2.2, 2.6, and 3.2.

3.7.3 Certificate fingerprint

A certificate fingerprint that matches the stored (factory default) fingerprints in AMT firmware image can be used to initiate the remote configuration process.[3]

[2]CA can be described as a network authority that issues, maintains and validates, public keys for message encryption. Next it validates the information submitted (i.e. the certificate requester) in order to issue the requested certificate.

[3]AMT platforms are supported in several different chipsets and architectures types, as well as different firmware revisions. According to [38] AMT firmware versions 2.0, 2.1, and 2.5 are documented as not supporting ZTC.

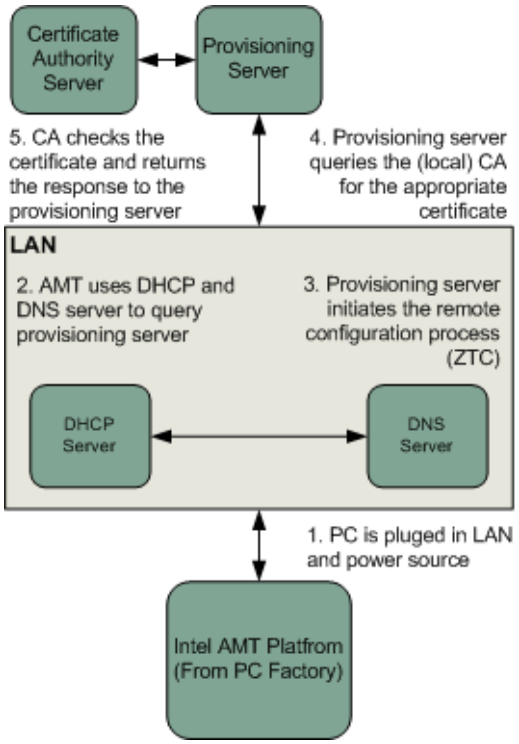

Figure 3.5. AMT provisioning flow

Table 3.3. SSL certificate fingerprints

Certificate vendor	**ID**	**SHA1 fingerprint**
Comodo	OID	d1eb 23a4 6d17 d68f d925 64c2 f1f1 6017 64d8 e349
Go Daddy class 2	OU	2796 bae6 3f18 01e2 7726 1ba0 d777 7002 8f20 eee4
VeriSign class 3 G1	OU	742c 3192 e607 e424 eb45 4954 2be1 bbc5 3e61 74e2
VeriSign class 3 G3	OU	132d 0d45 534b 6997 cdb2 d5c3 39e2 5576 609b 5cc6
Starfield class 2	OU	ad7e 1c28 b064 ef8f 6003 4020 14c3 d0e3 370e b58a

Typically every digital certificate that follows the X.509 standard[64] from version 2 or later contains the following fields:

Version: defines the issued certificate version

Serial number: uniquely identifies the certificate

Signature algorithm: the algorithm used to issue the certificate

Issuer: the organization or entity that issues the certificate, this contains the following values:

- Common Name (CN)
- Organization (O)
- Organization Unit (OU)

Issued to: the person or entity that is identified, this contains the following values:

- Common Name (CN)
- Organization (O)
- Organization Unit (OU)
- Serial Number

Validity: specified the period when the certificate is valid. This has two components:

- Issued on: issued date
- Expires on: expiration date

Fingerprint algorithm: the hash algorithm used to generate the certificate

Fingerprint: the hash of the certificate

Requirements for remote provisioning

To successfully provision an AMT system we need the following components:

1. A specific type of SSL certificate
2. A DHCP and DNS server with specific parameters
3. A provision server, for instance Intel's Setup and Configuration (SCS) service.

The resources needed in order to utilize zero touch remote provision via a LAN connection are freely available from Intel's website[51]. Additionally a deluxe SSL certificate needs to be purchased. In the next section we provide specific details of how to order such a certificate. First we need to clarify the required parameters that are needed for the digital certificates(s) and how the certificate signing request (CSR) procedure operates.

Acquiring a SSL certificate

Acquiring an appropriate certificate is rather easy. As we have explained before we need a deluxe assurance SSL certificate from vendors that are supported by Intel's AMT remote provision process. These CA vendors were listed in Table 3.7.2. In our implementation we have used the GoDaddy as our CA vendor. Although there many vendors that offer certificates intended for Intel AMT remote configuration process we found this vendor to be the least inexpensive and they issued the

certificate immediately after the verification process. Normally an entity (individual or company) that would like to obtain an SSL certificate from CA vendor needs to prove that has the requisite rights or is the owner of the domain that requests an SSL certificate - in order for the vendor to validate this information. The following steps are needed to acquire the special type of certificate that will be used for the remote configuration (ZTC) process.

1. Own a domain name that matched the same entity name (individual or organization) as the name registered for the deluxe SSL certificate order as written on the WHOIS request.
2. Order a single deluxe SSL certificate from GoDaddy.com Inc.
3. After successfully purchasing the certificate, the request is processed automatically.
4. Select the appropriate certificate type: corporate or small business/sales proprietor.
5. Then enter the necessary personal information (certificate requester information) that matches the billing data (in the previous step that we ordered the certificate). Note that the WHOIS directory service information must match the domain name for which the certificate will be issued.
6. Finally generate and submit the Certificate Signing Request (CSR) as explained in section 3.7.4.
7. As part of the individual authentication process (our case), you will be asked to send two forms of identification:
   a) Provide driver's license, passport, or a valid government issued photo ID.
   b) Provide a bank statement or a credit card statement

Note that in all identification fields the requirement is to enter the exact data (even a hyphen matters) that match the entity name (organization or individual), the WHOIS directory service information of the domain that is registered with the certificate, and the billing information when ordering the certificate.

3.7.4 Intel AMT remote provision configuration

There are many options for generating a CSR request, in our lab we have used the OpenSSL cryptographic toolkit[5]. The important data required for the certificate fields are listed below:

- OU must be "Intel(R) Client Setup Certificate".
- CN must be "SystemName".domain.com matching the FQDN of the host that will perform the remote provisioning configuration and will generate the CSR.

- O must match the entity name (individual or organization).

Additionally the following entries must match the domain name WHOIS directory service lookup.

- C: country two-character abbreviation
- ST: state or province (full listing)
- L: locality or city name (full listing)

Generating a CSR request for AMT

To simply the creation of the CSR request we created a configuration file (shown in listing 3.11). Given the configuration file the CSR generation command can be invoked as shown in listing 3.13. This generates a CSR request in privacy enhanced mail (PEM) format. The next step is to export this PEM formatted certificate as a PFX certification[49]. This is done using the command shown in listing 3.12.

```
RANDFILE=./.rnd
default_bits=2048
default_keyfile=keyfile.pem
encrypt_rsa_key=no
default_md=sha1
req_extensions=req_extensions_section
prompt=no
distinguished_name=req_distinguished_name_section

[req_distinguished_name_section]
C=DE
ST=Berlin
L=Friedrichshain
O=Name entity (individual or organization)
OU=Intel(R) Client Setup Certificate
CN=provisionSystemName.mydomain.com
emailAddress=user@DomainName.com

[req_extensions_section]
basicConstraints=CA:FALSE
keyUsage=digitalSignature
extendedKeyUsage=critical,serverAuth,2.16.840.1.113741.1.2.3
subjectKeyIdentifier=hash
```

Listing 3.11. CSRrequest.cfg file

```
#openssl req -new -config CSRrequest.cfg -out
    RemoteProvisionCSR.pem -keyout RemoteProvisionkey.pem
```

Listing 3.12. CSR generation command

```
pkcs12 -export -in RemoteProvisioncrt.pem -out
    RemoteProvisioncrt.pfx -inkey RemoteProvisionkey.pem -
    name "Intel(R) RCFG Certificate" -password "pass:
    aSecurePassword?"
```

Listing 3.13. PFX certificate generation

The File RemoteProvisionCSR.pem contains the CSR request ready for submission to the CA. Simply copying the contents of this file for instance using a text editor by starting from —-BEGIN CERTIFICATE REQUEST—- to the —-END CERTIFICATE REQUEST—- and paste it to the appropriate form of the CA vendor or upload it as a file (if such an option is available from the vendor) - the CA will now provide you with a signed certificate.

Now that you have a certificate from the CA you will make use of the Intel's setup and configuration service (SCS) setup wizard that is provided (for free) on Intel's website[4]. The SCS setup wizard provides automated installation of the required services and dependencies needed to utilize the zero touch remote configuration process. Given the deluxe SSL certificate you can start remote provisioning of AMT systems without any physical interaction. Note that there is a stripped down version of the SCS named Manageability developer tool kit to support zero touch remote provisioning of up to 20 (twenty) AMT systems. This lightweight version uses less hardware resources and can be downloaded from Intel's website for free[46]. The overall cost for the single deluxe SSL certificate from GoDaddy cost 60.23 € before tax.

Hello packet

When in setup mode the AMT device transmits Hello packets. a Hello packet is a special network packet transmitted periodically to the provision server and follows the back-off algorithm shown in Table 3.4. The result is that for the first minute the host will send 5 Hello packets during this minute, then it will send 5 more Hello packets over the next 10 minutes, and 5 more packets over the next hour.

The network interface of AMT continues transmitting hello packets by default for 24 hours; except for the AMT versions 2.2, 2.6, and 3.0 that cease transmitting Hello packets after 6 hours. Note that custom OEM versions of AMT can prolong the Hello packet network transmission, up to 255 hours. The back-off algorithm is restarted after a firmware reset of the AMT, by restoring AMT to factory default settings, or by power cycling[26] the AMT enabled system. The Hello packet has the format depicted below.

Table 3.4. Hello packet back-off algorithm

Time intervals	Transmission retries
1 minute	5
10 minutes	5
60 minutes	5

2 bytes header:

- Byte 0 is the hash algorithm, 0 for MD5 and 1 for SHA1
- Byte 1 has a hash length of 16 (MD5 hash) or 20 (SHA1 hash) bytes

16 or 20 bytes Hash corresponds to a root certificate from a CA.

The certificate hashes start at byte offset 25 and each hash entry consists of a header and the hash itself.

3.7.5 Intel AMT remote provisioning: attack scenario

In this attack scenario an attacker performs an unprivileged ZTC remote provisioning to an ATM platform. The attack scenario is based upon the remote provisioning flow of the Intel AMT as discussed in section3.7.1. Our implemented attack is divided in six steps:

1. An unsuspecting user connects to a network (typically a LAN) with his/her Intel AMT PC.
2. Automatically AMT acquires the appropriate DHCP and DNS configuration parameters.
3. Once connected AMT sends a hello packet to determine the availability of the provisioning server.
4. Attacker initiates a (rogue) provisioning server.
5. AMT platform queries the provisioning server and initiates the remote provisioning process.
6. Finally upon completion of the remote provisioning process the attacker has full control of the AMT platform.

In order to orchestrate this attack scenario an attacker can find the available tools for free (see section 3.7.3). An additional requirement is a specific type of a SSL certificate that is applicable to all Intel AMT platforms (see section 3.7.3). This attack could be used indefinite times to compromise the Intel's AMT remote

provisioning process and subverts the security of the non configured PCs that include the AMT functionality even while it is ***disabled*** within the BIOS configuration as presented in section 3.7.6.

3.7.6 Vulnerability: ZTC implemented when AMT is disabled

In our laboratory environment (see section 3) we have tested and found that the ZTC remote provisioning can be implemented even while the Intel AMT functionality is disabled within the BIOS as illustrated in Figure 3.6. Surprisingly the AMT platform broadcasts an ARP request packet upon connecting to a wired network (typically a LAN) and follows the sequence described in section 3.7.1. From this point and beyond the attacker operates the SCS and could manipulate the PC according to his/her malicious activities (see section 3.7.5 even while the *Intel AMT is* ***disabled*** *in BIOS*.

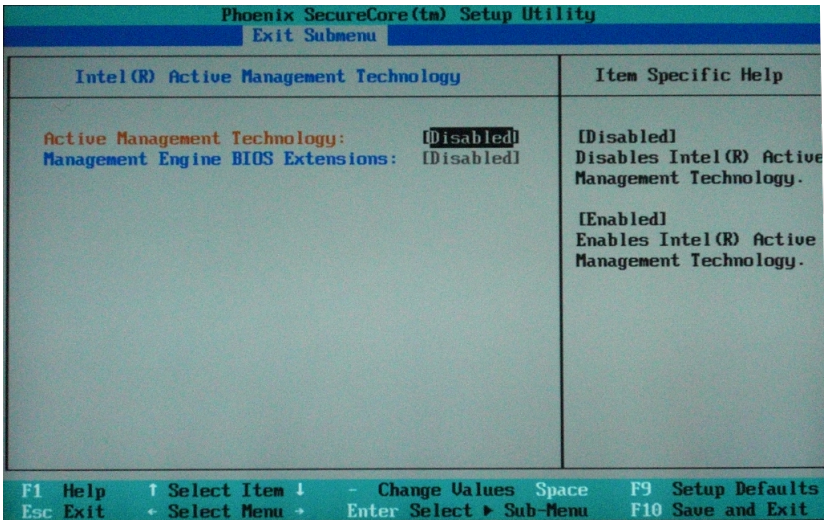

Figure 3.6. Intel AMT disabled in BIOS

Figure 3.7 illustrates the SCS application console screenshot that discovers the not configured AMT platform. Figure 3.8 illustrates the SCS console screenshot

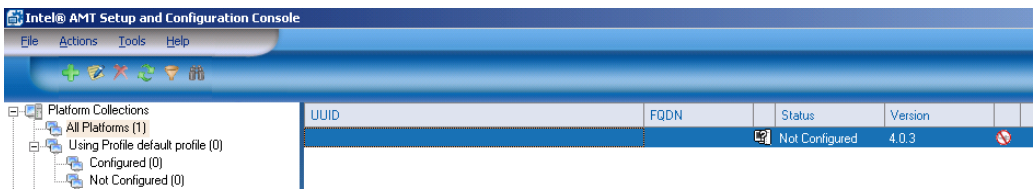

Figure 3.7. Intel AMT SCS console – not configured

pushing our desired configuration profile with success. Obviously the ZTC

provisioning process has been realized while Intel AMT is disabled in BIOS. Surprising after enabling the AMT functionality in BIOS (since it was disabled)

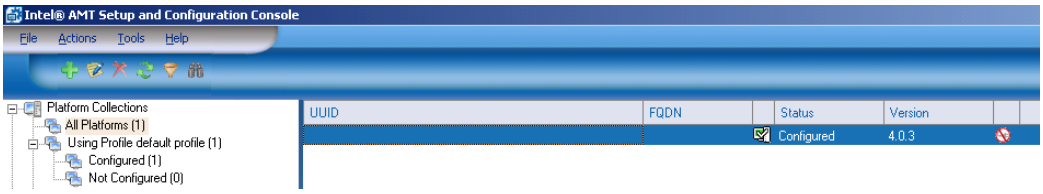

Figure 3.8. Intel AMT SCS console – configured

the configuration remains the same as the one configured with ZTC provisioning while AMT was disabled. That gives a clear advantage to a malicious person implementing his desired attack vector by pushing the configuration to the PC when AMT is disabled. Finally enables the AMT option in BIOS and the configuration remains the same (as the one forced when disabled). Keep in mind that **all** running AMT operations (i.e. remote management operations) are *transparent* to the user due to local environment restrictions implied by the AMT technology. Note that the requirements mentioned in section 3.7.3 are needed (available for free) for this attack vector to be successful.

The PCs along with the released version are illustrated in Table 3.5 have been tested and perform the vulnerability (i.e. ZTC implemented when AMT is disabled) mentioned in this section. Most important, that this critical vulnerability affects even the **latest Intel AMT ME firmware version** [55] 4.2.20.1036 released on the 4th of March, 2010.

Table 3.5. Vulnerability affected PCs

Brand Model	Chipset	Version
Fujitsu Lifebook T5010	GM 45	4.0.3
Lenovo Thinkpad X200	GS 45	4.1.3
Lenovo Thinkpad X200	GS 45	**4.2.20**

3.8 Mobile version of AMT

An important feature of Intel AMT is the mobile version that offers support for most of the remote management control capabilities offered by the wired (default) version of AMT. As we have mentioned earlier the important issue for AMT in general is that it relies on a dynamic IP addressing scheme, thus creating more attack vectors since most of the unsophisticated or "default" type attacks will apply for AMT. The best practice for wired and wireless networking in the case of AMT mobile version would be to use a static IP addressing scheme with a reduced subnet

size an option which cannot be implemented since AMT enterprise provision model (the most secure) operates only in dynamic IP addressing scheme (i.e DHCP). The important note on this issue is that the attacker can deploy the attacks described below with limited resources; typically a notebook with 2 wireless interfaces (for better diversity) is sufficient.

3.8.1 How mobile AMT works

The wireless manageability version of Intel AMT is implemented by using two layers: the wireless network interface controller (NIC) and the interface driver executed on the host platform. The wireless NIC is responsible for managing the radio frequency (RF) communication network connection. In order to utilize the mobile version of AMT the platform must be attached to AC power since AMT activity is limited when the system is operating on batteries[43]. However, when in S0 power state the AMT platform can be configured to operate even when operating on battery mode. In AMT release version 4 and above there is extensive support for remote management in all permitted power states (see section 2.1.1).

3.8.2 Activating AMT mobile version

By default mobile AMT version is disabled. First the mobile version of AMT needs to be enable by specifying an applicable power policy (subject to AMT version) as illustrated in Figure 3.10. Next a wireless profile needs to be configured with an appropriate SSID network name, the credential data, and the appropriate security settings that correspond to the wireless access point (AP) as depicted in Figure 3.10. The mobile version of AMT will receive an IP address from a DHCP server. Note that a static IP addressing scheme is not provided in the mobile version of AMT.

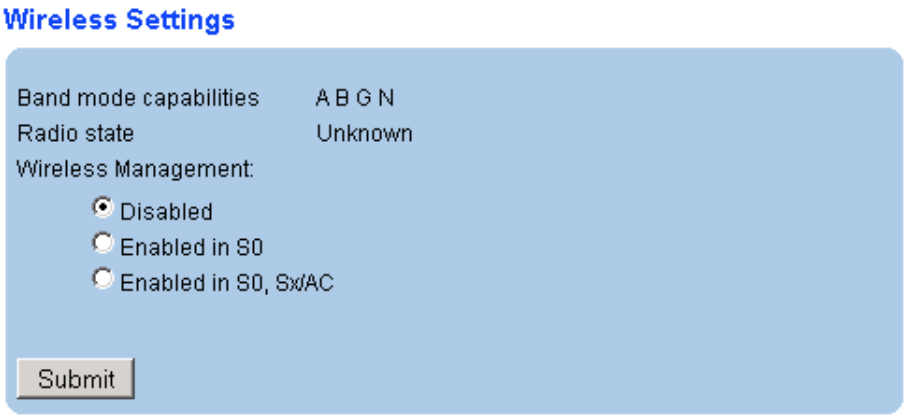

Figure 3.9. AMT mobile power policies

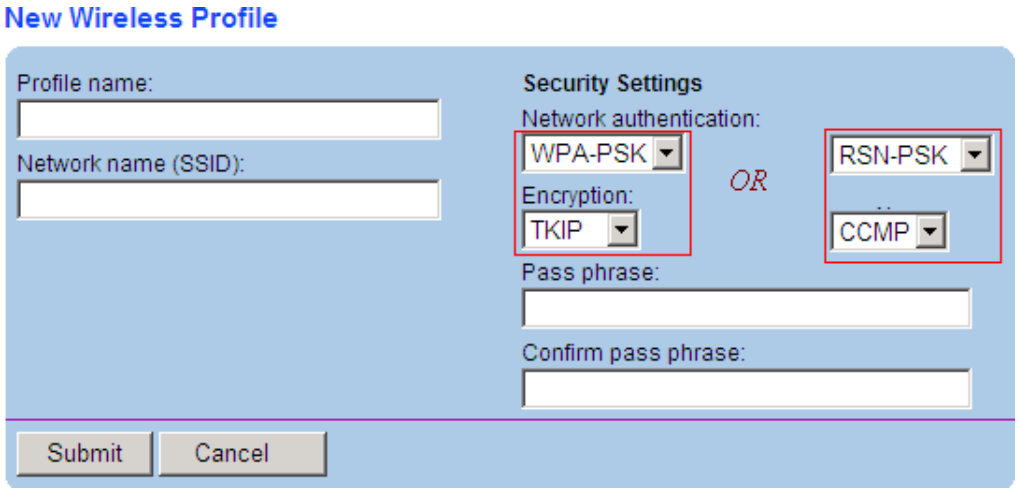

Figure 3.10. AMT mobile version web UI

Additionally, when the wireless manageability features of AMT are active you cannot access any WLAN except those configured in the wireless profile parameters, regardless of the connection's establishment. This means that the wireless LAN adapter can only be used for the AMT mobile provisioning and not for any other WLAN connectivity since the current implementation does not allow simultaneous access to the wireless LAN adapter.

3.8.3 Implementation fault

There is no built in wireless security support, hence the only configuration settings are related to the connectivity settings with the AP as illustrated above in Figure 3.10. The wireless profile parameters for the mobile version of AMT include the following configuration settings:

- Profile name
- Profile priority
- Network Name (SSID)
- Security Settings
  - Key Management approach: Wi-Fi Protected Access (WPA) or Robust secure Network (RSN)
  - Encryption Algorithm: Temporal Key Integrity Protocol (TKIP) or Counter Mode CBC MAC Protocol (CCMP)
- Authentication: Passphrase or IEEE 802.1x profile

The AMT wireless protocol stack broadcasts the following packets at an interval rate of 120 seconds.

- Three IEEE 802.11 Probe requests:
  - One with a packet size of 106 bytes and
  - Two with a variable packet size of 106 bytes with the SSID character bytes added.
- In every probe request the broadcast includes the following:
  - The supported bit rates of the AMT wireless NIC.
  - The supported modulation and coding scheme set
  - The supported high throughput capabilities
  - The SSID parameter set.
- In two out of the three packets, the configured SSID set in the wireless configuration parameters of AMT is broadcasted.

The implementation fault of the mobile AMT version introduces several serious weaknesses (additional details are presented in section 3.8.4).

Automatic network SSID selection imposes serious risks since the mobile platform of AMT tries to associate with the configured SSID (wireless profile) every two minutes. As demonstrated in section 3.8.3 the AMT broadcasts the SSID (along with other data) of the AP, as configured in the wireless profile settings. Therefore an attacker simply listens for the probe requests that are broadcasted, noting the configure SSID, by passively monitoring the wireless channel. Next the attacker impersonate an AP with this SSID, hence it can now deceive the AMT platform into establishing a connection to it (the rogue AP). A more secure practice would limit the AMT to establishing a connection with a specified AP based on its MAC address. This specific MAC address would need to be manually configured. However such a configuration is not available on Intel AMT mobile version. Detailed information is covered in section 3.8.6.

3.8.4 Wireless attacks on AMT

Surprisingly the AMT mobile version is vulnerable to many known vulnerabilities and wireless network attacks.

3.8.5 Attack types

By definition IEEE 802.11 LANs are shared medium networks[1], hence wireless NICs and APs establish connections by utilizing protocols over a shared radio channel. All packets can be monitored by any WLAN interface that is compatible with the transmitter. We classify these types of attack vectors into: confidentiality, integrity and availability attacks.

1. Confidentiality attacks:
    - Fake AP: masquerading as a legitimate AP by using the SSID learned from the broadcast probe SSID frames
    - Man in the Middle: intercepting all the network traffic by using a fake AP.
2. Integrity attacks:
    - 802.11 frame injection: Sending arbitrary 802.11 frames.
    - 802.11 data deletion: Jamming to prevent delivery of frame, while simultaneously spoofing ACKs for deleted data frames.
    - 802.11 data replay: Capturing and replaying modified data.
3. Availability attacks
    - RF jamming: Transmitting at the same frequency as the targeted WLAN is using, perhaps at a power that exceeds the regulated equivalent isotopic radiated power (EIRP) – thus ensuring that no legitimate stations can communicate.
    - 802.11 beacon flood: Generating large amounts of 802.11 beacons making it difficult for the AMT stations to associate with the corresponding AP.
    - 802.11 de-authentication: Probing AMT stations with custom de-authentication packets causes rapid disassociation from a legitimate AP.

3.8.6 Confidentiality attacks

In this section we analyze the confidentiality attacks that apply to the AMT mobile version.

Normally a fake AP implementation is utilized for one reason; to lure unsuspected clients to connect with a malicious AP rather than connecting with a legitimate AP (the one that the user intended to use). In our case the client that establishes a connection with the AP is the AMT system since no defense mechanisms are implemented on the mobile version of AMT. Implementation of a fake AP can be done with open source tools such as Airbase-ng[72]. An overview of this attack combined with the de-authentication attack is illustrated in Figure 3.11.

In principle a man in the middle attack (MITM) occurs when an attacker intercepts the communication between two or more devices, and could eavesdrop, craft, and send arbitrary packets at will. In detail a station "C" (client) is associated and authenticated with a legitimate AP "B". An attacker "A" performs a de-authentication attack (described above) to the station "B" in order to lure the client "B" into establishing a connection to the attacker by impersonating the legitimate AP. While the client "C" successfully associates with the attacker's AP or (a fake

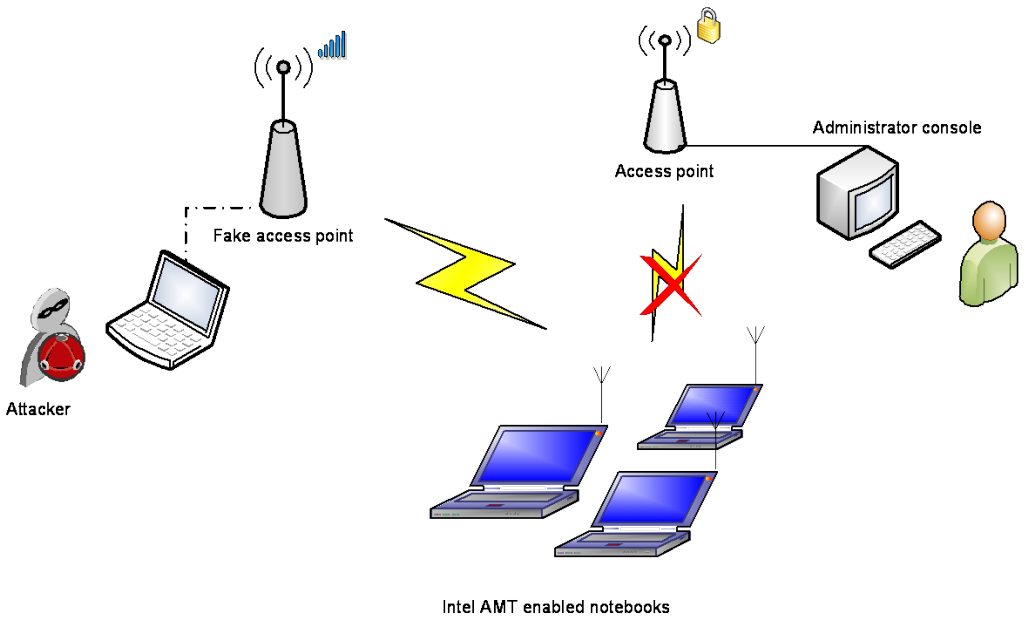

Figure 3.11. Fake AP attack

AP) as described above the attacker "A" will re-transmit all the frames received from "B" to the legitimate AP or frames to the client "C" – hence enabling "A" to eavesdrop and/or modifying at will any of the frames. Note that a MITM attack can be realized without the need for de-authentication or a fake AP attack. The attacker can simply acts as an illicit AP, by using a PC (usually a notebook) with two wireless NICs and perhaps with directional antennas to establish a good connection with the legitimate AP ("B") and with the client ("C").

3.8.7 Integrity attacks

IEEE 802.11 is shared accessed medium and malicious persons can attack the integrity of transmitted data by using frame injection and frame replay tool. By implementing IEEE 802.11 frame manipulation an attacker can transmit data to be the station (client) by transmitting arbitrary IEEE 802.11 packets.

The attacker will generally begin by capturing interesting IEEE 802.11 frames that will fit his attacking schema, then manipulating and sending forged 802.11 frames to the legitimate AP or targeted station. Note that the frames can be manipulated off-line or in real-time. Open source tools such as airpwn[15] provides a framework for IEEE 802.11 packet injection. Airpwn listens for incoming wireless data frames and uses pattern matching to select "interesting" frames. If the data matches a pattern specified in the configuration files, a customized response is injected or spoofed accordingly to/from the wireless access point. As long as airpwn responds prior to the legitimate AP, the station (client) will accept these frames as

valid. From the client side airpwn seems to be a "legitimate" AP.

3.8.8 Availability attacks

Availability attacks are closely related to denial of service (DoS), thus making unavailable a service or a resource hosted on a system usually by resource exhaustion of the specific services. These attacks are implemented by transmitting carefully crafted control and management packets using IEEE 802.11 frames. DoS attacks are extremely difficult to control and prevent, since the attack is not easily detectable, when implemented by sophisticated attackers. Such an attack could last from milliseconds to hours.

Because the IEEE 802.11 network is a shared medium, RF jamming provides a very effective attack. RF jamming denies service to the legitimate users, in our case the Intel AMT administrator and client. An attacker can disrupt operation of the IEEE 802.11 network communications by increasing interference, leading to the signal to noise ratio of legitimate traffic being too low for successful detection by the legitimate receivers. An attacker can launch such attack by using modified WLAN NIC (for example with an amplifier between the NIC and the antenna or using a highly direction antenna to focus the NIC's emissions in a given direction). The attacker could also use a powerful signal generator to flood the relevant IEEE 802.11 radio channels.

A beacon flood attack overloads the communication channel with counterfeit packets, so that the AP spends all of its resources processing the flood packets rendering it unavailable to legitimate users. This type of attack can performed with open source tools such as Fake AP[70] to generate the appearance of a very large number of rogue AP, thus "confusing" the legitimate client when it tries to select a legitimate AP. Such an attack is illustrated in Figure 3.12.

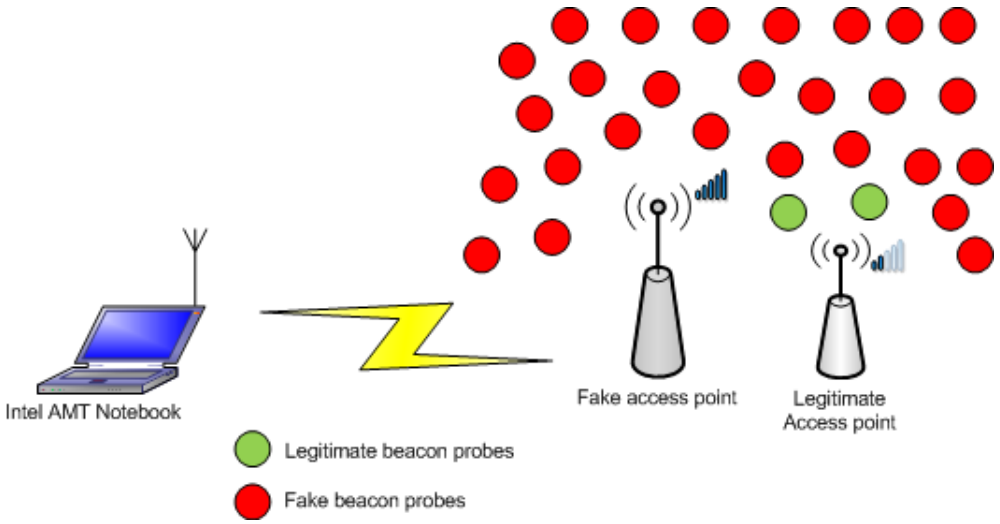

Figure 3.12. Beacon flood attack

An IEEE 802.11 station (client) associates with and authentication with a legitimate AP by transmitting IEEE 802.11 frame. An attacker can interfere with this process by sending spoofed network frames impersonating the AP or the station. Such an attack can target a specific or denying access to the whole communication channel. This attack is quite powerful since the attacker can monitor the WLAN channel and transmit de-authentication frames to newly associated stations. Such an attack can be launched with freely available open source tools, for instance aireplay-ng[73]. An example command line is shown in listing 3.14. In this example, you simply fill in the client's MAC address and the ESSID that you wish to target.

```
aireplay-ng --deauth 15 -c 'client's mac' -a 'essid mac'
   ath1
```

Listing 3.14. De-authentication attack example

3.9 Intel AMT Privacy threat

Intel claims that their AMT system has been designed to securely manage systems, while allowing administrators and end-user to opt-in or out-out:

> Intel AMT has several built in mechanisms to ensure that computer management happens securely and only by authorized IT administrators belonging to the organization's IT departments. It also provides several options that offer choices and control to IT administrators and end users of the computers for opting in or opting out of various capabilities.
> ...Mechanisms are also available to protect against attacks by rouge administrators.

[[13]; page 288-289] However, it is not clear that this statement is actually true in fact. We will examine some of the specific issues in the following subsections.

3.9.1 Privacy protection mechanisms in AMT

The user can opt-out from AMT whenever he wants according to the above quotation, but how does the user do so? Does the user need to know a password? We noted earlier that they are two passwords: one for MEBx and one for the WEB/XML server; but these are not synchronized. So which of these passwords does the user need and how can he/she use it?

If the AMT mechanism can be used even if the user has disabled it in the BIOS - how can the user know that no one else can activate it and re-provision their computer?

3.9.2 End user notification

As we have previously discussed AMT acts regardless of the OS or without OS at all, so why does AMT platform depend on the OS in order to display an end-user awareness message status of AMT since Intel cares about the end-user's privacy while being monitored or managed?

> [...] the end-user has little knowledge whether his or her computer is being managed and monitored by another entity–the IT administrator. From a privacy standpoint, the impact of such a mode of operation might range from an uncomfortable feeling for the end-user to something more significant. Therefore. Intel developed a host software-based modification mechanism (an application icon in the system tray) that explicitly made the end-user aware that Intel AMT is available within the platform, and some other information regarding its status. The main functionality of the tray icon application is to give Intel AMT status to the end-user of the computer with Intel vPro technology. This icon application is available for the most popular versions of Windows and Linux (Windows XP, Windows Vista, Red Hat Linux and Novell SuSE Linux).[[13]; page 293-294]

Surprisingly the AMT status notification in our tested system displayed that the AMT platform while we were remotely managing the device as illustrated in Figure 3.13. We have used version 3.1.0.31 of the MEstatus package; a GUI application used to monitor Intel AMT status by providing a system tray icon on the desktop. It depends on the uns package with version 3.2.0.24; an application that receives messages from the AMT platform and transmits them to the local OS event log, in order to notify the end-user.

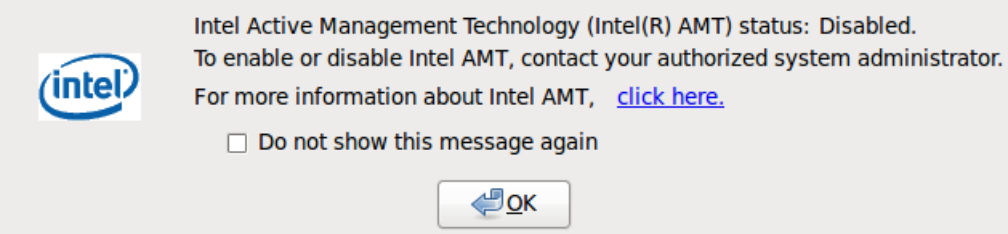

Figure 3.13. AMT status notification

3.9.3 Privacy concerns from publishers and end-users

Rick C. Hodgin posted an article on TG Daily[66] about Intel AMT technology part of it quoted:

> Big Brother potentially exists right now in our PCs, compliments of Intel's vPro. The ability for a CPU, chipset and network chip to operate independently of the OS through commands given to it from hidden, out-of-band communications is a telltale sign that it is possible. And while there may be many applications which benefit from such technology (Intel indicates billions of dollars saved, including hundreds of thousands of tons of greenhouse gas emissions, through the use of vPro's ability to operate even if the machine is off), the enabling factors are there for vPro to be used by another type of system; something like Big Brother.

From GIGA-BYTE technology corporation community forums[28] as quoted:

> How to disable Intel vPro spyware
> Please enable your BIOS with an option to permanently and effectively disable all Intel vPro technology. The last thing I want is hackers or the US government spying on my computer and downloading my files, passwords, etc. without authorization.

Tony Dennis by Incisive Media published this article[74] part of this are quotationd:

> Intel proudly shows off snooping tech ...Details about a vPro or Centrino based PC are saved into non-volatile memory. But, scarily, this information can be read even if the machine's power switch is in the 'off' position. ...Obviously Intel claims this kind of stuff is mega secure. But what if it were hacked? Or what if they hacked it? You could potentially be woken up in the middle of the night by the sounds of somebody completely reconfiguring your laptop.

Chapter 4

Conclusions and Future Work

4.1 Conclusions

A vicious employee, organization, or even an OEM can "accidentally" force a rigorous configuration customized to his needs in order to completely control the AMT platform, thus gaining (and maintaining) complete control of the computer system. As a result it is possible to install a powerful rootkit (i.e AMT) over the Internet as Intel presented in[6]. They showed that it is possible to control the IDE-R functionality of AMT for remote diagnosing and repairing remote sites over the Internet or a mesh networking by using a distributed network architecture. This enables the installation and control of a botnet now on the hardware level. AMT was introduced in 2004[31] and many IT administrators as well as OEMs do not recommend to upgrade firmware (see appendix A). Some vendors luckily include an upgrade of the AMT firmware bundled within the BIOS firmware upgrade, while surprisingly some others are not aware of the AMT functionality and the potential capabilities of this remotely controllable powerful rootkit, given the fact the tremendous insecurity that the ZTC remote provisioning process provides.

Customers and end users lack knowledge of this backdoor (i.e. AMT) when they buy a PC that includes the AMT platform. This can potentially lead to havoc should there be a well publicized attack that would make them concerned about their privacy and ability to control their own computers. Since AMT is enabled by default in many computers and the end user often lacks technical expertise this leads to unexpected vulnerabilities for the user. Unfortunately today AMT is shipped on "home" PCs and as we previously discussed the insecurities of this embedded device are scary, especially as IT administrators, end customers of PCs, notebooks, and servers are unfamiliar with this technology, thus malicious persons could remotely manipulate these systems. Over the last decade IT technology has advanced tremendously, but many of the systems that have been developed are insecure.

4.2 HTTP digest access authentication issues

Apparently Intel made a poor choice implementing the HTTP digest access authentication scheme[48] to the AMT platform for crucial operations, specifically for remote management. The implemented access authentication method dates back to 1999 and could be exploited by a malicious person carrying out an off-line brute force attack. Depending on the attacker's resources, ingenuity, the use of specialized hardware and software contributes to the recovery –cracking of the administrative credentials, and then managing the AMT platform accordingly(see section 3.6). Intel does not follow the recommendations of the Internet engineering task force (IETF) who noted in June 1999 in section 4.14 of[48] that with respect to the cryptographic standards of 1999 that Digest Authentication is weak. We should note that by the standard of 2010, the security offered by Digest Authentication is largely unacceptable for most practical applications, hence AMT should be revised to utilize a TLS implementation in all setup and configuration and note only in the enterprise provision model (see Table 2.6).

4.3 Mobile version issues

Wired networks (typically Ethernet) have less concern for security compared with wireless networks. In principle wireless connectivity does not provide weaker security that wired networks –unless the physical wires of the network are protected. However, use of a wireless connection facilitates easy connectivity for both good and bad purposes. Catastrophic consequences can occur to the managed PC's hardware when "sensitive" AMT functionalities are used. For instance a remote BIOS upgrade of a platform via the wireless network link. Hence the insecurity of Intel's AMT OOB remote management can be easily exploited as presented in section 3.8.4.

4.4 Gratis hardware rootkit

The Intel AMT platform can provide the basis for secretly operating a platform's hardware based on a backdoor, since its included in a prominent fraction of PCs (desktop and notebooks) servers, POS, embedded systems and even ATMs as illustrated in Table 2.2 and 2.3. By design the AMT architecture uses a separate OOB channel independent of any network traffic transmitted to and from the other parts of the computer. Therefore this interface provides a covert communication channel that introduce a set of issues: allowing malicious parties to perform surveillance, to monitor, to carry out espionage, and fully control a system. Malicious parties include government agencies, individual attackers, and industries. A catalytic factor for this powerful backdoor arises from the functionality of AMT architecture that was designed to operate and perform remote management, even though the system is turned off.

Additionally though a separate communication channel via the TCP/IP protocol stack, the platform can be controlled remotely over a wired or a wireless interface, even if the operating system implements a firewall or other security countermeasures. In this thesis we have demonstrated (see section 3.7 and 3.2) different ways of fully controlling an Intel AMT enabled system via the ZTC functionality and bypassing local access restrictions to the platform.

4.5 Recommendations

The AMT technology aimed to introduce a powerful remote management tool, creating a secure infrastructure that is secure against inbound and outbound attack vectors, especially for critical environments such as enterprises. Unfortunately, this means that the underlying technology requires careful design and implementation of security, especially with respect to network centric attacks - in order to successfully prevent critical attack vectors from being exploited. Our suggested recommendations for the AMT platform would be to enforce the TLS/SSL security implementation in all provisioning models and not only in one out of the three setup and configuration models.

Additionally extended security features should be implemented in the mobile version of to protect the AMT since it is highly dependent upon the access point's security scheme. As discussed in section 3.8.4 the mobile version of AMT falls into a variety of attacks and could be implemented by an attacker with limited resources. Last but not least the ZTC remote configuration process should be implemented in such an extend that malicious entities and attackers would be almost impossible to obtain this special type of certificate for qualifying and successfully employing the ZTC remote provisioning as discussed in section 3.7.2.

4.6 Future work

In this section we will enumerate some of the suggested future work that could extend and enhance our research done in this thesis given the fact that there is almost not related research in the field. The logical continuation to this thesis is a security evaluation of the SSL/TLS implementation (i.e. enterprise provisioning model) of the AMT platform in respect to integrity and confidentiality attack vectors. Additionally an interesting and approach would be the security analysis with respect to the hardware layer of the AMT platform. Finally surveying the AMT's certificate based protection would be compelling to follow up on this research.

Appendices

Appendix A

Vendor e-mail communication

- Quoting the e-mail communication from Acer Group Customer Service [7]:

  We would not recommend flashing your BIOS due to the inherent risk of rendering your machine inoperable. We are able to repair your machine if the BIOS has been flashed. Unfortunately this problem is not covered by the warranty and would be classed as a chargeable repair. an initial charge of £51.99 would need to be taken.
  ...

- Quoting the e-mail communication from Fujitsu Technology Solutions technical support [27]:

  Regarding your request to LIFEBOOK T5010 iAMT ME firmware version. The answer from our development:
  the current iAMT ME Firmware version is: 4.1.3.1038.
  The iAMT firmware version can not be flashed by BIOS upgrade.
  The iAMT firmware version will be flashed during production process.
  ...

Bibliography

[1] IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. June 1997.

[2] Intel Centrino Pro and Intel vPro Processor Technology. 2007. URL http://download.intel.com/pressroom/kits/centrino/CentrinoPro_vPro_whitepaper.pdf.

[3] BarsWF MD5 Cracker. October 2008. URL http://3.14.by/en/md5.

[4] Intel SCS Setup Wizard, September 2008. URL http://www.vproexpert.com/59JHE/.

[5] OpenSSL Project, June 2009. URL http://www.openssl.org.

[6] Abdul Bailey, Teri McFaul and Ylian Saint-Hilaire. Small & Medium Business: How to Deliver Intel vProTM Technology, September 2009. URL http://www.intel.com/go/idfsessions.

[7] Acer Group Customer Service. Personal e-mail communication. December 2009.

[8] Alex Golod. Security FAQ for Intel vPro Technology. 2008. URL http://communities.intel.com/docs/DOC-1989.

[9] Alexander Peslyak. John the Ripper. URL http://www.openwall.com/john/.

[10] Alexander Tereshkin and Rafal Wojtczuk. Introducing Ring -3 Rootkits. 2009. URL http://invisiblethingslab.com/resources/bh09usa/Ring%20-3%20Rootkits.pdf.

[11] Alexander Tereshkin and Rafal Wojtczuk. Introducing Ring -3 Rootkits proof of concept code, July 2009. URL http://invisiblethingslab.com/resources/bh09usa/ring-minus-3-tools-1.3.tgz.

[12] ARC International. ARC4 processor. April 2003. URL http://web.archive.org/web/20040926134313/http://www.arc.com/upload/download/F1010.5_ARC+A4_4-9-03_FINAL.pdf.

[13] Arvind Kumar, Purushottam Goel and Ylian Saint-Hilaire. *Active Platfrom Management Demystified*. Intel press, June 2009.

[14] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. pages 390–420.

[15] Bryan Burns. Airpwn. June 2006. URL `http://airpwn.sourceforge.net/Airpwn.html`.

[16] Business Wire. Atos Origin Takes Pole Position with Intel AMT from Intel, June 2005. URL `http://findarticles.com/p/articles/mi_m0EIN/is_2005_June_6/ai_n13797230/`.

[17] CCITT (Consultative Committee on International Telegraphy and Telephony). Recommendation X.509: The Directory Authentication Framework, 1988.

[18] Center for Democracy and Technology Group. Anti-Spyware Coalition Group. URL `http://www.antispywarecoalition.org/documents/glossary.htm`.

[19] Damon Poeter. Intel vPro Chipset Lures MSPs, System Builders. August 2007. URL `http://www.crn.com/white-box/201802550`.

[20] Distributed Management Task Force Inc. Alert Standard Format Specification.

[21] Distributed Management Task Force Inc. Common Information Model. URL `http://www.dmtf.org/standards/cim/`.

[22] Distributed Management Task Force Inc. Desktop and mobile Architecture for System Hardware Initiative. URL `http://www.dmtf.org/standards/mgmt/dash/`.

[23] Distributed Management Task Force, Inc. *Web Services for Management*, December 2008. URL `http://www.dmtf.org/standards/wsman/`.

[24] Distributed.net. Distributed computing project, 1997. URL `http://distributed.net/`.

[25] Erik Zachte. Wikimedia Visitor Log Analysis Report - Operating Systems, December 2009. URL `http://stats.wikimedia.org/wikimedia/squids/SquidReportOperatingSystems.htm`.

[26] Free Online Dictionary of Computing. power cycle term. URL `http://foldoc.org/power+cycle`.

[27] Fujitsu Technology Solutions Technical Support. Personal e-mail communication. December 2009.

[28] GIGA-BYTE technology community forums. How to disable Intel vPro spyware, October 2009. URL `http://forum.giga-byte.co.uk/index.php/topic,724.msg3233.html`.

[29] Gregory Regan. University of Plymouth & PC Power Management. September 2008. URL http://www.heepi.org.uk/jisc/events/Sheffield%20Presentations/Gregory%20Regan%202.9.08.ppt.

[30] Hewlett-Packard Corporation, Intel Corporation, Microsoft Corporation, Phoenix Technologies Ltd. and Toshiba Corporation. *Advanced Configuration and Power Interface Specification*, June 2009. URL http://www.acpi.info/DOWNLOADS/ACPIspec40.pdf.

[31] IDG Communications. CA announces plans to support Intel Active Management Technology, October 2004. URL http://www.arnnet.com.au/mediareleases/3706/ca-announces-plans-to-support-intel-active-managem/.

[32] Intel. *Active Management Technology System Defense and Agent Presence Overview*. URL http://cache-www.intel.com/cd/00/00/32/09/320960_320960.pdf.

[33] Intel. Intel Active Management Technology Technology Brief, February 2005. URL http://cache-www.intel.com/cd/00/00/23/63/236366_236366.pdf.

[34] Intel. Improve IT efficiency white paper. 2006. URL http://www.cgs4u.com/docs/intel%20active%20management%20technology.pdf.

[35] Intel. Active Management Technology Small Business Configuration User Guide. May 2007. URL http://www.intel.com/technology/manage/downloads/amt_smbusiness.pdf.

[36] Intel. Case study: Nottingham University Hospitals NHS Trust puts Intel vPro technology. July 2007. URL http://communities.intel.com/docs/DOC-1131.

[37] Intel. Architecture Guide: Active Management Technology. 2008. URL http://software.intel.com/en-us/articles/architecture-guide-intel-active-management-technology/.

[38] Intel. Intel AMT Remote Configuration Certificate. 2008.

[39] Intel. Intel Centrino 2 with vPro Technology and Intel Core2 Processor with vPro Technology. 2008. URL ftp://download.intel.com/products/vpro/whitepaper/crossclient.pdf.

[40] Intel. Intel Technology Journal. 12, 2008.

[41] Intel. *Intelligent Platform Management Interface Adopters List*, August 2008. URL http://www.intel.com/design/servers/ipmi/adopterlist.htm.

[42] Intel. *Intelligent Platform Management Interface reference*, August 2008. URL http://www.intel.com/design/servers/ipmi/spec.htm.

[43] Intel. Mobile Computing with Intel AMT. July 2008. URL http://software.intel.com/en-us/articles/mobile-computing-with-intel-amt/.

[44] Intel. Technical Considerations for Intel AMT in a Wireless Environment. 2008. URL http://software.intel.com/en-us/articles/technical-considerations-for-intel-amt-in-a-wireless-environment/.

[45] Intel. 2009 Intel IT Performance Report. January 2009. URL http://communities.intel.com/docs/DOC-4776.

[46] Intel. Manageability Developer Tool Kit, May 2009. URL http://software.intel.com/en-us/articles/download-the-latest-version-of-manageability-developer-tool-kit/.

[47] Internet Assigned Numbers Authority. Port numbers. URL http://www.iana.org/assignments/port-numbers.

[48] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen and L. Stewart. RFC 2617, HTTP Authentication: Basic and Digest Access Authentication. June 1999. URL http://www.ietf.org/rfc/rfc2617.txt.

[49] John Linn. Privacy Enhancement for Internet Electronic Mail — Part I: Message Encryption and Authentication Procedures. February 1993.

[50] Josh Hilliker. Known Issues, Best Practices, and Workarounds. November 2007. URL http://communities.intel.com/docs/DOC-1247#Using_international_keyboards_to_create_MEBx_passwords_via_Setup_and_Configuration_Service_SCS.

[51] Josh Hilliker. Tools and Utilities for Intel vPro Technology. October 2007. URL http://communities.intel.com/docs/DOC-1171.

[52] K. Fiftal, E. Herold, R. Klleiman and J. Nicholes. *Intel Active Mananagement Technology Security Configuration Guide*, December 2008.

[53] Kelsey Witherow and Steve Lewis. Order an Intel vPro technology "Activation-Ready" PC. December 2009. URL http://communities.intel.com/docs/DOC-2033.

[54] Ken Munro. Database security an oxymoron? December 2006.

[55] Lenovo. Intel AMT 4.2 Management Engine Firmware. March 2010. URL http://www-307.ibm.com/pc/support/site.wss/MIGR-71137.html.

[56] MD5.My-Addr.com. My-Addr Project MD5 database, October 2008. URL http://md5.my-addr.com/.

[57] Michiel Timmers and Adriaan van der Zee. LIA project: Intel AMT and network management. April 2009. URL https://www.os3.nl/2009-2010/courses/lia/start.

[58] Naren Kumar. *Order an Intel AMT enabled Embedded Product*, August 2009. URL http://communities.intel.com/docs/DOC-3737.

[59] Ned Freed and Nathaniel S. Borenstein. Multipurpose Internet Mail Extensions (MIME) — Part One: Format of Internet Message Bodies. November 1996.

[60] NVIDIA Corporation. CUDA Architecture Overview, April 2009. URL http://developer.download.nvidia.com/compute/cuda/docs/CUDA_Architecture_Overview.pdf.

[61] Openwall Community Wiki. How to extract tarballs and apply patches. URL http://openwall.info/wiki/john/how-to-extract-tarballs-and-apply-patches.

[62] Project RainbowCrack. The Time-Memory Tradeoff Hash Cracker. URL http://project-rainbowcrack.com.

[63] R. Fielding and J. Gettys and J. Mogul and H. Frystyk and L. Masinter and P. Leach and T. Berners-Lee. RFC 2616, Hypertext Transfer Protocol – HTTP/1.1. June 1999.

[64] R. Housley, W. Ford, W. Polk and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. 1999.

[65] R. Shirey. Security Architecture for Internet Protocols: A Guide for Protocol Designs and Standards Internet Draft. November 1994.

[66] Rick C. Hodgin. Big Brother potentially exists right now in our PCs, September 2008. URL http://bit.ly/DcG3A.

[67] Romain Raboin. HTTP Digest Access Authentication patch. URL http://syscall.eu/romain/.

[68] Sreelekshmy Syamalakumari. Intel MEBx and QWERTY Keyboard. May 2008. URL http://software.intel.com/en-us/blogs/2008/05/16/intel-mebx-and-qwerty-keyboard/.

[69] Stat Owl. Operating System Version Usage, December 2009. URL http://www.statowl.com/.

[70] Stuart Stock and Ken Beames. Fake AP, March 2005. URL http://www.blackalchemy.to/project/fakeap/.

[71] The Jargon File. Out-of-band terminology, December 2003. URL http://www.catb.org/jargon/html/O/out-of-band.html.

[72] Thomas d'Otreppe and Christophe Devine. Airbase-ng, September 2009. URL http://aircrack-ng.org/.

[73] Thomas d'Otreppe and Christophe Devine. Aireplay-ng, September 2009. URL http://aircrack-ng.org/.

[74] Tony Dennis. Intel proudly shows off snooping tech, September 2006. URL http://www.theinquirer.net/inquirer/news/1045799/intel-proudly-shows-off-snooping-tech.

[75] Victorian Electronic Democracy. Scrutiny of Acts and Regulations Committee. May 2005. URL http://www.parliament.vic.gov.au/SARC/E-Democracy/Final_Report/Glossary.htm.

[76] Ylian Saint-Hilaire. Top 5 things I would change about Intel AMT. URL http://software.intel.com/en-us/blogs/2008/02/24/top-5-things-i-would-change-about-intel-amt/. February 2008.

[77] ZDNET Asia. Bangkok General Hospital Finds Intel vPro Technology With Intel AMT Capability Heals Asset Management Woes, May 2007. URL http://www.zdnetasia.com/itlibrary/enterprise-applications/0,3800009948,40574930p,00.htm.

TRITA-ICT-EX-2010:37

www.kth.se