LIA project: Intel AMT and network management

Michiel Timmers & Adriaan van der Zee
michiel.timmers@os3.nl - adriaan.vanderzee@os3.nl,
System and Network Engineering,
Universiteit van Amsterdam
Netherlands

April 1, 2009

1 Abstract

This article describes how Intel AMT network filters perform in a network and what there capabilities and limits are. We have tested multiple filters and see what there affect was. This report also describes the various administrative decisions you will have to make if you decide to deploy Intel AMT in your network.

2 Acknowledgments

From the System and network Engineering program of the University of Amsterdam we would like to thank Karst Koymans and Jaap van Ginkel for bringing us into contact with Intel.

Special thanks go to Marc Beckers from Intel Brussels for arranging the delivery of three Intel vPro enabled desktop machines to our lab classroom, on only a weeks notice. These machines have enabled us to conduct real life experiments with the Intel AMT Network Defense system.

Contents

3 Introduction

This report is the final product of a project that has been done for the course called Large Installation Administration (LIA) [3], which is part of the one-year curriculum of the master study System and Network Engineering [4] from the University of Amsterdam (UvA). The self-chosen subject of this project is Intels AMT Network Defense system [1], in a LIA context. Experiments have been carried out on Intel supplied AMT equipped desktop machines.

3.1 Background information about the subject

Intel's Active Management Technology (AMT), currently marketed by Intel as part of the vPro suite [2], allows for out of band management of desktop pc's and notebooks. This could aid administrators of large heterogeneous networks, as the AMT interface can be managed independently from (the state of) the operating system, and even while the machine is powered down.

One of AMT's features is the so called System Defense system [1]. This system comprises of packet filters that can be programmed remotely, and which are claimed to be implemented on the network interface hardware. Therefore it should not be possible to circumvent these filters from within the operating system. This means that the logical view of the network infrastructure can be extended from the access switches to the network interfaces of Intel AMT equipped desktop or notebook machines. As the AMT System Defense network filters can be used to block or rate limit specific types of network traffic, it might be possible to use this to enforce individual network access policies to machines in a situation where it is not feasible, or even impossible to do so more centrally in the network.

3.2 General description of the project

Intel markets its AMT System Defense especially for its heuristic filters, which should detect and prevent malicious network traffic initiated by worms and viruses on the host operating system. The goal of this project, however, is to take a more centralised approach, and find out to what extent the Intel AMT System Defense technology can be used to aid network administration more in enforcing specific kinds of individual or group-based network access policies. Both security and performance of the network filters will be taken into account. Furthermore, these capabilities will be placed in the broader perspective of manageability and scalability across large networks.

3.3 Problem definition and research questions

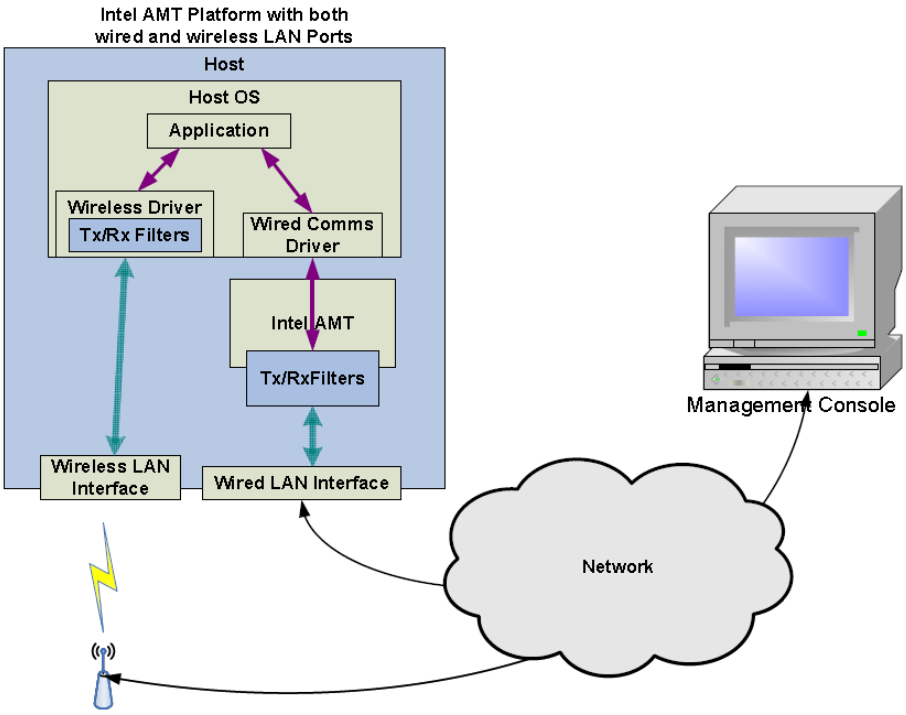
How suitable is Intel's AMT System Defense technology for centralised management of individual desktop network access policies in large networks?

- What are the capabilities and limits of the network filters?
- How secure and operating system independent are the network filters?
- Do the network filters affect the performance of the network interface and the host system?
- Is management of individual or group based network access policies scalable to large networks?

3.4 Outline of this report

In section 4 Intels AMT technology is discussed. Subsection 4.1 compares AMT to ASF, another remote management technology; subsection 4.2 explains the network filter options, which the core of this project is about, and subsection 4.3 lists some of the limitations of AMT. Section 5 discusses how to setup and access a AMT client and talks about the different provisioning modes that Intel AMT can operate in and what the key differences are. The experiments that have been done during this project will be described in section 6. We conclude this report with a conclusion about AMT and it's use in the network.

4 Intel AMT



(Figure 1: Intel AMT System Defense and Agent Presence Overview [2])

Intels AMT technology provides a number of useful features for system and network administrators of large installations. Asset information (system serial number and installed hardware) that is stored and updated on the client system, and accessible even when the system is shut down. With AMT the system can be powered up and down remotely, pre-boot BIOS configurations are accessible, and the console as well as IDE devices can be redirected to the management system, which enables booting a remote machine from local media. However, the feature that this project has focused on is the System Defense system, which enables the remote management of network filters and/or rate limiters that are implemented on the network interface card (NIC) in hardware.

The AMT technology is available for both wireless and wired NICs but as the truly hardware based network filtering is only possible on wired NICs (see figure 1) this project has only focused on the wired AMT solution. The position on the Ethernet controller is ideal to implement network filters, as they will be transparent to the host of the machine. Furthermore, the position on the regular Ethernet interface means that it is accessible for the management console through the already existing network infrastructure. More information about the ways to connect to an AMT system and the possibilities for encryption will be discussed in section 5.

4.1 Compared to ASF

An alternative technology that aims to ease remote management of desktop computers is the Alert Standard Format (ASF)[5], which is an open standard developed by the Distributed Management Taskforce (DMTF). This standard has been last updated in 2002, and describes ways to send and receive messages to and within an ASF enabled device.

Compared to Intels AMT, ASF is less interactive. Although it is possible to remotely power up and down a machine, and the built-in possibility of an ASF system to report (failure) conditions to a management console, it lacks the interactivity of AMT which for example enables IDE redirection for remote boot from local media, initiated by the management console.

One great advantage of ASF is that it is an open standard that describes the exchange of messages, also within an ASF system. This means that vendors of any hardware component can integrate ASF support, and send and receive message to and from the ASF controller. For example, a video card could send critical status messages (i.e. over heating) to the ASF controller, which can propagate them to a management console.

Although Intels AMT is proprietary, and DMTFs ASF is a completely open standard, Intel has worked to make its product easy to adopt by third parties by using SOAP to exchange messages between the client and management console. Internally, however, AMT is still a closed standard, whereas ASF enables any hardware vendor to adopt its standard.

The System Defense system, with its network filters and rate limiters are unique for Intel AMT based systems.

4.2 Network Defense system

The two basic functions the AMT System Defense network filters can perform are filtering predefined network packets, both incoming and outgoing, and limiting the rate at which they are allowed to enter or exit the network interface.

The two building blocks of the Network Defense system are filters and policies. Up to 32 ingress and 32 egress filters, and up to eight policies can be defined. A policy can consist of any combination of filters, and when applied, all the filters of the policy work together.

A policy can be selected manually, or based on heuristics. The latter means that certain thresholds can be set for specific network traffic, which is meant to identify a virus or other threat, which can trigger the AMT system to enforce a predefined policy, for instance to put a system into quarantine. This technique, however, is outside the scope of this project, as the project focuses on centralised network management.

The network filters can be specified for a number of different packet types. Depending on the type of packet extra different finer-grained specifications can be given. There are three levels at which packets can be identified: Ethernet frame type, network layer protocol and transport layer protocol. When Ethernet frame type is chosen as selection criterion, one can only specify its type, such

as IPv4, IPv6, ARP and PPP, and no further. Network layer protocols, such as IPv4 and IPv6 can be further specified as for which transport layer protocol (such as ICMP, TCP, UDP and L2TP) they should apply, and in addition a network address range can be specified. When traffic is identified at the third level, transport layer protocol, TCP or UDP port ranges (either source or destination) can be specified, as well as network layer addresses.

Each of these filters can either be configured to drop the specified traffic, pass it through (in case a general drop filter is combined with specific pass-through filters), or to set a limit on the maximum number of packets that are allowed to pass through in a second. The rate limiter works by starting each second as a pass-through filter, counting the number of matching packets, and when the maximum has been reached, to transition to a blocking state for the remainder of the second. This means that the rate limit is set to a number of packets per second (PPS), and not the more common unit of bits or Bytes per second. With the maximum transmit unit (MTU) of 1500 bytes that is typical for Ethernet 10 PPS translates roughly to 15 kilobytes, or 120 kilobits, per second.

In addition to these specific user configurable filters there is also a built-in anti-spoofing filter that can enforce traffic to only be allowed to pass when it belongs to the IP address that has either been manually configured in the AMT configuration, or obtained with DHCP (which AMT snoops).

4.3 Limitations of AMT

Even though the network filters are highly configurable and fine-grained (see section 4.2), they do not keep track of session states, and are therefore not equivalent to a stateful packet inspection filter or firewall.

Although most of the AMT features, including access to the network filters, is possible even when the system is turned off, the target computer still needs to be connected to both the network, and a power source.

Furthermore, the network filters can be used to prevent malicious or excessive data streams to enter the network, but only on a designated NIC on an AMT capable system. Therefore it does not protect the network itself, because non-AMT machines or even AMT machines with a secondary NIC can access the network freely. Even though AMT has its advantages regarding end-system based security, reducing the need for expensive network equipment at the access layer, network-based access control will still be needed when non-AMT systems can be attached to the network.

5 Set Up and Access Intel AMT

The following section describe what must be taking in consideration before setting up AMT on your clients. There are different approaches to this that will accomplish the same from a client view, but for a network administrator these difference's are very important.

5.1 Small Business Mode / Enterprise Mode

Intel AMT can be configured in two provisioning modes: Small Business (SMB) Mode and Enterprise Mode[6]. The most pointing differences between these two modes are the security features and the flexibility of each. While conducting our experiments we have tested the two modes and found out that the SMB mode was far more easy to setup in comparison with Enterprise mode. The use of SMB mode however should only be done in a test environment where you want easy and fast configuration to be done for a small amount of AMT clients to perform some test or to get familiarized with Intel AMT. In a production environment we would always recommend Enterprise mode over SMB as pointed out below.

Security
While sniffing the network while performing our experiments (See section 6) we found out that Intel AMT settings are configured with the use of Simple Object Access Protocol (SOAP)[7] messages that rely on XML for its format. When using SMB mode these messages are sent in clear text to the Intel AMT client and could easily be sniffed and a replay attack can be performed by retransmitting a modified SOAP package to the client. If you use Enterprise mode all these messages are encrypted using TLS-PSK (TLS Pre-Shared Key) or TLS-PKI (TLS Public Key Infrastructure).

DHCP / Static
When it comes to IP management then the SMB is more flexible than the Enterprise mode. With SMB you can choose between DHCP or static for IP assignment or you can configure it statically, if you use Enterprise mode you can only have DHCP as a option.

When configuring a client with static IP addresses you must use the same IP address for the host operating system and the AMT Management Engine, the hostnames must be unique[8].

If you have configured DHCP then the AMT Management Engine and host OS use the same IP address. If the host OS is powered off and you sniff the network for DHCP packetes than you will see a normal DHCP flow (discover, offer, request and ack) for the Management Engine. You can ping the address that has been assigned to that machine. If you power up the machine you will notice that ping will stop for a very brief moment and that there will be a new DHCP discover, this time initiated by the OS. This behavior is because the Intel AMT DHCP client will release its IP address and will enter a passive state while

the operating system is running. This way the OS can request a IP address and will get the one that the AMT DHCP client has just released. The same thing happens in reverse when you power down the operating system, the OS will release its IP and the AMT DHCP client will become active again.
If using DHCP and Enterprise mode it's recommended[6] that the DHCP client supports Option 81 to report its FQDN with a DNS server. This is important for TLS so that it can reach the machine not only at a IP address but also on a FQDN.

Management consoles support
Be careful with choosing a management console. Not all these solutions support the use AMT Enterprise mode but only SMB mode.

Network Filters
For our experiments that focuses on the network filters we have looked at ways on how the network filter can be deployed. As stated before the configuration settings are sent to the client by using SOAP. By using this developers can easily deploy large numbers of network filters to multiple host. The free "Manageability Developer Tool Kit"[9] that can be downloaded from Intel only supports filters to be deployed to one client at the time. However when using a third-party software vendor these option will be available.

5.2 Access and configure a AMT client

You must set a profile on the AMT client in order to start using AMT. These profiles consists of simple settings like username/password, network settings, certificates and power policy. The AMT client profiles can be configured in three different ways:

- **Manual Entry** Start the machine and press Ctrl+P. All configuration must be done manually on the machine itself.
- **OEM pre-provisioned** Most vendors offer a service to setup a client profile for you. This could be useful if the device is delivered to the user directly without passing the IT department. You must take into account that this service, depending on the vendor, will mostly come with a extra fee.
- **USB One-touch** You can create a AMT client profile from a management station and put it on a USB stick. If you boot the client with the USB stick attached to the machine it will copy the configuration over to the AMT Management Engine.

Once a profile is set you can access it with a management console from a centralized location and perform administrative task, like setting network filters. For our experiments (see section 6) we used the free "Manageability Developer Tool Kit"[9] that Intel made available on there site. This tool kit is supposed

to be a reference implementation for developers so they can use it as a example. Intel advices not to use this tool in production environments and we found it indeed buggy while using. There are a couple third-party vendors that support Intel AMT (for example Altiris[10]).

6 Experiments

In order to verify the effectiveness and usefulness of the Network Defense system a number of experiments have been carried out. For these experiments Intel has borrowed us three HP dc7800 desktop machines that have Intel vPro (which includes AMT) enabled.

The goal of the experiments was to find out a number of things: how effective a blocking filter is, how effective a rate limit is, if a rate limit on one protocol affects the throughput of another protocol, and if setting a number of different blocking and rate limiting filters affects the overall network performance. Each of these questions will be discussed in the following sub sections.

6.1 Blocking outgoing ICMP

For this experiment just one filter was used, one that blocked outgoing ICMP traffic. This filter was then added to a policy.

Two ping sessions were used, one from the machine with the filter to a non-filtered machine, and one the other way around. When the policy was being applied the effect was visible within seconds: the ping sessions started to report time-outs. Upon removing the policy the effect was as could be expected: the ping sessions started to succeed again.

One interesting detail is that upon inspection of a Wireshark[11] packet trace on the filtered machine, is that even after the filter was applied, it showed incoming ICMP request from the other machine's ping session, as well as the outgoing responses. This is due to the fact that the packet is being filtered on the hardware of the NIC, transparent to the OS.

6.2 Rate limiting a single protocol

In order to see how effective a rate limit would be we opened a single stream of data. As all out test machines ran Windows XP, we opted for Windows file sharing. Unfiltered a performance of just over 200Mbit/s was measured, which is not very close to the theoretical maximum of gigabit Ethernet, but nevertheless reasonably close to what can be expected in a real world environment. All file transfers have been done with a 2,5GB large ISO image containing an operating system.

In turn a number of different filters have been applied, ranging from 20 to 10000 PPS, and each time the measured throughput was fairly close to the set set limit, taking 1500 Bytes(the MTU of Ethernet) as a typical packet size.

6.3 The effect of a rate limit on other traffic

In the previous subsection the effectiveness of rate limiting a Windows file sharing session was shown. Such a rate limit, when applied, could have adverse effects on other network traffic that is passing through the AMT NIC. To test

this an HTTP server has been set up to host the same ISO image that was used on the Windows file share.

The tests from the previous subsection have been repeated, but now with a simultaneous HTTP transfer. In al cases the summed total of measured network traffic equalled the maximum throughput measured without filters in place. This indicates that a rate limit that is being used does not adversely affect other network throughput.

6.4 The effect of many non-matching filters

Putting a large number of filters in a policy could slow down non-matching network traffic. This is because a non-matchin packet will have to be compared to all configured filters, for the system to accept it as a non-matching packet. Therefore increasing the number of filters could negatively affect network performance. However, we have not seen this effect in our tests, indicating that the AMT filters are well capable of keeping up with high bandwidth demands.

7 Conclusions

All the test that we have done within our experiments can also be done by most switches. However bringing network filters to the client could have its benefits. With a large number of network filters in place on a single switch the load could increase on that switch and authorized traffic could find hinder in this. Another benefit is that with moving clients (in particular notebook users) the filters stay in place, whereas that with network filters on a switch the filters are mostly bound to a port. This is particular useful for notebook users that work from different locations (read networks) but still want to have some kind of security. In contrast to this you could use a software package that will do the filtering on the clients. One of the benefits for using this kind of approach is that you could let the users take control over the network filters is you are not able (think of a situation where the client needs a filter to be removed but he is not reachable because he is on a different network). However software depends on the operating system that you use and could increase the CPU load on the system. Another thing to keep in mind is that most software filters are not manageable form a centralized location and that would only increase the work load on a IT staff.

This paper only handled the network filters that could be used within Intel AMT but Intel AMT has more benefits like the ones that where described in section 4. Intel AMT nowadays is mostly combined within a Intel vPro solution that is getting adopted by most management software solutions. These kind of solutions offer things like patch management and remote desktop solution. With the integration of Intel AMT (and vPro) in these management solutions there could be a real benefit because with a increasing size of a network to manage a solution to manage desktops from a centralized location is getting more and more important these days.

8 Division of work

- **Michiel Timmers**:
  - Writing first draft project proposal
  - Editing final version project proposal
  - Research focused on Intel Commander Tool
  - Setting up test environment
  - Defining and carrying out experiments
  - Presenting AMT network filters, experiments and conclusions
  - Carrying out demo
  - Writing report, primarily chapters 5 and 7
  - Editing final version report
- **Adriaan van de Zee**:
  - Writing body project proposal
  - Research focused on AMT technology and capabilities
  - Setting up test environment
  - Defining and carrying out experiments
  - Creating first draft, and final editing presentation slides
  - Presenting introduction and AMT theory
  - Presenting demo
  - Writing report, primarily chapters 3, 4 and 6

Bibliography

[1] Intel Active Management Technology
http://www.intel.com/technology/platform-technology/intel-amt/,

[2] Defense and Agent Presence Overview
http://cache-www.intel.com/cd/00/00/32/09/320960_320960.pdf,

[3] OS3: Large Installation Administration
https://www.os3.nl/2008-2009/courses/lia/start,

[4] UvA: System and Network Engineering
http://www.studeren.uva.nl/ma-syst,

[5] Alert Standard Format Specification
http://www.dmtf.org/standards/documents/ASF/DSP0136.pdf,

[6] Intel vPro Technology Quick Start Guide
http://rss.intel.com/click/~rss-146721-c1-151626/download.intel.com/business/vpro/pdfs/deployment_guide.pdf,

[7] W3C: Simple Object Access Protocol (SOAP)
http://www.w3.org/TR/soap/,

[8] Blog: DHCP or Static IP for Intel AMT?
http://www.symantec.com/connect/blogs/dhcp-or-static-ip-intel-amt,

[9] Intel AMT Developer Tool Kit (DTK)
http://www.intel.com/software/amt-dtk/,

[10] Altiris service-oriented management solutions
http://www.symantec.com/business/theme.jsp?themeid=altiris,

[11] Wireshark network protocol analyzer
http://www.wireshark.org/,