Vladimir Porokhin

Ming Chow

COMP 116 Final Project

December 13, 2017

Security of Out of Band Management Technologies

Abstract

Out of band (OOB) management technologies, also known as lights out management (LOM), are ubiquitous and yet often misunderstood. In enterprise settings, where reliability and cost of ownership are of utmost importance, the inclusion of LOM has become a *de facto* standard feature used by system administrators to remotely configure computer equipment and resolve issues in an efficient manner. Over the years, OOB implementations have found their way into almost all computers and these days it's difficult to find a modern device that does not incorporate the technology in some capacity, whether it's a server, a laptop, or a home entertainment PC. By design, LOM has almost an unrestricted access to the machine and is therefore a lucrative target for malicious actors. Unfortunately, the low-level and opaque nature of those technologies makes understanding their security and potential impact a challenging task for the public and even IT professionals. This paper provides an introduction to out of band management, explains risks associated with it, and presents some results from current security research in this area of computing, focusing primarily on the Intel platform.

Introduction

There are two types of strategies used for remote access to computers: in-band and out-of-band management. The terms borrow their meaning from electrical engineering: in-band communication relies on the same channels as the ones used for normal data transmission, whereas the latter uses a dedicated pathway for its traffic [9]. Although this distinction may appear clear at first, there is subtlety when it comes to computer management. Services such as SSH [13] or Microsoft RDP are in-band management systems – they use the normal network connection for communication and operate within the realm of the operating system. Out-of-band access, on the other hand, may or may not use the same physical network as the one used for production traffic and it is generally outside of the operating system's control.

The iDRAC – integrated Dell remote access controller – is an example of a proprietary out-of-band management system used in the Dell PowerEdge line of servers. The system uses a dedicated [10] port for communication and allows the use of a separate network for management, making it physically inaccessible from within the production environment. The HPE iLO – integrated lights-out – is a similar [11] offering from Hewlett-Packard. The IPMI specification implemented by many vendors is an out-of-band system supporting both the use of a dedicated port as well as the integrated NIC, which is sometimes called the "sideband" mode of operation [12]. Intel Active Management Technology (AMT) is another example of such a system and it is particularly interesting because, unlike the previously mentioned technologies found in servers only, AMT is an inseparable part of all

high-end [1] Intel mobile and desktop products.

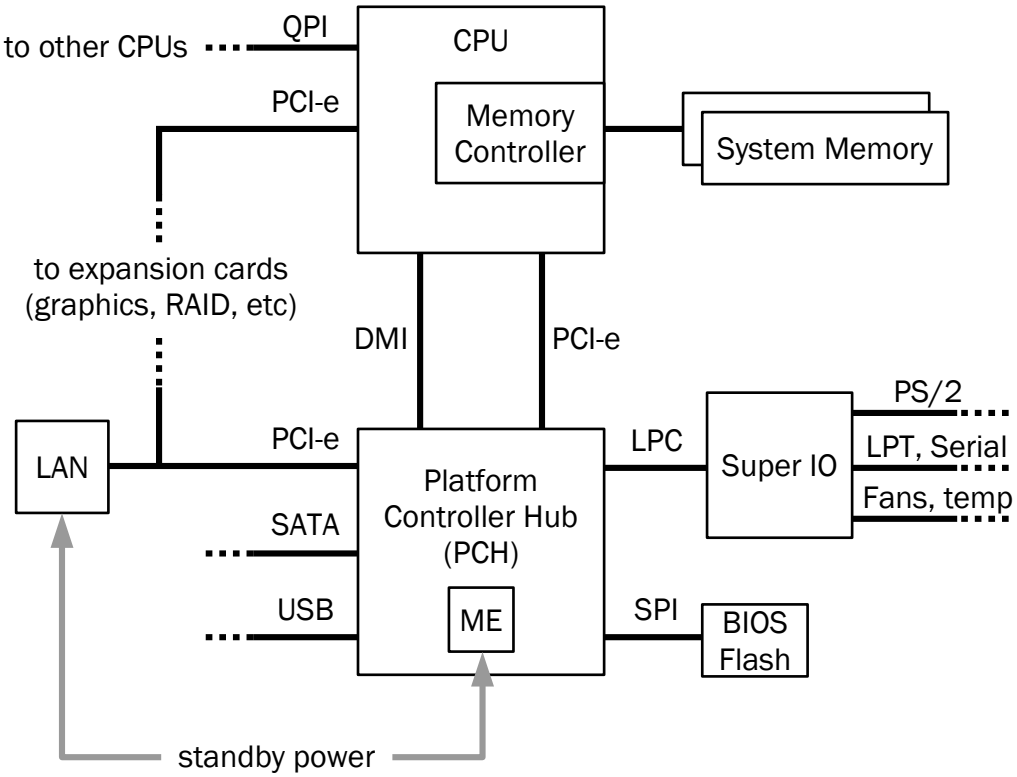

Figure 1: Typical modern Intel architecture showing the ME and the built-in NIC.

AMT is a part of a larger environment called the Intel Management Engine (ME) present in all Intel platforms since 2010 [3]. In its current incarnation, the ME is a separate full-fledged x86 32-bit processor core running a derivative of MINIX inside the Platform Controller Hub (PCH) [4], independently from the main CPU. For communication with the outside world, ME features direct access to both wired and wireless NICs integrated into the motherboard, and any incoming AMT packets are routed directly to the ME, bypassing the OS networking stack [7]. On vPro systems, [8] the AMT process runs together with the ME even when the computer appears to be powered off [7], and while non-vPro machines do not officially support

AMT nor seem to have a permanently-active ME, the hardware is exactly the same for both and there are ways to enable it there as well [3]. As such, the operation of the ME is transparent to the OS and the user, making any malware infection on this level particularly persistent and nearly impossible to detect.

By design, the ME has unfettered access to the entire platform, including the RAM contents, mouse and keyboard events, and video memory [8]. These privileges allow the ME to implement many features useful in enterprise, such as interactive remote control, attachment of virtual storage (IDE-R, IDE redirection) and serial devices (SOL, serial over LAN), and asset discovery on a large network [7]. However, this convenience comes at a great security cost – should an adversary get access to the AMT, or worse yet, the ME entirely, the whole system becomes compromised, and the complexity of the ME architecture provides a substantial attack surface for doing so. It needs to be said that ME cannot arbitrarily access data and peripherals when the main CPU is not operational, but this is of little deterrence considering that AMT allows the computer to be remotely turned on as well.

To the Community

Out of band management technologies are incredibly useful to system administrators, but as new hardware emerges, they appear to become an increasingly common sight in all computers whether the intended market is enterprise or personal computing and entertainment. Yet, despite their profound impact on cybersecurity, they remain a niche interest for researchers and a mystery [1] to the public.

Contrary to its name, ME is not solely a remote management technology. Although that's one of it's most notable functions, on current platforms, the ME is also responsible for system initialization and a number of other features. For example, during the boot phase, the ME configures Boot Guard – a BIOS integrity verification mechanism [5] – and Integrated Clock Control – which is essential for the hardware to function [4]. As another example, the ME implements the Anti Theft technology and a couple variants of DRM – SGX and PAVP [3]. Considering the tight integration into the platform as well as desirable (for vendors) possibilities for vendor lock-in and DRM implementations, it is likely that ME is here to stay and its influence and complexity will continue to expand. Furthermore, since other hardware vendors are not oblivious to those advantages either, they'll be likely to follow suit in the foreseeable future. In this light, the FSF's campaign to make Intel (or any other LOM technology vendor) release their stack as free software [1] is perhaps an unreasonable undertaking, but raising awareness about the issue and inclining vendors to publicly document their technologies is very important and would benefit everyone in the long run.

Just like SQL injection should come as no surprise to web developers, anyone working with sensitive information on their computer should be aware of the features built into their hardware and plan their defense strategy accordingly. "Closing services on a computer that are not necessary," as we discussed in the first week of COMP 116, and using the most up-to-date software alone is no longer sufficient to assure the security of a computer system!

Action Items

Given the nature of the topic, one may ask whether out-of-band management is purely a theoretical threat. First of all, security research in this aspect of computer hardware is very challenging: LOM implementations tend to be vendor-specific and publicly available documentation is very sparse and usually outdated. Furthermore, looking for vulnerabilities requires specialized knowledge of the platform and there are few people capable of doing that. Secondly, exploiting low-level vulnerabilities tends to be difficult because they are often circumstance-specific and, once successfully used, grant access to a very privileged but simultaneously feature-limited environment. These factors help to mitigate the extent of the problem, somewhat, but they do not resolve it completely. The risks associated with built-in remote management still exist and not all vulnerabilities require substantial skills for exploitation. Over the years, several vulnerabilities were found in Intel's implementation, with one recent bug being particularly easy to take advantage of.

In early 2017, a team of security researchers at Embedi found an authentication bypass vulnerability in the platform dubbed "Silent Bob is Silent," which was listed as CVE-2017-5689 [6]. The Intel AMT implementation uses a range of ports between 16992 and 16995 for communication over the network. Ports -92 (HTTP) and -93 (HTTPS) are used for the built-in management web interface, with TLS being optional depending on the platform SKU [7]. It is therefore possible access the web interface of a computer with provisioned AMT, although it requires authentication. For this purpose, AMT supports the Kerberos protocol and HTTP

Digest Authentication described in RFC 2617. The latter implementation turned out to be vulnerable. In particular, the researchers have found that leaving the "response" parameter in the authentication header empty causes a string comparison check in AMT to malfunction and accept invalid credentials as valid ones (see figure 2). This can be done by an unsophisticated attacker using just a proxy server and the consequences are potentially devastating: the adversary can, for example, force the computer to boot their malicious system image through IDE-R, or get interactive KVM (keyboard, video, and mouse) access to the machine [6] and perform arbitrary actions using the privileges of the user logged into the OS, such as installing malicious software or exfiltrating confidential information.

```
GET /index.htm HTTP/1.1
Host: 127.0.0.1:16992
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Authorization: Digest username="admin",realm="Digest:048A0000000000000000000000000000",
                      nonce="qTILAAUFAAAjY7rDwLSmxFCq5EJ3pH/n", uri="/index.htm", response="",
                      qop=auth, nc=00000001, cnonce="60513ab58858482c"
5

HTTP/1.1 200 OK
Date: Thu, 4 May 2017 16:09:17 GMT
Server: AMT
Content-Type: text/html
Transfer-Encoding: chunked
Cache-Control: no cache
Expires: Thu, 26 Oct 1995 00:00:00 GMT
04E6
```

Figure 2: Accessing Intel AMT using CVE-2017-5689. Adapted from [6] and [15].

In 2009, Vassilios Ververis from the Royal Institute of Technology conducted an in-depth study of the AMT security and found a number of trivially exploitable flaws and vulnerabilities. For example, it was possible to circumvent the restriction

preventing local application access to AMT simply by using an external USB NIC. In another example, at the time, the "small-business" versions of AMT had the TLS support purposely disabled, making the users choose between upgrading to an Enterprise setup or using insecure protocols vulnerable to man-in-the-middle attacks. Unfortunately, the Enterprise setup was not secure either because even though it supported TLS, it still transmitted SOL/IDE-R credentials in-the-clear. The researcher also found the AMT authentication process to be vulnerable to brute-force, and its password requirements to be lacking. And lastly, the Zero Touch Configuration feature used to set up AMT on wireless devices, was found vulnerable to a rogue provisioning server attack, even when AMT was disabled in BIOS [7].

Other researches were also able to find arbitrary code execution vulnerabilities in the ME itself, including a buffer overflow issue found in June 2017 and presented at Black Hat Europe 2017 last week [3]. Unlike the authentication bypass problems, code execution vulnerabilities are considerably more difficult to exploit and therefore only accessible to sophisticated attackers. In this case, the risk becomes associated with its value to the adversary – whether it is economically feasible for them to invest substantial resources into weaponizing the vulnerability – more than anything else. However, because the ME technology is so widespread and has the potential to affect a lot of people, it is an attractive target for authors of all sorts of malicious software regardless of the difficulties involved.

There are several ways the risks can be mitigated. Back in 2003, the SANS institute outlined several principles for securing out-of-band management systems, and although the technology has moved on quite a bit since the telephone-based

remote management referenced in the paper [14], the recommendations are still relevant. First of all, it is important to make a list of potential risks and vulnerabilities that can occur, regardless of whether any were found yet, and assign a risk level to each of them. For example, an attacker could access the management system by using its default credentials. In another example, the authentication system may contain a hypothetical bug that permits access without using valid credentials – even though it may not be the case, assuming that the management system is inherently vulnerable can only improve its security. Then, possible mitigation strategies should be considered for each and implemented, if possible and feasible according to the risk level. Following through with this approach can help prevent many common security problems and ensure that a single point of failure does not compromise the entire system or network. For the first example, the administrator would have to make sure the default credentials are changed. In the second example, the adversary should be denied any opportunity to interact with the management system in the first place – by physically separating the production and management networks, for instance. In the case of ME, this can be accomplished by using a separate non-Intel-supported NIC for normal traffic and the built-in one for management [1], or configuring the networking equipment to permit access to the management ports only from specific machines. Of course, it also helps to keep the ME software up-to-date by installing the latest BIOS updates available from the manufacturer and keeping track of new developments in the world of cybersecurity.

Conclusion

Modern management systems built into modern platforms are remarkably sophisticated. In effect, they can be considered as computers "within" computers, with their own CPUs, operating systems, and vulnerabilities. Since they are tightly controlled by the manufacturer and not widely known to the public, keeping them secure and up to date is difficult. Furthermore, because they provide unrestricted access to machines and are becoming increasingly widespread, they are an attractive target to malicious actors.

Since the introduction of AMT in 2005, researches have uncovered multiple vulnerabilities in the technology. The ease of their exploitation varies, with some of them being trivial to take advantage of. Other bugs pose a more theoretical threat, albeit of no smaller proportions. Knowing that the technology is there and keeping in mind the potential risks associated with its presence is half the battle in building a secure system. The other half is acting on this knowledge – responding to new vulnerabilities in a timely fashion and following best security practices.

References

[1]: Ward Vandewege, Matthew Garrett, and Richard M. Stallman, "'Active Management Technology': The obscure remote control in some Intel hardware." *Free Software Foundation.* June 2014. https://www.fsf.org/blogs/community/active-management-technology

[2]: Erica Portnoy and Peter Eckersley, "Intel's Management Engine is a security hazard, and users need a way to disable it." *Electronic Frontier Foundation.*

May 2017. https://www.eff.org/deeplinks/2017/05/intels-management-engine-security-hazard-and-users-need-way-disable-it

[3]: Mark Ermolov and Maxim Goryachy, "How to Hack a Turned-Off Computer, or Running Unsigned Code in Intel Management Engine." *Black Hat Europe 2017*. December 2017. https://www.blackhat.com/docs/eu-17/materials/eu-17-Goryachy-How-To-Hack-A-Turned-Off-Computer-Or-Running-Unsigned-Code-In-Intel-Management-Engine.pdf

[4]: Mark Ermolov and Maxim Goryachy, "Disabling Intel ME 11 via undocumented mode." *Positive Technologies*. August 2017. http://blog.ptsecurity.com/2017/08/disabling-intel-me.html

[5]: Embedi Researchers, "Bypassing Intel Boot Guard". *Embedi Blog*. October 2017. https://embedi.com/blog/bypassing-intel-boot-guard/

[6]: Embedi Researchers, "Silent Bob is Silent." *Embedi Whitepaper*. 2017. https://theswissbay.ch/pdf/_to_sort/Silent-Bob-is-Silent.pdf

[7]: Vassilios Ververis, "Security Evaluation of Intel's Active Management Technology." *KTH Information and Communication Technology*. December 2009. https://people.kth.se/~maguire/DEGREE-PROJECT-REPORTS/100402-Vassilios_Ververis-with-cover.pdf

[8]: Anonymous authors, "RE: How to disable Intel ME?" *StackExchange Information Security*. March 2017. https://security.stackexchange.com/a/154877

[9]: T. Andrew Finn, "Chapter 11: Signaling Systems". *Telecommunication*

Technologies. George Mason University. http://mason.gmu.edu/~afinn/html/tele/tech%20chapters/T11.htm

[10]: Miguel Fra, "How to Remote Control Your Dell Server using Integrated DRAC (Dell Remote Access Controller)." *Falcon IT Services Knowledgebase.* June 2012. https://www.falconitservices.com/support/KB/Lists/Posts/Post.aspx?ID=55

[11]: Paul Reissner, "RE: How do you access HP ILO web interface on server running IIS?." *Spiceworks Community Forum.* August 2012. https://community.spiceworks.com/topic/253941-a

[12]: Paul Paradise and Jeff Atwood, "RE: Configuring SuperMicro IPMI to use one of the LAN interfaces instead of the IPMI port?" *StackExchange Serverfault.* 2012. https://serverfault.com/a/362019

[13]: Boris Tulman, "In-Band and Out-of-Band Network Management." *Learn Computer.* May 2010. http://www.learncomputer.com/in-band-out-of-band-network-management/

[14]: Marc S. Kolaks, "Securing out-of-band device management." *SANS Institute InfoSec Reading Room.* 2003. https://www.sans.org/reading-room/whitepapers/networkdevs/securing-out-of-band-device-management-906

[15]: Dmitriy Evdokimov, "Intel AMT vulnerability. Life after CVE-2017-5689." *Black Hat USA 2017.* July 2017. https://www.blackhat.com/docs/us-17/thursday/us-17-Evdokimov-Intel-AMT-Stealth-Breakthrough-wp.pdf