Aspects of Information Security of Computer Systems

Dmitry P. Zegzhda [1] and Igor Y. Zhukov [2]

[1] *Peter the Great Saint-Petersburg Polytechnic University, Polytechnicheskaya st. 29, St.Petersburg, 195251, Russia.*
[2] *JSC «RAMEC VS», Volgogradsky Prospekt, 2, Moscow, 109316, Russia*

Abstract

Presented is a comprehensive analysis of the issues of technological independence and information security of the computing system. The systematic approach involves the analysis of protection technologies at all levels of architecture and its interaction with each other. Hardware protection technologies at the processor and command system level are considered. Hardware support for virtualization turns out to be a necessary security function and is not less important than the already traditional security functions like an access control, identification, authentication, audit and security control. In addition to hardware support for virtualization we discuss the implementation of the group of technologies for hardware support for a separate trusted environment in modern processors. The analysis of the possibility of using hardware protection technologies by intruders for malicious influences is carried out. Based on the research results, an approach to create a domestic protected architecture of computer equipment using foreign hardware protection technologies is proposed. At the same time, import substitution should not be limited solely to the replication of foreign solutions, since foreign computer equipment contains a lot of undocumented capabilities and, as a result, pose a threat to information security.

Keywords

Computer system architecture, hardware protection technologies, virtualization, malicious impacts, technological independence, import substitution

1. Introduction

Over the past few years, modern hardware protection technologies have developed strongly [1-7]. These technologies are the source of pride of any IT-company. In this article, hardware protection technologies are divided into ten groups:

- Privileged mode support and address space control
- Protection against threats from hardware devices
- Hardware support for independent system integrity monitoring during the boot process
- Hardware support for virtualization
- Hardware support of a separated trusted environment
- Countering exploitation of vulnerabilities
- Implementation of cryptographic primitives and secure key storage
- Hardware-based identification and biometric user authentication
- Protection against hardware malfunctions
- Autonomous system control chip

BIT-2021: XI International Scientific and Technical Conference on Secure Information Technologies, April 6-7, 2021, Moscow, Russia
EMAIL: dmitry@ibks.spbstu.ru (A. 1); i.zhukov@inbox.ru (A. 2)
ORCID: 0000-0002-0232-7248 (A. 1); 0000-0002-4429-8799 (A.2)

Each group contains several decisions at once, and all of them are seek to address one of the system's security concerns.

Privileged mode support and address space control is base technology that almost all modern microprocessors have.

First of all, it builds two necessary processor modes: division of software into system one – working in privileged mode, and application software – working in processor «user» mode.

Second technology of this group is realization base control recourse to isolated address space. Isolation function provides the opportunity of mistake localization within the same application process and it doesn't break the whole system work. Overall, it creates base entities that can be authenticated by subjects that are needed to provide circulation control.

Technology of virtual memory is being implemented in the context of «protected mode» which solves several problems at once: isolating every system process from one another in separate address space and providing every process with the same size of address space no matter what the real amount of RAM is. This technology is implemented within MMU (memory management unit) which is the part of almost any modern microprocessor.

The second group of technologies **protection against threats from hardware devices** are technologies providing functions to monitor the handling of external (relating to CPU) devices to hardware devices: RAM and so on.

Modern devices have learned to work with one another, CPU, and memory at high speed, mostly after creating DMA technology (direct memory access). This technology actually gave appliances access to RAM not using CPU. In this case processor becomes just one of RAM consumers [6].

Main threat from hardware devices is implementation of malicious software in firmware.

Good example of the device like that is modern video adapter that has got his own RAM (video RAM), a powerful computer with hundreds of processor cores, complicated software and realizes virtual memory functions. Thus, delimitation of external device circulation to RAM and their mutual isolation are current tasks [8].

Trusted Download Hardware and Software Modules (TDHSM) are widely used domestically. The purpose of these systems is integrity ensuring and system configuration with step-by-step checking the integrity of the hardware and software composition during the whole loading procedure. The result is system user's confidence that malware has not implemented into the system and the integrity has not been violated since previous launch. It is highly necessary for achieving the security of any information system.

The point of traditional TDHSM is in static integrity control when inspections are carried out strictly consistently: each successive loadable component is firstly checked by the integrity control system and only after that (successful check) component is executed.

If the execution sequence is out of order or one component is skipped, there will be a serious violation of security that can be used by an offender. To start the system in a trusted way, integrity control needs to be realized for each component, also for the ones that are only intermediate and are not required for long-term operations by users. Meanwhile, traditional technology is already in use in the domestic market and uses certified algorithms in its composition.

However, interesting technology of dynamic integrity control was developed by companies Intel and AMD [9, 10]. The point of dynamic integrity control is that in the initial stages of system start-up, integrity checks are not performed at all. Then right before OS kernel run (a trusted component of the entire core system that users will be working with) the system configuration integrity control is monitored in an atomic way - in a single instruction (e.g. there is an instruction GETSEC[ENTER] in Intel TXT). It turns out that the technology is embedded in processor itself. Anyway apart from involving processor, it also involves capacity of chipset and external security devices. Processor checks if only one kernel works in a particular way, interruptions are turned off and all the system is configured properly (so that no other devise or code cannot break or stop further checking); and later it executes control over software integrity and system configuration in accordance with chosen security policy. It all happens unconditionally and atomically, this fact provides robust integrity control mechanism. Furthermore, this instruction may be called at any time and not once.

Unfortunately, dynamic integrity control technologies do not use domestic cryptographic algorithms and cannot be widely used in domestic market. These functions require realization inside processors.

Of course, this fact meets the requirements of the regulations in terms of purpose, but not in terms of control algorithm methods that are used. Nonetheless, it is possible to change algorithms and get all the advantages of this powerful technology.

Hardware support for virtualization is very important. It provides support of OS virtualization from hardware. We can say that virtualization is becoming a necessary security function and as important as classic functions, like access control, identification, authentication, auditing and security control. The high level of importance of virtualization is justified by fact that it is the only function that can provide and guarantee the isolation of few computing environment from one another.

Actually virtualization splits hardware platform into separate program platforms (virtual machines). It is important that these exact platforms include full-fledged OS with the whole variety of application and system software that are placed inside virtual machine controlled by hypervisor.

In practice this effect is reached by appearing two mode groups in processor: virtual machine monitor modes (root or host) and virtual machine modes (non-root or guests). Also apart from traditional MMU there is an additional layer of control and virtualization of RAM.

The concepts of hypervisor (it is also called as virtual machine monitor) and virtual machine were introduced in 70s of 20th century. However, due to creation of hardware support for virtualization and development of cloud-based systems, the concepts (VM and VMM) are widely used in IT-industry these days.

Nowadays the ensuring of computing environment isolation is very urgent task. It is more and more often needed to use information systems consisted of several operating systems and part of them is trusted and the other part is not. It is important for users to use all the potential of every part. Composing of trusted and untrusted parts together in the context of the same OS is impossible! Only function of virtualization can provide their guaranteed isolation [9, 10].

As OS virtualization function is highly important for security ensuring, its hardware support is the cornerstone of many modern security features. In fact we know that hypervisor can be employed with no hardware technologies. The question is how effective it will be and how secure it will be in terms of reliability of operation.

Besides isolation function, OS virtualization is able to provide also a number of useful functions that are very important as well:

- Control over all software tools, including OS kernels, virtual machines
- Management of input/output flows between OS and external devices
- Ensuring of protected data exchange among virtual machines
- Control over nested hypervisors and virtual machines

Actually hypervisor can manage the behavior and even prevent unwelcome behavior of devices that are untrusted. The last function is quite interesting because it may be used recursively: hypervisor enables other hypervisors to run in a virtual machine. This hypervisor is referred to as «nested», just like his virtual machines. In that case, the nested hypervisor is not privileged anymore and the primary principle will be applied: the first launched on PC hypervisor is the privileged one and all other nested hypervisors are subordinate.

In addition to hardware support for virtualization in modern processors, there is another group of technologies: **hardware support of a separate trusted environment.**

The trust problem is not only in Russia, it is a global one. Even Intel and Microsoft are willing to create a kind of an environment with a high level of trust within a common computing space [11-13]. Such isolated trusted environment is necessary for key handling and other sensitive confidential information. In this case a full operating system is not needed, so virtualization function is not needed as well, the hardware support group of a separate trusted environment includes the following technologies:

- Placing privileged code in a protected area of memory
- Separation of the computing environment into trusted and untrusted
- Obligatory signature verification for trusted code

In order to achieve this objective, appropriate technologies have been developed to ensure, first and foremost, integrity and authenticity of trusted medium [14-16]. Moreover, they limit interactions between trusted environment and the rest of the system. Actually it is the traditional protection but implemented directly within the hardware architecture.

Within this group, several interesting technologies should be highlighted and described in more detail.

Firstly, it is important to highlight ARM TrustZone technology [13] that makes available to separate code, data and external devices into two domains (trusted and untrusted). This not only controls the integrity of the trusted domain at the hardware level, but also prohibits external devices from accessing trusted domain data.

Secondly, SMM (System Management Mode) technology. This technology is known for a long time. At the beginning it was solving the problems that had almost nothing common with security. As stated in the official documentation, SMM is the most privileged processor mode nowadays. Code in this mode is isolated from the rest of the code and hardware devices have no access to it. De facto SMM forms a full-fledged isolated environment.

Thirdly, the most «recent» technology should be considered – Intel SGX (Safer Guard Extension). This technology is interesting because it, unlike the others, provides creating the isolated computing environment within unprivileged application OS process. In classical information systems architecture, the more privileged component is usually more secure and trusted. Intel SGX technology is different: it makes possible to create an enclave (protected part of code and data) within unprivileged application process. Also the access to the enclave is denied to other processes, the OS kernel, the hypervisor and any external devices at the hardware layer. Such an interesting result is achieved by storing code and data in encrypted form in RAM. This powerful move was realized thanks to Intel modern processors that complete many cryptographic functions right inside crystal. As the result, the enclave's data exist in unencrypted way only deeply inside processor kernel. Also technology provides code integrity hardware control that enters the enclave by means of signature-based hardware check before the code is executed.

Countering exploitation of vulnerabilities. From the point of view of practice protected system is the one that has no vulnerabilities. Vulnerability may be defined as a specific kind of code mistake that allows system security to be compromised. Any program has got some mistakes but not any mistake is vulnerability. Not all errors allow attack to be carried out, much less allow attackers to get into the system [18].

Apart from that, there is notion «zero-day vulnerability». This is the type of vulnerabilities that no one in system knows about, so that no security system can confront it. Consequently, this vulnerability is available for an attacker [19].

That is why OS and processors developers introduced measures to counteract vulnerabilities, what effectively formed the second echelon of protection. In others words, if in the first echelon (standard controls handling in all trusted boot loaders, hypervisors etc.) there is a vulnerability that an intruder has found and used, the second echelon creates range of obstacles that an intruder has to deal with.

Type enforcement: the processor is told where the code, the stack and the data are, what can be executed, what cannot be modified, etc. In addition, a whole group of technologies prevent the transfer of control from one privilege level code to code at another privilege level:

- Control over execution, reading, writing for virtual address space
- Cache type management for memory areas
- Segment boundary management and type enforcement
- Prohibiting recourse to unprivileged code from privileged mode

All of the above are effective enough functions. They really work in practice and provide the second echelon of protection. The architecture of modern OS often prevents the elements are very similar to the exploitation of vulnerability. For example, antimalware or virtualization tools often use techniques that ease the software development and rise up productivity but at the same time they do not comply with safety requirements and break the rules, making their behavior look like vulnerability. This is why it is unfortunately not possible to apply these protection technologies 100%, even though the perpetrators are being further hindered.

However, the need for technology to protect against exploitation of vulnerabilities has necessitated a change. As the result necessary functions have been added in virtual paging (virtual page memory model). This approach let developers mark out code, stack, data and other memory segments as they want. This specifies the access attributes to the exact page (4Kb): for a given memory area: reading, modification, execution, systemic and applied areas. This approach turned out to be simple and easy so that it was integrated into all modern OS.

Cryptography is important security function. Full-fledged implementation of cryptographic functions that provides both acceptable productivity and key information protection is needed for modern systems. Let's look at a few of these technologies in more detail.

Firstly, FPGA systems are typically used to solve cryptographic function acceleration problems, which is essential for modern systems. However, modern processors solve this problem by adding new instructions: Intel AES-NI and MIPS OmniShield, for example. These technologies implement western cryptographic algorithms that are already built into the processor. However, Intel has already long ago announced the Intel Xeon FPGA technology, which should allow software developers to add their own instructions to the processor to implement cryptographic primitives that users need.

Secondly, the popular TPM (trusted platform module), which is an insulated chip whose main task is to provide a secure key storage. This is achieved by the fact that the chip itself stores a unique key that never leaves the chip and is physically protected physically from all sorts of X-ray scanners and other means of chip reversing (reverse-engineering). The chip itself implements many cryptographic functions (more than 20) which can use the same internal key.

Biometric authentication is another important function that is important in terms of system architecture. The need for authentication arises all the time and users cannot create that many keys and passwords, and most importantly, remember them. This creates a huge risk of leakage and compromise of the keys [20]. Biometric authentication is very easy to use and minimizes this threat. Therefore, support for such authentication is also being built into in many modern computing systems and is also an element of architecture.

Errors in software can lead to vulnerabilities and attacks on systems, while errors in hardware and natural influences on hardware can lead to data loss and denial of service. Therefore, modern architecture also equips itself with hardware fail-safes to overcome such problems.

In other words, in addition to mistakes made by developers, external factors can also cause data integrity issues. Therefore a number of technologies have been introduced into modern systems to protect against hardware failures. These technologies are implemented almost on every hardware component of modern PCs. They are transparent to users and programmers.

What constitutes trust is a rather complicated question. It is clear that trust has a kind of hierarchical nature: where some components depend on others. For example, the operating system controls the application. Accordingly, the application trusts operating system. At the same time the hypervisor controls the operating system and the operating system trusts the hypervisor. The processor controls the operating system and the hypervisor. Consequently, the hypervisor and the operating system trust the processor.

What does processor control? Earlier external devices on the motherboard did not include control functions and were quite simple. Over time, however, a chip has appeared on the motherboard of modern PCs that implements independent control and management of the system. This is a chipset which is often referred to as a «south bridge». The motivation for this development was the need to implement remote system diagnostics, management and disaster recovery. The need has arisen because today's system administrators are required to maintain a large number of computers simultaneously. This is not only about huge data centers with cloud-based systems, but also about standard enterprise segments with standard PC configurations. As a result, virtually all modern computers on the market are equipped with this independent stand-alone chip control (Figure 1).

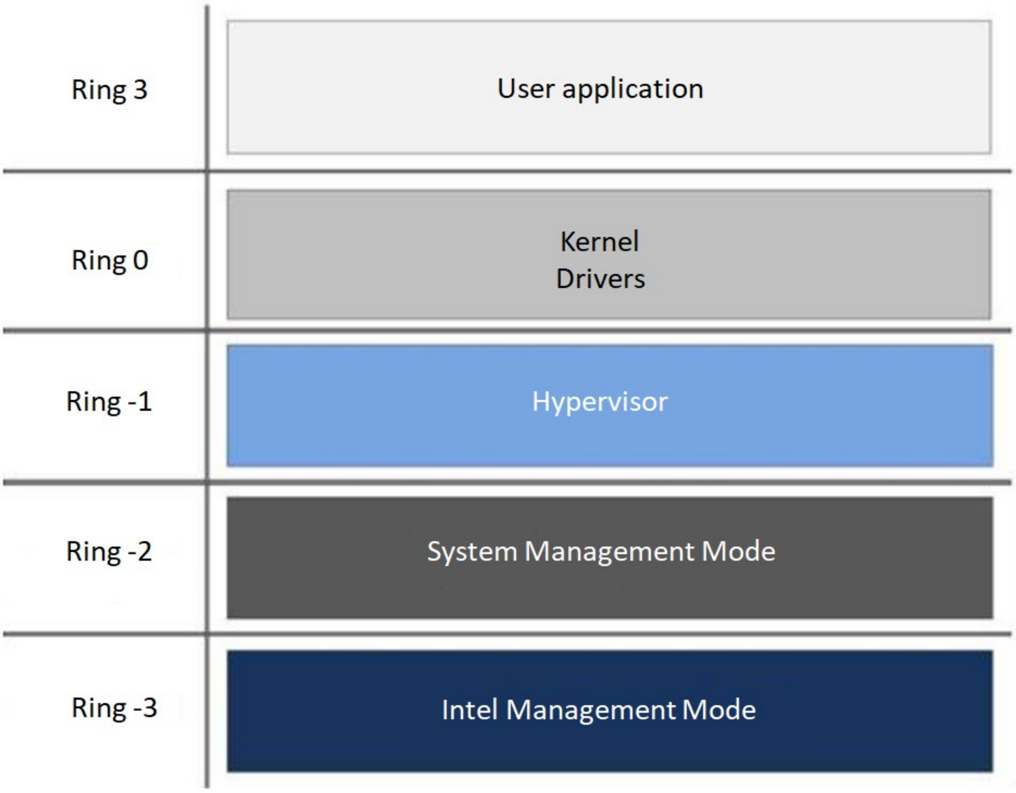

Figure 1: A new level of control and management in the system – «chipset»

Remote control and diagnostic are not the only technology implementing by this chip. Particularly, autonomous control chip plays a role in implementation of anti-theft and copyright protection features on laptops (so-called DRM – digital rights management). If stolen, this chip can delete all the confidential data from the computer on its own, or inform the user about its location.

Actually autonomous control chip is a computer inside another computer. He has an access to RAM, video card, power management system, the ability to filter network traffic and to communicate actively and independently over the network (e.g. there is a full-fledged web server implemented as part of Intel AMT technology). The description of the security technologies that are implemented with this chip suggests that all of these features are indeed necessary for the chip. However, it is fairly obvious that there are actually many more security features on this chip. This is because a certain level of trust is developed in the application, the operating system and the processor, but in fact this autonomous chip, the «south bridge», needs to be trusted because the rest of the system depends on it. Unfortunately, the issue of trust of this chip becomes especially complicated due to almost complete absence of his documentation.

As a result of research, Table 1 has been compiled, presenting all the considered technologies and their availability in the various modern processors.

Table 1 shows that Intel processors are the leader in terms of the number of security technologies implemented. This conclusion is quite clear, as Intel is currently the leader in the microprocessor market. This company produces a full range of microprocessors, motherboards, graphic and network adapters, chipsets and other devices. In doing so, Intel microprocessors are found in all types of modern computing from smart home controllers (Intel Edison) and mobile phones to supercomputers.

Table 1
Prevalence of hardware protection technologies among modern microprocessors

	Intel x86, (Skylake)	ARM Cortex-A, Baikal-M	AMD Bulldozer	MIPS Varrior Baikal-T1	VIA Nano	PowerPC	Elbrus	Sobol	TPM	Intel 100 series	CRYPTON-LOCK
Protection against threats from hardware devices	yes		yes					yes		yes	
Hardware support for independent system integrity monitoring during the boot process	yes	yes	yes					yes	yes		yes
Hardware support for virtualization	yes	yes	yes	yes	yes	yes					
Hardware support of a separated trusted environment	yes	yes	yes	yes			yes				
Countering exploitation of vulnerabilities	yes	yes	yes	yes	yes	yes	yes				
Implementation of cryptographic primitives and secure key storage	yes	yes	yes		yes			yes	yes		yes
Hardware-based identification and biometric user authentication								yes			yes
Autonomous system control chip	yes		yes							yes	
Protection against hardware malfunctions	yes	yes	yes	yes	yes	yes	yes	?		yes	?

Popularity of Intel processors and x86 architecture is inextricably linked to two aspects.

Firstly, popularity provokes a lot of attention, which in practice makes the security issue the most pressing one for this exact platform. Before the advent of mainstream computers the information security issue has not been so acute. Anyway through widespread use, unskilled users and consumers have come into contact with this technology. Also the security problem has arisen because security without people exists. Due to this popularity, as much protection as possible has been demanded from Intel processors, as they are the ones with the highest number of threats.

Secondly, the life and technology development have impact on choice of architecture or process, so that it is more important if a company has a large amount of software for its platform and therefore if there are enough developers and users trained in that software. With such «leverage», Intel is intensively introducing its technology into all other segments. As the result, all powerful supercomputers are based on the Intel architecture because users want to use popular software already developed for x86 architecture.

2. Computer equipment security technologies

The widespread availability of x86 PCs and their software development systems is because of the distribution of this architecture into all computer equipment classes.

Modern PCs have gained both huge popularity and all possible range of functionalities: they take part in network communication, can be powerful computing equipment, can use different types of sensors, manage simple mechanisms, interoperate with WiFi/GSM radio channels, launch VMs, redirect network traffic and provide various network services to other users etc.

Overall, PCs are universal computer equipment. With the development of modern technology, it has become clear that a move away from versatility towards device specialization can lead to better results and greater benefits. The possibility of identifying two major trends in the development of the personal computer has led to emergence of new types of computer equipment, which are shown in Figure 2.

Firstly, the development of the PC towards more computing power and parallelism has led to the emergence of supercomputers. In addition, the interconnection of many computers into a network has led to the development of internet servers and cloud-based computing.

Secondly, reduced power consumption and lighter functions of modern processors has led to the emergence of mobile devices. As a development of this trend, cyber-physical systems emerged in which computer hardware acts as low-powered but «smart» sensors. In addition, reduced power consumption and the focus of computing technology on a single task have led to the emergence of telecommunications equipment.

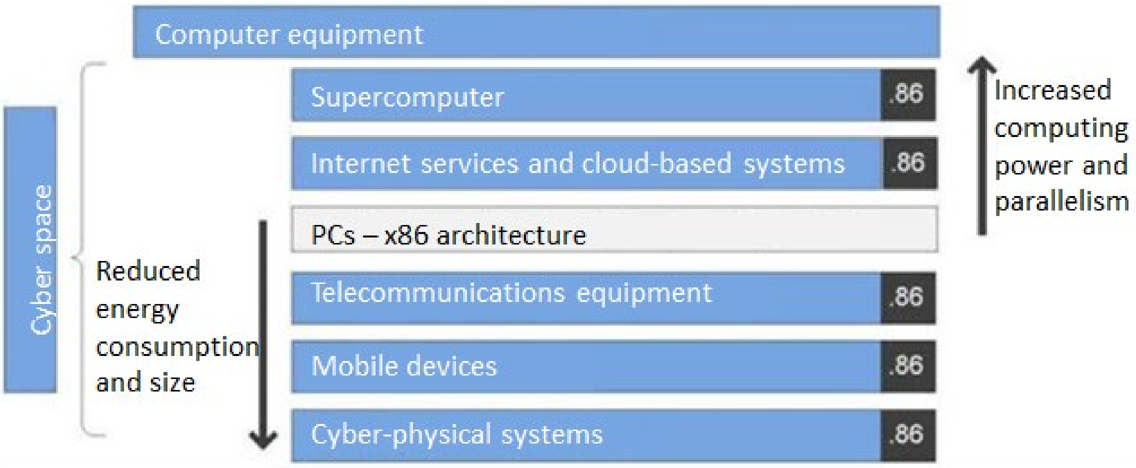
Computer equipment
Supercomputer
.86
Internet services and cloud-based systems
.86
PCs – x86 architecture
Telecommunications equipment
.86
Mobile devices
.86
Cyber-physical systems
.86
Cyber space
Reduced energy consumption and size
Increased computing power and parallelism

Figure 2: The emergence of various types of Computer Equipment from Personal Computers

Dedicated hardware protection technologies are effectively used in one of the most widespread infrastructures in the world for Apple, using security technologies at the processor, computer equipment (computer and/or mobile device), operating systems (macOS and/or iOS) and cloud-based system (AppStore, iTunes).

3. Threats of hardware-based protection technologies

However, some hardware protection technologies can be used not only for protection tasks, but can also be used by attackers to enhance their means of attack. Two of the hardware-based defense technologies discussed have such double-edged features:

- Autonomous system control chip
- Hardware support for virtualization

Consider how these hardware technologies can be exploited by malicious software. Malware can use hardware virtualization technology to take over control of a system without disrupting its operation.

In addition, malware can use this technology to hide its presence and its activity. This is because the control devices (hypervisor) are in fact a separate target of an attack. In doing so, the attack is directed to places where there are no means of protection. This is completely impossible to ignore. Practice shows that it is already known that there are attacks on the hardware and on the software sides that are heavily involved in hardware technology.

Figure 3 shows the most known attack named «Blue Pill», that was demonstrated in 2006 by Joanna Rutkowska at the Black Hat conference. The point of this attack is that software tool Blue Pill becomes hypervisor instead of Windows program. Hypervisor creates a VM and moves OS into VM, afterwards hypervisor controls OS's behavior and hides himself. In fact, attacker creates virtual operating system, uses his tools in hypervisor outside the OS. In doing so, the intruder becomes absolutely transparent and invisible to security equipment. Security equipment operates within OS and plays by the rules that the attacker is already able to change. As a result, an attack like that will not be detected by anti-viruses or any other protection tools. It is important to emphasize that this is not because defenses are bad, but because they fundamentally cannot detect the impact on system because it lives inside this VM hosted by an attacker. Since such a practical experiment has been demonstrated, it is reality, not theory.

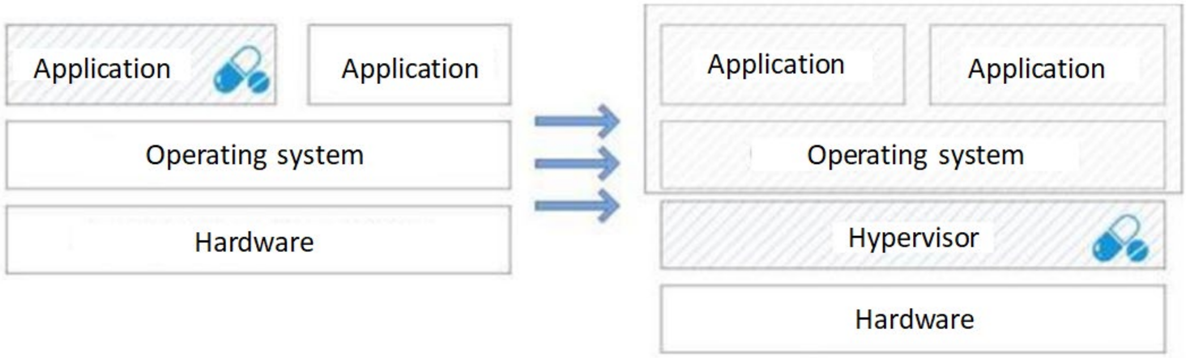

Figure 3: Using virtualization in the Blue Pill experiment

Other group of protection hardware technology that a malware can use is autonomous system control chip. This group of technologies will be discussed using the examples of Intel ME (Management Engine) and Intel vPro for illustrative purposes. This technology literally implements autonomous OS based on chipset (the second-most important hardware chip in computer after processor). As the result chipset is more attractive aim than processor and OS because after entering the Intel ME operational environment it is possible to do all the actions that an attacker usually wants to. In doing so, attacker stays absolutely invisible and unreachable to security equipment launched on CPU because implementation of these functions is outside of its OS.

Originally, Intel ME technology is intended to control the process and other computer hardware. Besides, kaleidoscope of different technologies is based on Intel ME technology and it allows detecting computer's state in data-centers remotely, turning off and deleting confidential data remotely, uploading from virtual DVD disks, installing operating systems, limiting access to given screen places to protect video materials from copying, performing network packet filtering, hiding active interaction with network (including Wi-Fi). As experts think this technology contains the most serious threat. The list of known possibilities if there is a successful interception of control over Intel ME:

- Control over computer equipment at any stage of OS launching
- Computer system environment manipulating
- Protection hardware technology manipulating
- Having own DMA-controller avoiding MMU
- Maximum isolation of ME/AMT executable code from OS
- Continuous operation on standby power
- Access to OnBoard-devices (GPU, USB/Ethernet/Wi-Fi/NFC)
- Remote control of computer environment (centralized/decentralized schemes)

- Listening to network traffic / collection of password-address information
- Available redirecting/modifying/mirroring of network traffic
- Organizing a p2p-network to go beyond the LAN
- User activity monitoring
- Image spoofing
- Collecting an information from all connected networks (flash drives / telephones / tokens / sensors)
- Interfacing with OS APIs to collect user information (phone address book/media content/token data)

How to counteract the attacker that can use such dangerous architecture opportunities like hardware support for virtualization and autonomous system control chip? First of all, there is a temptation to refuse from these technologies: if an attacker can use them then it should be deleted from the system.

As experts think, it is an incorrect approach because it is a step back, because not only attackers but also users and developers will not be able to use for benefit purposes. Virtualization technology and Intel ME really give useful and wonderful opportunities that improve efficiency and solve more difficult problems. The easiest example: data-center cannot be built without these modern technologies. The economic costs will be too high without virtualization and due to absence of technologies like Intel ME it fails to ensure the administration of a center like this and carries out timely identification of hardware failures that happen pretty often. Actually there is no need to abandon the technology. This information security issue has got a decision.

Consider virtualization hardware support: how is it possible to avoid use of this technology by attacker? Firstly, the hardware-based independent integrity control technology in the loading process works effectively. This technology guarantees that hypervisor cannot be embedded into set PC weaving. Secondly, this system control chip (even Intel ME) should be used to prevent hypervisor's lock out by attacker or installation of hypervisor (like Blue Pill) into the system. Consequently, these technologies form a hierarchy, which is shown in Figure 4.

A hierarchy is characterized by control sequence: technology with higher priority in terms of hardware can control another technology; the base of hierarchy is an autonomous system control chip because it is on the lowest system layer because it is not even a processor, it is a separate chip. Obviously, if attacker intends to override the defense and cannot do it point blank, he needs to get around this on the lower layer, and that is what Joanna Rutkowska successfully did not only in 2006, but also in 2009 when she has embedded her code into Intel ME.

It turns out that in order to ensure security, trust must be placed in the component that is at the lowest level of the hierarchy. As a result, this component has the highest level of management and control, and therefore is the root of trust.

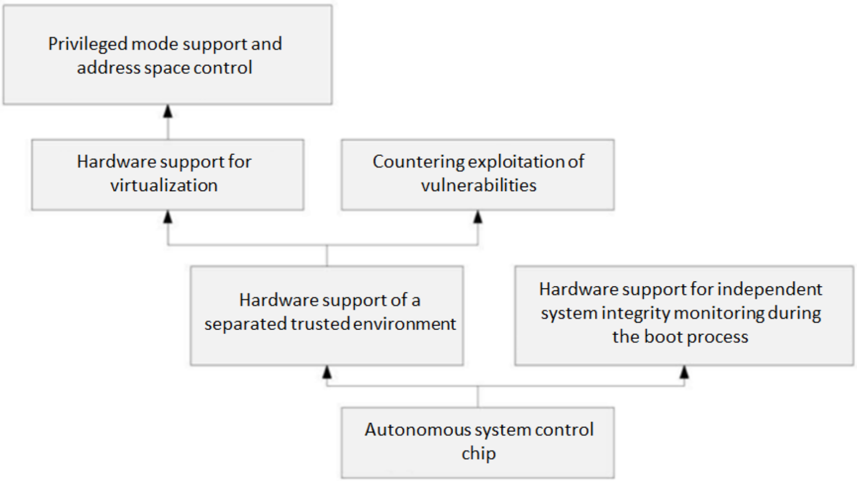
Privileged mode support and address space control
Hardware support for virtualization
Countering exploitation of vulnerabilities
Hardware support of a separated trusted environment
Hardware support for independent system integrity monitoring during the boot process
Autonomous system control chip

Figure 4: Hierarchical dependence of technologies

It is used to be thought that this component was processor kernel. However, it is clear nowadays that this component is extracted from processor. As experts think, this interpretation of term «trust» is what import substitution should boil down to. Import substitution must not replicate something that already is on the market. It is necessary to concentrate on the system component that controls all computer equipment elements. The scientific and technical backlog also must be taken into account to provide a higher trust level in the whole system.

Domestic computing equipment architecture should include autonomous security feature that completely controls the use of the information system and provides security and safety [21].

4. The architecture of protected computer equipment

Considering the issue of building modern computer equipment using overseas technologies, some important moments should be highlighted.

Firstly, import substitution should not be confined to replication of foreign decisions. Instead of this, it is better to create and improve domestic ones. It allows providing the encouraging domestic production and the development of scientific and engineering research and implementation.

Secondly, there is the previous reported problem: foreign computer equipment contains a lot of undocumented features (UDF) and therefore there is a threat of information security. In doing so, reliable analysis with UDF for high-tech computer equipment is impossible in practice. Nonetheless, it is impossible to completely abandon from compatibility of domestic computer equipment with overseas program software and hardware resources too. The solution is that architecture of modern computer equipment has to include autonomous hardware protection completely controlling functioning of information system and ensuring trust and safety.

Thirdly, the necessary condition to implement a solution is relevance by consumers, both corporate and mass-produced.

Fourthly, in order for a piece of hardware or software to be used in the public sector it must be safe, which is guaranteed by the implementation of a certification procedure. This procedure confirms that a number of requirements specified in the legislation have been met. At the same time, the existing domestic secure hardware has a number of disadvantages comparing to overseas analogues:

- Incomplete compatibility with modern mass-produced hardware. As the result, continuous adaptation to new devices, whose diversity generates unique requirements to each of them, is necessary
- Limited compatibility with apps that is called by necessity of software and weaving version fixing and absence of certain functionality what is the consequence of meeting the requirements in terms of certification procedure
- Limits on external environment. Once certified, security guarantees remain only when working in a trusted environment. However, there are no guarantees when the tool interacts with an untrusted environment
- The need for a certification procedure actually means a lack of confidence in the remedies, as their effectiveness is only recognized while working with a limited set of certified solutions

Consequently, it is necessary to develop a modern domestic computer equipment architecture that takes into account all of the above factors and if possible, has no indicated disadvantages. For software protection, it is proposed to use the so-called integration paradigm, which is based on four postulates:

- Control over all interactions in the system
- Protection invariance with respect to applications and resources
- Control of information interactions that is based on strictly determined rules making up the formal model
- Assessing the safety of both the current state of the system and predicting the safety of future states

Consider the use of the integration paradigm to protect software systems with x86 architecture, and then move on to the secure architecture of computer equipment. Inherent in today's computer equipment is hierarchical control at the hardware level. For x86-based systems this hierarchy creates a hierarchy of software components. However, not all modes are in a hierarchical relationship. Some are on the same level. For example, a kernel mode may contain several sub-modes depending on the size of the code being executed or the data being used, which does not create a hierarchy of control. All of these modes have the same privileges. Nevertheless, three groups of hardware protection technologies do produce a hierarchy of control:

- Privileged mode support and address space control
- Hardware support for virtualization
- Hardware support of a separated trusted environment

Based on these technologies, using integration paradigm for software into a hierarchy system like that, the experts decided that implementing a small compact and very simple but effective hypervisor, placed on the lowest level, enables the system to provide both control and safety to all systems embedded within it (Figure 5).

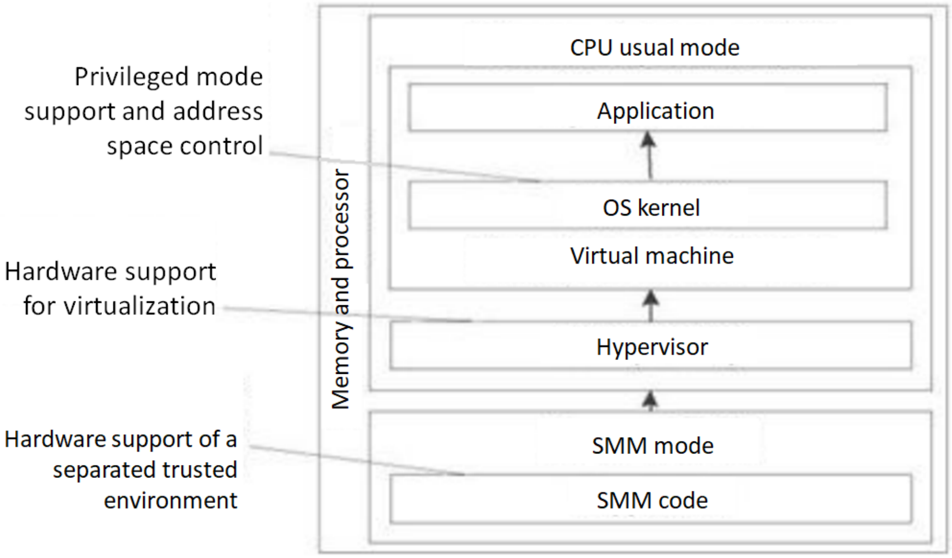
Privileged mode support and address space control
Hardware support for virtualization
Hardware support of a separated trusted environment
Memory and processor
CPU usual mode
Application
OS kernel
Virtual machine
Hypervisor
SMM mode
SMM code

Figure 5: Software hierarchy in x86 systems

This architecture is also called «thin» hypervisor or single virtual machine hypervisor. This hypervisor does not provide scheduling of the virtual machines – it just has one virtual machine and it is always running. Instead, it controls the behavior of all software components in the virtual machine as well as all the communication and information flow between the software components in virtual machine and the real hardware. The proposed approach has been successfully implemented in the form of a prototype. The first versions of the developed prototype were simply a hypervisor and operating system, whereas in newer editions there is a more complex design by adding support for SMM mode.

However, the meaning has not changed. It is the matryoshka doll principle, where some components are put inside the other components. What is on the outside is in complete control and controls the components on the inside. Through this architecture protection system prevents intruders from solving the problem of cyber-attack and taking over control.

The main advantages of using the integration paradigm on the example of the hypervisor are (Figure 6):

- Security management is centralized and concentrated in an isolated component
- The ability to control the exchange of the virtualized system with the external environment in order to eliminate hidden information leakage channels
- The behavior of the virtualized system is identical to that of the real system and the software operates without modification
- The ability to simulate the external environment for the virtualized system
- Hypervisor code size is negligible compared to the OS and applications
- Exploitation of vulnerabilities will not compromise the isolated defense
- Failures in the virtualized system can be quickly neutralized by rollback or reset

According to experts, the same attitude applies to the hardware approach. With ME technology, Intel does, in a way, the same step by moving the control functions to an external chipset.

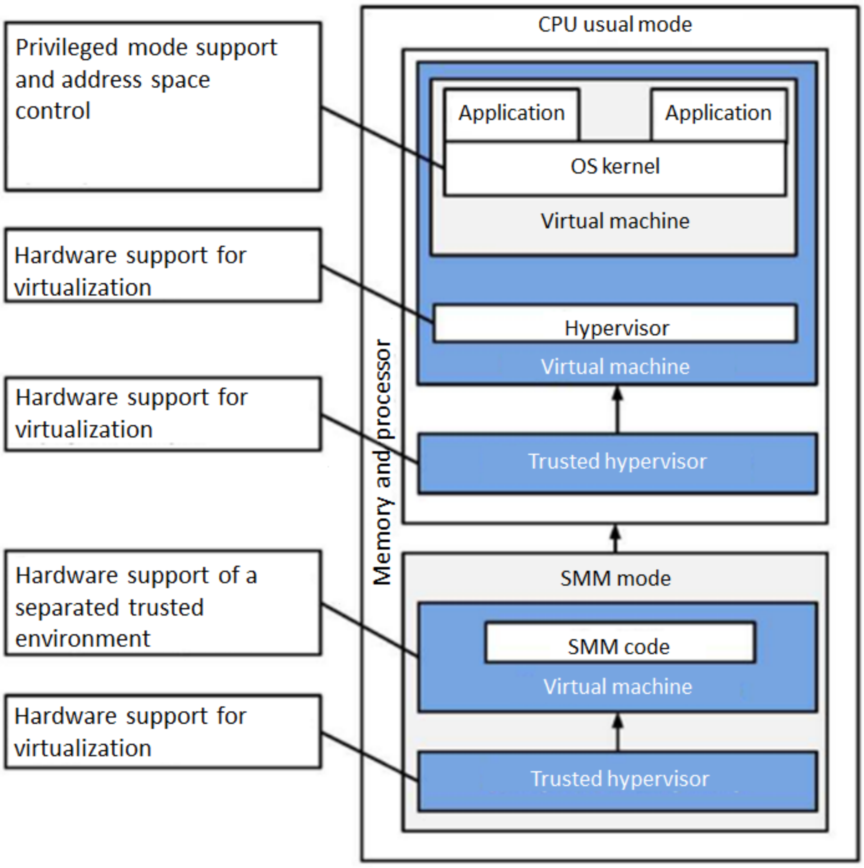
CPU usual mode
Privileged mode support and address space control
Application
Application
OS kernel
Virtual machine
Hardware support for virtualization
Hypervisor
Virtual machine
Hardware support for virtualization
Trusted hypervisor
Memory and processor
Hardware support of a separated trusted environment
SMM mode
SMM code
Virtual machine
Hardware support for virtualization
Trusted hypervisor

Figure 6: Using the hypervisor as a means of implementing the integration paradigm

An integration paradigm can be proposed to solve the problem of building of a secure computer equipment tool, as it is shown in Figure 7.

In essence, it is proposed to design and build another chip to control the Intel ME (chipset) controlling the processor, or to learn to control the processor's behavior with some of their own chipsets. In this way, it will be the same matryoshka, which will ensure security by the fact that the outermost layer (the component that controls everything else) is trusted. Experts believe that this would be the best way to meet the challenge of achieving technological independence and digital sovereignty. The main features of the proposed solution are:

- Consistent implementation of trusted computer equipment by integration of overseas and domestic components
- Control of all information flows and interactions by domestic trusted features at all stages of integration
- Use of the matryoshka principle – control of embedded components and compatibility with external technologies, processors and chipsets
- A universal mechanism of confirming trust by means of domestic cryptography
- Full preservation of the functionality of the monitored systems

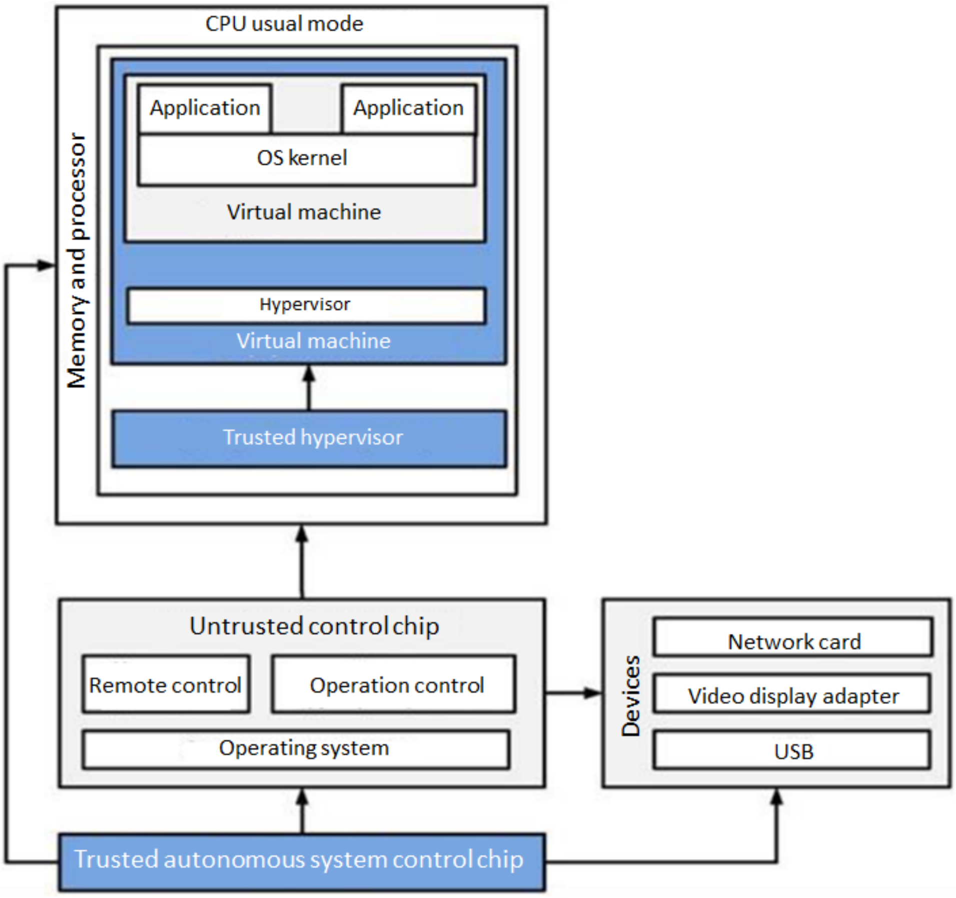

Figure 7: Using a trusted autonomous control and management chip as a means of implementing the integration paradigm

5. Conclusion

When selecting a technical and/or software platform, addressing the challenges of objectives interoperability with already existing backlog of developed and applied stock of products, each agency is planning its development forward-looking. In doing so, it is advisable to use the most comprehensive set of certified solutions.

In the context of acute information confrontation, domestic developments should not be reduced solely to the replication of foreign solutions.

6. References

[1] A. S. Tanenbaum, T. Austin. Structured computer organization. 6th ed., Piter, Saint Petersburg, 2019.

[2] Intel® 64 and IA-32 Architecture Software Developer’s Manual, volume 1: Basic Architecture. URL: https://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-softwaredeveloper-vol-1-manual.html.

[3] G. Rechistov, Ten names for a single architecture, 2013. URL: https://habrahabr.ru/company/intel/blog/201462/.

[4] M. J. Murdocca, V. P. Heuring. Computer architecture and organization: an integrated approach, 1st ed., Wiley, New York City, 2007.

[5] Fowler M. Architecture of corporate software applications, Williams Publishing House, London, 2006.

[6] J. Rushby, Design and Verification of Secure Systems, in: 8th ACM Symposium on Operating System Principles, Pacific Grove, California, 1981. pp. 12–21. doi: https://doi.org/10.1145/800216.806586.

[7] Lavrova D., Zegzhda D., Zaitceva E. Simulation of Complex Objects Network Infrastructure to Solve the Problem of Counteraction to Cyber Attacks. Voprosy kiberbezopasnosti. 2019, No 2(30), pp. 13-20. DOI: 10.21681/2311-3456-2019-2-13-20.

[8] A. Voica, New OmniShield platform implements multi-domain security for connected devices. URL: https://www.imgtec.com/blog/omnishield-multi-domain-security-connected-devices/.

[9] Intel® Virtualization Technology for Directed I/O, Architecture Specification, 2016. URL: https://software.intel.com/sites/default/files/managed/c5/15/vt-directed-io-spec.pdf.

[10] AMD I/O Virtualization Technology (IOMMU) Specification. URL: https://www.amd.com/system/files/TechDocs/48882_IOMMU_3.05_PUB.pdf.

[11] Intel® Trusted Execution Technology: Software Development Guide. URL: https://cs.technion.ac.il/~cs236376/readings/intel-txt-software-development-guide.pdf.

[12] D.A. Dudarev, A. Yu. Kravtsov, V. M. Poletaev, A.V. Poltavtsev, Ju.V. Romanets, V. K. Syrchin Device to create trusted execution environment for special-purpose computers, Patent RU 2569577 C1. Application RU 2014132337/08, Filled July, 6th, 2014, Issued November, 27th, 2015.

[13] ARM Security Technology, Building a Secure System using TrustZone® Technology. URL: https://static.docs.arm.com/genc009492/c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf.

[14] Intel® Virtualization Technology Specification for the IA-32 Intel® Architecture. URL: http://andrewl.dreamhosters.com/library/docs_intel/virtualization_Apr05.pdf.

[15] G. Lettieri, Intel VMX technology, 2015 URL: http://lettieri.iet.unipi.it/virtualization/2016/vn05.pdf.

[16] AMD Secure Virtual Machine Architecture Reference Manual, 2005. URL: https://www.mimuw.edu.pl/~vincent/lecture6/sources/amd-pacifica-specification.pdf.

[17] Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 3B: System Programming Guide, Part 2. URL: https://software.intel.com/content/www/us/en/develop/articles/intel-sdm.html.

[18] Markov A.S., Sheremet I.A. Enhancement of Confidence in Software in the Context of International Security. CEUR Workshop Proceedings, 2019, V. 2603, pp. 88-92.

[19] A.V. Barabanov, A.S. Markov, V.L. Tsirlov, Statistics of Software Vulnerability Detection in Certification Testing, in: Journal of Physics: Conference Series, 2018, Vol. 1015, p. 042033. DOI :10.1088/1742- 6596/1015/4/042033.

[20] Probabilistic Modeling in System Engineering / By ed. A. Kostogryzov – London: IntechOpen, 2018. 278 p. DOI: 10.5772/intechopen.71396.

[21] Barabanov A., Markov A. Modern Trends in the Regulatory Framework of the Information Security Compliance Assessment in Russia Based on Common Criteria. In Proceedings of the 8th International Conference on Security of Information and Networks (Sochi, Russian Federation, September 08-10, 2015). SIN '15. ACM New York, NY, USA, 2015, pp. 30-33. DOI: 10.1145/2799979.2799980.