Денис О. Стасьев
Московский физико-технический институт
(национальный исследовательский университет),
Институтский пер., 9, Долгопрудный, Московская область, 141701, Россия
e-mail: stasev.do@phystech.edu, https://orcid.org/0000-0002-6972-4635

КОНТРОЛЬ ЦЕЛОСТНОСТИ КОМПОНЕНТОВ ВИРТУАЛЬНЫХ МАШИН, СОЗДАННЫХ НА БАЗЕ ГИПЕРВИЗОРА KVM

DOI: http://dx.doi.org /10.26583/bit.2020.2.09

Аннотация. Сегодня виртуализация широко используется для предоставления масштабируемых ресурсов. Поскольку эта технология означает разделение ресурсов с помощью некоторого абстрактного слоя, то размещение и обработка информации различного уровня доступа в таких системах создаёт угрозу безопасности. Для решения этой проблемы необходимо создавать дополнительные системы контроля целостности виртуальной инфраструктуры. Цель работы: выделить компоненты ВМ, созданных на базе гипервизора KVM, для которых необходим контроль целостности, описать существующие способы обеспечения контроля целостности и выбрать наилучший для применения в централизованных системах. Методы исследования: метод правдоподобного рассуждения, системный анализ, формализация. В работе выделены компоненты виртуальных машин, созданных на базе гипервизора KVM, для которых необходим контроль целостности, описаны существующие способы обеспечения контроля целостности. Определен способ для применения в централизованных системах.

Ключевые слова: виртуализация, гипервизор, KVM, виртуальная машина, целостность, контроль целостности, компоненты виртуальных машин, резидентный компонент безопасности.
Для цитирования: СТАСЬЕВ, Денис О. КОНТРОЛЬ ЦЕЛОСТНОСТИ КОМПОНЕНТОВ ВИРТУАЛЬНЫХ МАШИН, СОЗДАННЫХ НА БАЗЕ ГИПЕРВИЗОРА KVM. Безопасность информационных технологий, [S.l.], v. 27, n. 2, p. 118-131, 2020. ISSN 2074-7136. Доступно на: <https://bit.mephi.ru/index.php/bit/article/view/1275>. Дата доступа: 08 june 2020. doi:http://dx.doi.org/10.26583/bit.2020.2.09.

Denis O. Stasyev
Moscow Institute of Physics and Technology (National Research University),
Institutsky Lane, 9, Dolgoprudny, Moscow region, 141701, Russia
e-mail: stasev.do@phystech.edu, https://orcid.org/0000-0002-6972-4635

Integrity monitoring of virtual machines components based on the KVM hypervisor

DOI: http://dx.doi.org /10.26583/bit.2020.2.09

Abstract. Today, virtualization is widely used to provide scalable resources. Since this technology means sharing resources using some abstract layer, the placement and processing of information of various access levels in such systems poses a security risk. To solve this problem, it is necessary to create additional systems for monitoring the integrity of the virtual infrastructure. Objective of the article: to isolate the components of VMs created on the basis of the KVM hypervisor for which integrity monitoring is necessary, describe existing methods for ensuring integrity monitoring and choose the best one for use in centralized systems. Research methods are plausible reasoning method, system analysis, formalization. The article identifies the components of virtual machines created on the basis of the KVM hypervisor for which integrity control is required, and existing methods for ensuring integrity control are described. The result of the article was the determination of the best method for use in centralized systems.

Keywords: virtualization, hypervisor, KVM, virtual machine, integrity, integrity control, components of virtual machines.
For citation: STASYEV, Denis O. Integrity monitoring of virtual machines components based on the KVM hypervisor. IT Security (Russia), [S.l.], v. 27, n. 2, p. 118-131, 2020. ISSN 2074-7136. Available at: <https://bit.mephi.ru/index.php/bit/article/view/1275>. Date accessed: 08 june 2020. doi:http://dx.doi.org/10.26583/bit.2020.2.09.

Введение

Виртуализация – это технология, позволяющая скрыть сложность аппаратного обеспечения с помощью создания абстрактного слоя системных ресурсов для программного обеспечения (ПО) рабочей среды [1]. Виртуализация позволяет использовать полную мощность физического компьютера, распределяя его возможности среди множества пользователей, поэтому часто выделяют ещё одну задачу виртуализации – разделение ресурсов.

Исследования в этой области начались более 50 лет назад. Первые технологии и разработки в области виртуализации были реализованы такими компаниями, как IBM (с 1967 года) и VMware (с 1998 года). В [2] сформулированы требования к виртуализации: эффективность, управление ресурсами (владение монитором) и эквивалентность (физической машине), которые остаются актуальными и в настоящий момент.

В настоящее время виртуализация является основой облачных вычислений. Данная технология широко используется для предоставления масштабируемых ресурсов. Например, разработчики различных проектов в промышленности используют виртуализацию с сервисами облачных вычислений для сокращения неактивных аппаратных ресурсов и достижения эффективной эксплуатации систем.

Виртуальная машина (ВМ, VM) – программная система, эмулирующая аппаратное обеспечение некоторой платформы и исполняющая программы для гостевой операционной системы (ОС) (target-платформа – целевая, заданная платформа) на хостовой ОС (host-платформа – платформа владельца, хозяйская) [3] с помощью технологии виртуализации.

Примером использования технологии виртуализации может быть широко известный сервис “инфраструктура как услуга” (IaaS), который означает предоставление экземпляров ВМ по требованию [4]. Сервис IaaS широко используется для обеспечения необходимых вычислительных ресурсов в средах с общими данными. Например, существуют сервисы Amazon Web Services (AWS), Google Compute Engine, Microsoft Azure, которые предлагают услуги IaaS [5].

Поскольку виртуализация означает разделение ресурсов с помощью некоторого абстрактного слоя системных ресурсов, то возникает ряд особенностей, которые не свойственны обычным хостовым системам. Это проблемы в отношении безопасности, изоляции и производительности гостевых ОС. При разработке систем виртуализации необходимо учитывать угрозы безопасности[1], связанные с нахождением информации различного уровня доступа на различных сегментах виртуальной инфраструктуры (ВИ) [6]. ВИ – это система, поддерживающая виртуализацию серверов (ВМ), сети и хранилищ данных.

В данной работе проводится анализ контроля целостности ВМ. Поскольку попытка обеспечения целостности всех компонентов ВМ может снизить производительность этих ВМ, то требуется отдельно рассмотреть целесообразность и необходимость применения контроля целостности для различных компонентов ВМ. Также следует определить существующие методы обеспечения контроля целостности. Поскольку в будущем планируется создание централизованной системы контроля целостности ВМ, то необходимо сравнить методы с целью дальнейшего применения в централизованных системах контроля целостности ВМ. В работе рассматриваются только ВМ, созданные на базе гипервизора KVM.

[1]Методический документ ФСТЭК России от 11.02.2014 «Меры защиты информации в государственных информационных системах».

Цель работы: выделить компоненты ВМ, созданных на базе гипервизора KVM, для которых необходим контроль целостности, описать существующие способы обеспечения контроля целостности и выбрать наилучший для применения в централизованных системах.

Основные задачи:

1. Изучить устройство и особенности ВМ, созданных на базе гипервизора KVM.
2. Проанализировать нормативную методическую базу в части защиты ВИ и контроля целостности их компонентов.
3. Определить компоненты ВМ, для которых необходим контроль целостности с точки зрения угроз безопасности и нормативной методической базы.
4. Изучить способы контроля целостности компонентов ВМ.
5. Определить наилучший способ контроля целостности компонентов ВМ для применения в централизованных системах.

Методы исследования: метод правдоподобного рассуждения, системный анализ, формализация.

Работа состоит из семи частей: описание серверной виртуализации; устройство гипервизора KVM; описание устройства ВМ, созданных на базе гипервизора KVM; анализ нормативной методической базы; общие сведения о контроле целостности ВИ; описание существующих моделей контроля целостности; способы контроля целостности компонентов ВМ и выбор наилучшего для применения в централизованных системах.

1. Виртуализация серверов

Понятие виртуализации в сфере информационных технологий (ИТ) существует давно, но его сущность с развитием ИТ несколько изменилась. В настоящий момент виртуализация применяется во многих областях ИТ, например, выделяют виртуализацию сети, виртуализацию хранения данных, виртуализацию серверов [7]. Рассмотрим подробнее последнюю, так как создание экземпляров ВМ относится к ней.

Виртуализация серверов (серверная виртуализация) – это маскировка ресурсов сервера от пользователей сервера. Основная цель серверной виртуализации – избавить администратора системы от необходимости понимать и управлять сложной архитектурой серверных ресурсов [7].

Для виртуализации серверов применяется несколько подходов, которые по типу реализации подразделяются на программные и аппаратные [8]. Примерами аппаратных являются технологии Intel VT (VT-x, Intel Virtualization Technology for x86) и AMD-V, их реализуют производители аппаратного обеспечения. Использование таких средств позволяет повысить производительность работы ВИ.

Основной частью программного подхода к виртуализации является гипервизор – основной компонент, обеспечивающий связь между оборудованием и ВМ [5]. Программная виртуализация осуществляется с помощью этого уровня абстракции. Гипервизоры основаны на реализации определённого программного подхода (стратегии), которые делятся на три основные категории: полная виртуализация (full virtualization), паравиртуализация (para-virtualization), виртуализация уровня ОС (OS-level virtualization) [5].

Полная виртуализация – обеспечивает виртуализацию без изменения гостевой (запускаемой) ОС, то есть виртуализация привилегированных инструкций может быть выполнена без поддержки аппаратной или ОС [5]. Паравиртуализация отличается от полной виртуализации тем, что требует изменений в ядре гостевой ОС [5]. Виртуализация уровня ОС (контейнеризация) позволяет запускать в рамках одной ОС на одном ядре несколько ВМ в изолированных разделах. Издержки в этой модели очень ограничены из-за

преимуществ работы в ОС с общим ядром. Гостевая ОС и хостовая должны иметь одну и ту же ОС или ядро [5].

При создании систем виртуализации серверов обычно используется аппаратная виртуализация совместно с гипервизорами, основанными на полной виртуализации. Наличие аппаратной виртуализации обеспечивает высокую производительность, а поддержка полной виртуализации обеспечивает удобство использования, то есть отсутствует необходимость внесения изменений в гостевые ОС.

Гипервизор иначе называют монитором виртуальной машины (MBM, virtual machine monitor, VMM). Эти два термина (гипервизор и MBM) обычно рассматриваются как синонимы, однако имеют существенное отличие. MBM представляет собой программное обеспечение, которое управляет центральным процессором (ЦП), памятью, передачей данных ввода-вывода, прерыванием и набором команд в данной виртуальной среде. Гипервизор может быть частью хостовой ОС со встроенной MBM [5], но это не всегда так.

Как правило, гипервизоры можно разделить на гипервизоры первого и второго типа в зависимости от уровня реализации [4]. Гипервизор первого типа работает непосредственно на физическом оборудовании, то есть связь между оборудованием и ВМ, запускаемой гипервизором, является прямой. ОС хоста не требуется в гипервизоре первого типа, поскольку он работает непосредственно на физическом обеспечении компьютера. По этой причине его иногда называют “аппаратным гипервизором” (“hardware hypervisor”). VMware vSphere/ESXi, Microsoft Windows Server 2012 Hyper-V, Citrix XenServer, Red Hat Enterprise Virtualization (RHEV) и система с открытым исходным кодом Kernel-based Virtual Machine (KVM) относятся к этой категории. Гипервизор второго типа находится в ОС [4], позволяя управлять ВМ с поддержкой конфигурации оборудования из ОС. Дополнительный уровень между оборудованием и ВМ в гипервизоре второго типа снижает эффективность по сравнению с гипервизором первого типа. VirtualBox и VMware Workstation можно отнести к этой категории [5].

2. Гипервизор KVM

KVM – полностью открытое программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86, которая поддерживает аппаратную виртуализацию на базе Intel VT либо AMD SVM (Secure VM).

ПО KVM состоит из нескольких компонентов. Основным является модуль ядра. Он состоит из загружаемого основного модуля kvm.ko, предоставляющего базовый сервис виртуализации, и процессорно-специфического загружаемого модуля kvm-amd.ko либо kvm-intel.ko. Для эмуляции аппаратного обеспечения (например, для запуска ОС, предназначенных под одну архитектуру, на другой или для эмуляции устройств ввода-вывода) гипервизор KVM использует программу QEMU [9]. Для управления работой экземпляров ВМ, созданных KVM, используется программа virt-manager [4], предоставляющая доступ из консоли и графический интерфейс (GUI/CLI). Ещё одним из наиболее распространённых методов управления гипервизором KVM, который частично повторяет функционал virt-manager, является доступ с помощью набора свободных инструментов libvirt. Libvirt – это свободная реализация программного интерфейса гипервизора KVM (API), сервис libvirtd (программа, работающая UNIX-подобных ОС в фоновом режиме) (демон) и набор инструментов для управления виртуализацией virsh [10,11], который особенно популярен для управления KVM в режиме командной строки. Ешё одной частью KVM является ядро Linux. Поскольку QEMU работает как обычный процесс [12], планирование соответствующей гостевой ОС выполняется самим ядром Linux.

3. Устройство ВМ, созданных на базе гипервизора KVM

ВМ – это комплексная система, состоящая из различных компонентов. Чтобы понимать принципы работы ВМ, различия между реализациями (типами) таких систем, необходимо определить основные интерфейсы и компоненты (уровни) архитектуры компьютерной системы (Computer system architecture). Интерфейсы связывают различные компоненты, обеспечивая их совместную работу в рамках единой архитектуры компьютерной системы. Определим основные компоненты (уровни) архитектуры компьютерной системы согласно основной иерархии обращений.

На самом низком программным уровне работает исполнитель оборудования (Execution hardware) [13]. Обычно он скрывает преобразование памяти (Memory translation), системную шину (System interconnect (bus)), устройства ввода-вывода и сети (I/O devices and networking), основную память (Main memory) и обеспечивает работоспособность аппаратного обеспечения.

Следующим уровнем архитектуры компьютерной системы является ОС [13], которая взаимодействует с исполнителем оборудования с помощью интерфейса ISA. ISA (Instruction set architecture) – архитектура набора команд, необходимая для запуска ПО на аппаратной платформе, позволяет ОС управлять физическими ресурсами системы.

Выше уровня ОС располагаются библиотеки [13]. Они могут обращаться как к исполнителю оборудования, так и к ОС. Интерфейс ABI (Application binary interface) реализует библиотекам доступ к аппаратным ресурсам (через пользовательскую ISA) и к системным вызовам ОС.

Самым старшим компонентом в иерархии архитектуры компьютерной системы является уровень прикладного программного обеспечения (ППО) [13]. Он взаимодействует с библиотеками через API (Application programming interface) с помощью высокоуровневых (high-level language, HLL) библиотечных вызовов. Компоненты, принадлежащие этому уровню, имеют право доступа к аппаратным ресурсам через пользовательскую ISA.

Существует два основных типа ВМ: ВМ процессов (Process Virtual Machine) и системные ВМ (System Virtual Machine) [13]. ВМ процессов предоставляют виртуальную среду ABI или API для пользовательских приложений. В различных реализациях ВМ процессов предлагают репликацию, эмуляцию и оптимизацию. Примерами таких ВМ могут быть Java Virtual Machine (JVM) и Microsoft Common Language Infrastructure, которая является основой .NET фреймворка (программная платформа, определяющая структуру программной системы, облегчает разработку и объединение различных компонентов программного продукта).

Системная ВМ предоставляет полную среду, в которой могут сосуществовать ОС и множество процессов, возможно принадлежащих нескольким пользователям. Используя системные ВМ, аппаратная платформа с одним хостом поддерживает одновременно несколько изолированных гостевых ОС. Эти возможности напрямую используются в серверной виртуализации, поэтому большинство “серверных” хостовых систем являются системными ВМ. Таким образом, сам гипервизор KVM является системной ВМ и его можно назвать ВМ, однако для разделения терминов будем называть экземпляры (инстансы), которые создаёт KVM (гостевые машины), виртуальными машинами (ВМ), а сам KVM – гипервизором.

При создании ВМ гипервизор создаёт набор определённых файлов на хосте. В отличие от обычной системы с установленной ОС, которая занимает выделенную область диска, гипервизор может динамически менять характеристики каждой отдельно взятой ВМ с помощью изменения конфигурационных файлов. Выделим основные файлы, которые

определяют ВМ. Полученное выделение установит основные компоненты ВМ, созданные на базе гипервизора KVM:

1) Файлы конфигурации описывают пользовательские и другие атрибуты ВМ [14]. Пользовательские атрибуты – это новые графы (поля), которые пользователь может добавить к ВМ и хостам, содержащие специфическую информацию, присущую конкретной организации. Например, если создать новый атрибут «Организация», то в списке ВМ кроме имени ВМ, её состояния, загрузки процессора и использования памяти будет доступна новая графа – «Организация» [15]. Удобство заключается в том, что поскольку информация о принадлежности ВМ к определенной организации является полем, то по нему может осуществляться сортировка. Также к этой категории относятся файлы, содержащие общую информацию о ВМ, то есть: определение сервера, сколько виртуальных процессоров (vCPU) выделено для этой ВМ, сколько оперативной памяти выделено, к каким устройствам ввода-вывода ВМ имеет доступ, сколько сетевых интерфейсных карт (NIC) находятся на виртуальном сервере и другие [14], а также файлы, определяющие параметры, такие как размер и другие, виртуального диска, к которому имеет доступ ВМ [14].

2) При включении или создании ВМ, создаются дополнительные файлы для ведения журнала учёта, подкачки памяти и других функций [14].

3) При копировании отдельных пользовательских файлов, находящихся внутри ВМ, создаётся не только резервная копия этих данных, но и копия всего сервера, включая ОС, приложения и саму конфигурацию оборудования [14]. Файлы, создающиеся при копировании данных, можно выделить в отдельный блок.

4) Файлы, содержащие ППО пользователей.

5) Файлы-образы гостевых ОС.

Таким образом, определено пять основных компонентов ВМ, созданных на базе гипервизора KVM. Компоненты типов (1), (5) существуют всегда, (2)–(4) могут отсутствовать (зависит от настроек системы и пользователя).

4. Анализ нормативной методической базы в части защиты ВИ

Согласно приказам ФСТЭК России[2] набора функций средств виртуализации для настройки ВИ с учетом требований безопасности[1] недостаточно, поэтому для их защиты от несанкционированных изменений необходимо использовать наложенные (внешние) средства защиты информации (СЗИ) [16,17].

Необходимо определить компоненты ВМ, для которых для которых требуется контроль целостности в соответствии с нормативной методической базой, среди определённых ранее компонентов ВМ, созданных на базе гипервизора KVM.

Определим рассматриваемую нормативную методическую базу. Согласно указанным выше приказам ФСТЭК России, основным требованием в области обеспечения контроля целостности ВИ является ЗСВ.7 (Защита среды виртуализации) («Контроль целостности виртуальной инфраструктуры и ее конфигураций»). Поскольку ВМ содержат

[2]Приказ № 17 ФСТЭК России от 11 февраля 2013 г. «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах».
Приказ № 21 ФСТЭК России от 18 февраля 2013 г. «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных».
Приказ № 31 ФСТЭК России от 14 марта 2014 г. «Об утверждении требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды».

файлы резервного копирования, то при разработке СЗИ следует учитывать ЗСВ.8 («Резервное копирование данных, резервирование технических средств, программного обеспечения виртуальной инфраструктуры, а также каналов связи внутри виртуальной инфраструктуры»). Поскольку ВИ широко используются в финансовых организациях, то включим в рассмотрение рекомендации в области защиты ВИ Банка России [18] для организаций банковской системы Российской Федерации.

Рассмотрим ЗСВ.7. Поскольку темой данной работы является обеспечение контроля целостности только компонентов ВМ, а не ВИ в целом, то будем выделять только аспекты контроля целостности ВМ. Начнём с ОЦЛ.1 (Обеспечение целостности информационной системы и информации), на которую ссылается ЗСВ.7:

1. В информационной системе (ИС) должен осуществляться контроль целостности ПО СЗИ, включая их обновления, по наличию имен (идентификаторов) и (или) по контрольным суммам всех компонентов СЗИ, как в процессе загрузки, так и динамически в процессе работы ИС с использованием криптографических методов в соответствии с законодательством Российской Федерации[1].

2. Должен быть реализован контроль целостности компонентов ПО (за исключением СЗИ), по наличию имен (идентификаторов) компонентов ПО и (или) по контрольным суммам, как в процессе загрузки, так и динамически в процессе работы ИС с использованием криптографических методов в соответствии с законодательством Российской Федерации[1].

3. Необходимо проведение тестирования с периодичностью, установленной оператором, функций безопасности СЗИ, в том числе с помощью тест-программ, имитирующих попытки несанкционированного доступа, и (или) специальных программных средств, в соответствии с АНЗ.1 и АНЗ.2 (Анализ защищённости информации)[1].

4. В случае если функциональные возможности ИС должны предусматривать применение в составе её ПО средств разработки и отладки программ, оператором обеспечивается выполнение процедур контроля целостности ПО после завершения каждого процесса функционирования средств разработки и отладки программ[1].

5. В ИС должна обеспечиваться блокировка запуска ПО и (или) блокировка сегмента (компонента) ИС (автоматизированного рабочего места, сервера) в случае обнаружения фактов нарушения целостности[1].

Таким образом, согласно пунктам 1–2 необходим контроль целостности не только компонентов ВМ, но и самого СЗИ и его обновлений, как в процессе загрузки, так и в процессе работы ИС. СЗИ должно тестироваться с определённой периодичностью. Для ВМ с ПО средств разработки и отладки программ необходимо выполнять контроль целостности ПО после завершения каждого процесса функционирования средств разработки и отладки программ. В ИС должна обеспечиваться блокировка запуска ПО в случае обнаружения фактов нарушения целостности.

В ЗСВ.7 выделяются аспекты применения СЗИ в ВИ:

1. В ИС должен осуществляться контроль целостности компонентов, критически важных для функционирования хостовой ОС, гипервизора, гостевых ОС и (или) обеспечения безопасности, обрабатываемой в них информации (загрузчика, системных файлов, библиотек ОС и иных компонентов).

2. Должен осуществляться контроль целостности состава и конфигурации виртуального оборудования.

3. Должен осуществляться контроль целостности файлов, содержащих параметры настройки виртуализированного ПО и ВМ.

4. Должен осуществляться контроль целостности файлов-образов виртуализированного ПО и ВМ, файлов-образов, используемых для обеспечения работы виртуальных файловых систем (контроль файлов-образов должен проводиться во время, когда файлы-образы не задействованы).

5. В ИС должен обеспечиваться контроль целостности резервных копий ВМ (контейнеров).

6. В ИС должен обеспечиваться контроль целостности базовой системы ввода-вывода вычислительных серверов и консолей управления виртуальной инфраструктуры.

7. В ИС должен обеспечиваться контроль целостности программного обеспечения облачных клиентов.

Получаем, что согласно пунктам 1–3 необходимо обеспечивать контроль компонентов типов (1), (3) предыдущего раздела. Пункт 4 соответствует (2), (5), а пункт 7 обязывает обеспечивать контроль файлов типа (4).

По рекомендациям Банка России[3] при создании базовых образов ВМ рекомендуется проводить процедуры, необходимые для выполнения последующего контроля их целостности. Созданный или измененный базовый образ ВМ перед размещением на основном оборудовании, реализующем технологию виртуализации, рекомендуется проверять в тестовом сегменте на соответствие настроек включенных в образ программных компонентов СЗИ требованиям, установленным соответствующей эксплуатационной документацией. Для каждого базового образа ВМ рекомендуется выполнять регламентированные процедуры контроля: соответствия настроек, включенных в образ программных компонентов СЗИ, требованиям, установленным эксплуатационной документацией; целостности ПО, включенного в образ ВМ. Рекомендуется выполнять регламентированные процедуры контроля целостности ПО, выполняемые при загрузке указанного ПО.

Поэтому, в соответствии с рекомендациями Банка России, необходимо обеспечивать контроль целостности ПО ВМ, в том числе выполняемый на этапе загрузки ВМ. Необходимо обеспечивать контроль файлов (1), (2) и (5), участвующих в загрузке ВМ.

Таким образом, несмотря на то, что существуют способы обеспечения контроля целостности файлов пользовательского ППО, содержимого памяти и контекстов центрального процессора в условиях полностью скомпрометированной ОС [4], проведённый анализ нормативной методической базы в части защиты ВИ показал необходимость обеспечения контроля целостности всех файлов ВМ. В частности, необходим контроль целостности конфигурационных файлов, ОС и ПО, установленного внутри экземпляров пользовательских гостевых ВМ.

5. Контроль целостности в серверной виртуализации

Целостность определяется как свойство безопасности информации, при котором отсутствует её несанкционированное изменение (изменение субъектами доступа, не имеющими на него право)[1]. Однако при таком определении целостности информация рассматривается исключительно как неделимый объект в том смысле, что его нельзя разбить на контролируемые и неконтролируемые части [6]. Поэтому под целостностью вычислительной среды (в контексте виртуализации) понимают стабильность в течение рассматриваемого периода в требуемом диапазоне состава объектов и процессов, их взаимосвязей и параметров функционирования [6,19]. Повторим, что контроль целостности

[3]Рекомендации в области стандартизации Банка России – Обеспечение информационной безопасности организаций банковской системы Российской Федерации – Обеспечение информационной безопасности при использовании технологии виртуализации. URL: https://www.cbr.ru/Content/Document/File/46925/rs-28-15.pdf

предполагает сравнение текущего состояния некоторого объекта с набором эталонов, то есть с выделенным состоянием объекта, которое считается корректным и безопасным [20]. Состав, параметры и взаимосвязи и в течение какого периода времени, определяется для каждого конкретного случая отдельно.

Всё множество функций контроля целостности можно сгруппировать в три класса: контроль целостности технических средств ЭВМ, контроль целостности системных областей жестких дисков, контроль целостности отдельных файлов и программных средств. Последний класс отвечает за осуществление контроля целостности ОС и необходимых программных компонентов, в том числе и иных средств защиты и элементов комплексов защиты, работающих после средства доверенного загрузки (СДЗ) в ДВС [19].

Таким образом, для организации тонкой настройки контроля целостности параметров вычислительной среды и повышения эффективности работы процессов, реализующих меры защиты, необходимо обеспечить возможность контроля отдельных элементов: файлов и каталогов, входящих в состав архива и влияющих на загрузку ОС [19].

6. Модели контроля целостности компонентов ВМ

ВИ – динамическая система, некоторые её связи, атрибуты объектов могут быть изменены и при этом состояние системы останется корректным (не нарушающим целостность). Например, для ВМ может быть определён набор хостов-гипервизоров, перемещение между которыми (миграция) разрешено. Если рассматривать эталонную конфигурацию ВИ в традиционном понимании, то есть как некоторый «снимок» конкретного состояния системы, то возникает проблема, связанная с тем, что выбор такого эталона зачастую невозможен – например, одним «снимком» нельзя охватить сразу несколько разрешённых состояний. Необходимо задавать такое представление эталона и текущей конфигурации, такое их сопоставление, которое позволит учитывать множество разрешённых состояний [20]. Поэтому исторически целостность среды оценивалась с использованием моделей целостности [21].

Первые всесторонние модели целостности были предложены в [21], в которых субъектам и объектам назначаются метки целостности на основе их начального состояния целостности, и эти метки располагаются в решётке целостности, где информация может передаваться только от объектов с более высокой целостностью к объектам с более низкой целостностью. Например, субъект может только читать объекты выше (или равные) в решетке целостности и записывать объекты ниже (или равные) в решетке целостности. Должно быть обеспечено начальное состояние целостности системы, которое предоставляет модель целостности Кларка-Вилсона с помощью явного определения процедуры проверки целостности, процесса проверки целостности системы во время инициализации, чтобы гарантировать высокую целостность начальной (отправной) точки [21].

Основная проблема в проверке целостности среды выполнения состоит в том, что системы с высокой степенью целостности могут получать ненадежные входные данные (например, из сети) [21]. Модель целостности Кларка-Вилсона определила эту проблему, в которой программы высокой целостности (называемые процедурами преобразования в модели Кларка-Уилсона) должны быть способны немедленно отбрасывать или обновлять ненадежные входные данные [21]. Однако необходима формальная уверенность в правильности (высокой целостности) этих программ для обоснования такого поведения в модели Кларка-Вилсона. Несмотря на то, что формальная гарантия для программ стала жизнеспособной технологией в 1987 году (то есть, когда была предложена модель Кларка-Вилсона) [21], в настоящее время широко распространено мнение, что формальная гарантия полной правильности нецелесообразна, поэтому в последнее время появились новые идеи

по обеспечению контроля целостности. Они требуют, чтобы программы с высокой степенью целостности были разработаны так, чтобы могли принимать ненадежные входные данные только на интерфейсах, где могут быть приняты решения о целостности.

Одной из реализаций этих идей является атрибутная модель контроля доступа [20]. Важной частью модели являются политики безопасности. Политики безопасности – это совокупность свойств, определяющих эталон ВИ. Обычно их устанавливает администратор безопасности в соответствии с решаемыми задачами. Эти политики передаются на общий сервер проверки. По запросу модуль, работающий на автоматизированном рабочем месте администратора, получает из ВИ данные, определяющие её текущее состояние, и формирует на их основе запросы. После передачи этих запросов на сервер проверки, начинается их оценка на соответствие созданным ранее политикам. Если проверка всех запросов на соответствие политикам прошла успешно, то целостность конфигурации ВИ сохранена. В противном случае имеют место нарушения целостности.

Атрибутная модель контроля доступа может применяться как при решении задачи разграничения доступа, так и при решении задачи обеспечения контроля целостности конфигурации ВИ [20]. Созданы способы упрощения процесса написания политик безопасности для администраторов. Например, использование готовых шаблонов.

При применении атрибутной модели контроля доступа решается основная проблема контроля целостности конфигурации – противоречие с традиционным подходом, при котором разрешённым может быть лишь одно состояние, совпадающее с эталоном [20].

7. Способы контроля целостности компонентов ВМ

В зависимости от расположения СЗИ выделяют несколько способов контроля целостности компонентов ВМ. Рассмотрим основные из них, которые можно использовать с гипервизором KVM. Поскольку KVM является гипервизором первого типа, то СЗИ может либо встраиваться между самим гипервизором и экземплярами ВМ, либо может существовать отдельно, то есть находиться снаружи от системы “гипервизор и экземпляры ВМ”. Выделим основные способы контроля целостности компонентов ВМ, созданных на базе гипервизора KVM:

1) СЗИ может встраиваться между гипервизором и экземплярами ВМ. Это можно реализовать с помощью создания СЗИ, которое будет перехватывать обращения ВМ к гипервизору, обрабатывать эти запросы и пересылать их гипервизору.

2) СЗИ может быть встроенным в гипервизор. Этот способ является развитием первого способа, может быть реализован с помощью изменения исходного кода базового гипервизора, что не всегда допустимо. Также этот способ не позволит полностью реализовать контроль целостности самого гипервизора (поскольку является его частью), которое требуется согласно нормативным методическим актам.

3) СЗИ может быть встроенным в БИОС ВМ. Такой способ может применяться при разработке децентрализованных систем.

4) СЗИ может быть находиться снаружи от системы “гипервизор и экземпляры ВМ”.

Рассмотрение последнего способа невозможно без понятий доверенной вычислительной среды (ДВС) и резидентного компонента безопасности (РКБ). Фрагмент среды электронного взаимодействия, для которого установлена и поддерживается в течение заданного интервала времени целостность объектов и целостность взаимосвязей между ними, называется ДВС [18]. В настоящее время модель ДВС остаётся одной из актуальных и практически применимых субъектно-объектных моделей защиты технологии электронного обмена информации [19]. Появляются предложения по новым реализациям и функциональному составу РКБ, наличие которого требуется для построения ДВС [18,19],

однако контроль целостности элементов среды остаётся одной из основ создания ДВС. Следовательно, одна из основных групп функций, которые должны быть реализованы РКБ, – это функции контроля целостности: контроль целостности технического состава ЭВМ и локальной вычислительной сети (ЛВС), контроль целостности ОС, контроль целостности ППО и данных.

РКБ не может быть встроенным СЗИ, так как в таком случае не может быть полноценно реализована концепция РКБ системы как активного элемента, независящего от защищаемой системы и реализующего заданный набор процедур её контроля, поэтому СЗИ должно быть наложенным. Данная концепция (независящего от защищаемой системы элемента) положена в основу всех программно-аппаратных комплексов компании «ОКБ САПР» для защиты различных инфраструктур виртуализации: «Аккорд-В» для VMware vSphere; «ГиперАккорд» для Microsoft Hyper-V; «Аккорд-KVM» для KVM, – каждый из которых является наложенным СЗИ [17].

Централизованная система контроля целостности компонентов ВМ, созданных на базе гипервизора KVM, должна быть наложенным СЗИ. Это СЗИ должно находиться снаружи от системы “гипервизор и экземпляры ВМ”, то есть должно обладать независимым подключением (прямым соединением) как к ВМ, так и к гипервизору KVM [22] для обеспечения контроля целостности ВМ и гипервизора согласно ЗСВ.7, то есть должно являться РКБ.

Примером успешной реализации подобной схемы с независимыми подключениями является система VIS (Virtualization Introspection System) для обнаружения различных атак [23]. Она обладает независимым подключением к экземплярам ВМ и к гипервизору KVM, поэтому может обнаруживать атаки не только на ВМ, но и на гипервизор [24]. Использование подобной схемы избавляет систему VIS от необходимости постоянно «доверять» гипервизору.

Применение контроля целостности к гипервизору обусловлено не только требованиями нормативной методической базы, но и существованием ряда атак на гипервизор [24], поэтому отказаться от применения к гипервизору контроля целостности, рассматривая задачу контроля целостности компонентов ВМ, нельзя. Полученная система наложенного СЗИ может реализовывать монитор безопасности объектов (МБО) [25].

При рассмотрении способов обеспечения контроля целостности компонентов ВМ следует помнить о том, что возможно разделение ВМ по зонам доверия [26] Возможна и допустима ситуация, когда на одном физическом сервере находятся, например, информация различного уровня доступа. Разработчикам средств контроля целостности компонентов ВИ следует предоставлять администраторам возможность написания соответствующих политик безопасности. Комплексные средства защиты платформ виртуализации таких производителей, как Trend Micro, Reflex Systems, позволяют изолировать машины из разных зон доверия, а также создавать профили и политики безопасности, автоматизирующие применение таких настроек [26]. Более того, при перемещении машины на другой сервер такой профиль может предотвратить ошибочное подключение внутренней системы к внешней сети.

Стоит отметить, что ИС, использующие виртуализацию на базе KVM, могут значительно различаться, но в любом случае необходимой мерой их защиты является обеспечение контроля целостности ВМ [27]. Средствам защиты для таких систем необходимо, но не достаточно решать эту задачу, кроме того, они должны соответствовать всем факторам, характеризующим защищаемую виртуальную инфраструктуру: ОС гипервизоров, способы подключения хранилища, гостевые версии ОС, системы управления и т.д.

Заключение

В работе проведен анализ средств виртуализация серверов, устройство гипервизора KVM и его компоненты: QEMU, virt-manager, libvirt и др.; выделены основные компоненты ВМ: файлы конфигурации, файлы ведения журнала учёта (и подкачки памяти), файлы копирования, файлы, содержащие ППО пользователей, и файлы-образы гостевых ОС.

Проведён анализ нормативной методической базы в части защиты ВИ с целью определения компонентов ВМ, для которых необходим контроль целостности. Определены особенности обеспечения контроля целостности в серверной виртуализации. Описывается возможность использования атрибутной модели для решения задачи обеспечения контроля целостности компонентов ВМ, созданных на базе гипервизора KVM.

Рассмотрены основные способы контроля целостности ВМ, созданных на базе гипервизора KVM. Определяется, что централизованная система контроля целостности компонентов ВМ, созданных на базе гипервизора KVM, должна быть наложенным СЗИ, которое должно находиться снаружи от системы “гипервизор и экземпляры ВМ”, то есть должно независимо подключаться как к ВМ, так и к гипервизору KVM для обеспечения контроля целостности ВМ и гипервизора.

Полученные в данной работе результаты могут быть использованы при разработке централизованной системы контроля целостности компонентов ВМ, созданных на базе гипервизора KVM.

СПИСОК ЛИТЕРАТУРЫ:

1. Durairaj M., Kannan P. A Study On Virtualization Techniques And Challenges In Cloud Computing. International Journal of Scientific & Technology Research. 2014. Vol. 3. P. 147–151. URL: https://www.ijstr.org/final-print/nov2014/A-Study-On-Virtualization-Techniques-And-Challenges-In-Cloud-Computing.pdf (дата обращения: 15.04.2020).
2. Popek G. J., Goldberg R. P. Formal Requirements for Virtualizable Third Generation Architectures. Communications of the ACM. 1974. Vol. 17. P. 412–421. DOI: https://doi.org/10.1145/361011.361073.
3. Дорошенко В. С., Шадрин Д. Б. Использование виртуальных машин в обучении. Новые образовательные технологии в вузе: материалы XII международной научно-методической конференции (НОТВ-2015). 2015. С. 106–108. URL: https://elibrary.ru/item.asp?id=29903669 (дата обращения: 15.04.2020).
4. YamunaDevi, L. & P, Aruna & Dorairaj, Sudha Devi & Priya, Navya. (2011). Security in Virtual Machine Live Migration for KVM. 10.1109/PACC.2011.5979008. DOI: 10.1109/PACC.2011.5979008.
5. Hyungro L. Virtualization Basics: Understanding Techniques and Fundamentals. School of Informatics and Computing, Indiana University. 2014. P. 1–5. URL: http://dsc.soic.indiana.edu/publications/virtualization.pdf (дата обращения: 21.04.2020).
6. Мозолина Н. В. Контроль целостности виртуальной инфраструктуры и её конфигурации. Вопросы защиты информации. 2016. Вып. 3. С. 31–33. URL: https://elibrary.ru/item.asp?id=26992575 (дата обращения: 21.04.2020).
7. Гордиевских В.М. Сущность, структура и классификация современных технологий виртуализации. В лабораторию учёного. 2015. С. 125–133. URL: https://elibrary.ru/item.asp?id=23400691 (дата обращения: 21.04.2020).
8. Елманова Н., Пахомов. С. Виртуальные машины 2007. КомпьютерПресс. 2007. Вып. 9. С. 29–42. URL: https://compress.ru/article.aspx?id=18046 (дата обращения: 21.04.2020).
9. Сайт проекта KVM. Главная страница. URL: https://www.linux-kvm.org/page/Main_Page (дата обращения: 18.10.2019).
10. Сайт проекта KVM. Средства управления. URL: https://www.linux-kvm.org/page/Management_Tools (дата обращения: 19.10.2019).
11. Сайт проекта Libvirt. Часто задаваемые вопросы. URL: https://wiki.libvirt.org/page/FAQ#What_is_libvirt.3F (дата обращения: 22.10.2019).
12. Goto Y. Kernel-based Virtual Machine Technology. FUJITSU Sci. Tech. J. 2011. Vol. 47. P. 362–368. URL: https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol47-3/paper18.pdf (дата обращения: 21.04.2020).
13. J. E. Smith and R. Nair, «The Architecture of Virtual Machines», Computer (IEEE), Vol. 38, No. 5, 2005. P.32–38. DOI:10.1109/MC.2005.173.
URL: https://www.scirp.org/reference/referencespapers.aspx?referenceid=449371 (дата обращения: 21.04.2020).
14. Operating Systems: Internals and Design Principles. Chapter 14. Virtual Machines.
URL: https://www.unf.edu/public/cop4610/ree/Notes/PPT/PPT8E/CH14-OS8e.pdf (дата обращения: 18.11.2019).

15. Использование пользовательских атрибутов в Vcenter Server. URL: http://it-pilot.ru/2013/10/14/atribut/ (дата обращения: 18.11.2019).
16. Рябов А. С., Угаров Д. В., Постоев Д. А. Безопасность виртуальных инфраструктур. Сложности и нюансы выполнения требований регулятора. Комплексная защита информации: материалы XXI научно-практической конференции. 2016. С. 217–220.
URL: https://www.okbsapr.ru/library/publications/ryabov_2015_2/ (дата обращения: 21.04.2020).
17. Лыдин С.С. Средства защиты информации для инфраструктуры виртуализации: встроенные или наложенные? URL: http://www.okbsapr.ru/lydin_kzi2017.html (дата обращения: 01.12.2019).
18. Конявский В. А., Гадасин В. А. Основы понимания феномена электронного обмена информацией. Минск: Беллитфонд. 2004. С. 239–245. URL: https://www.okbsapr.ru/upload/iblock/016/osnovi_ponim_el_obmen_inf.pdf (дата обращения: 21.04.2020).
19. Алтухов А. А. Доверенная загрузка и контроль целостности архивированных данных. Часть и целое. Вопросы защиты информации. 2016. Вып. 2. С. 35–39.
URL: https://www.okbsapr.ru/library/publications/altukhov_2016_1/ (дата обращения: 21.04.2020).
20. Мозолина Н. В. Решение задачи контроля целостности конфигурации, основанное на атрибутной модели контроля доступа // Вопросы защиты информации. 2017. Вып. 3. С. 23–25.
URL: https://www.okbsapr.ru/library/publications/mozolina_2017_2/ (дата обращения: 21.04.2020).
21. Schiffman J., Moyer T., Shal C., Trent J., McDaniel P. Justifying Integrity Using a Virtual Machine // 25th Annual Computer Security Applications Conference (ACSAC), 2009.
URL: http://www.patrickmcdaniel.org/pubs/acsac09c.pdf (дата обращения: 01.12.2019).
22. Liu D. A Research on KVM-Based Virtualization Security. Applied Mechanics and Materials. 2014. Vol. 543-547. P. 3126–3129. DOI: https://doi.org/10.4028/www.scientific.net/AMM.543-547.3126 (дата обращения: 21.04.2020).
23. Lee S., Yu F. Securing KVM-Based Cloud Systems via Virtualization Introspection // 2014 47th Hawaii International Conference on System Sciences. 2014. P. 5028–5037. DOI: 10.1109/HICSS.2014.617.
URL: https://ieeexplore.ieee.org/document/6759220 (дата обращения: 21.04.2020).
24. Авезова Я. Э., Фадин А. А. Вопросы обеспечения доверенной загрузки в физических и виртуальных средах // Вопросы кибербезопасности. 2016. Вып. 1. С. 24–30. URL: https://cyberrus.com/wp-content/uploads/2016/02/24-30-114-16_4.-Фадин.pdf (дата обращения: 21.04.2020).
25. Щербаков А. Ю. Современная компьютерная безопасность. М.: Книжный мир. 2009. – 352 с. URL: https://computer-museum.ru/books/computer_safety.pdf (дата обращения: 21.04.2020).
26. Зубарев И. В., Радин П. К. Основные угрозы безопасности информации в виртуальных средах и облачных платформах. Вопросы кибербезопасности. 2014. Вып. 2 (3). С. 40–45. URL: https://cyberrus.com/wp-content/uploads/2014/07/vkb_03_06.pdf (дата обращения: 21.04.2020).
27. Мозолина Н. В. Необходимо и достаточно, или контроль целостности виртуальных машин с помощью Аккорд-KVM. Information Security/Информационная безопасность. 2018. Вып. 5. С. 33
URL: https://www.okbsapr.ru/library/publications/mozolina_2018_1/ (дата обращения: 21.04.2020).

REFERENCES:

[1] Durairaj M., Kannan P. A Study On Virtualization Techniques And Challenges In Cloud Computing. International Journal of Scientific & Technology Research. 2014. Vol. 3. P. 147–151.
URL: https://www.ijstr.org/final-print/nov2014/A-Study-On-Virtualization-Techniques-And-Challenges-In-Cloud-Computing.pdf (accessed: 15.04.2020).
[2] Popek G. J., Goldberg R. P. Formal Requirements for Virtualizable Third Generation Architectures. Communications of the ACM. 1974. Vol. 17. P. 412–421. DOI: https://doi.org/10.1145/361011.361073.
[3] Doroshenko V. S., Shadrin D. B. Ispol'zovanie virtual'nykh mashin v obuchenii. Novye obrazovatel'nye tekhnologii v vuze: materialy XII mezhdunarodnoy nauchno-metodicheskoy konferentsii (NOTV-2015). 2015. C. 106–108. URL: https://elibrary.ru/item.asp?id=29903669 (accessed: 15.04.2020) (in Russian).
[4] YamunaDevi, L. & P, Aruna & Dorairaj, Sudha Devi & Priya, Navya. (2011). Security in Virtual Machine Live Migration for KVM. 10.1109/PACC.2011.5979008. DOI: 10.1109/PACC.2011.5979008.
[5] Hyungro L. Virtualization Basics: Understanding Techniques and Fundamentals. School of Informatics and Computing, Indiana University. 2014. P. 1–5. URL: http://dsc.soic.indiana.edu/publications/virtualization.pdf (accessed: 21.04.2020).
[6] Mozolina N. V. Kontrol' tselostnosti virtual'noy infrastruktury i ee konfiguratsii. Voprosy zashchity informatsii. 2016. Vyp. 3. S. 31–33. URL: https://elibrary.ru/item.asp?id=26992575 (accessed: 21.04.2020) (in Russian).
[7] Gordievskikh V.M. Sushchnost', struktura i klassifikatsiya sovremennykh tekhnologiy virtualizatsii. V laboratoriyu uchenogo. 2015. S. 125–133. URL: https://elibrary.ru/item.asp?id=23400691 (accessed: 21.04.2020) (in Russian).

[8] Elmanova N., Pakhomov. S. Virtual'nye mashiny 2007. Komp'yuterPress. 2007. Vyp. 9. S. 29–42. URL: https://compress.ru/article.aspx?id=18046 (accessed: 21.04.2020) (in Russian).
[9] Sayt proekta KVM. Glavnaya stranitsa. URL: https://www.linux-kvm.org/page/Main_Page (accessed: 18.10.2019). (in Russian).
[10] Sayt proekta KVM. Sredstva upravleniya. URL: https://www.linux-kvm.org/page/Management_Tools (accessed: 19.10.2019).
[11] Sayt proekta Libvirt. Chasto zadavaemye voprosy. URL: https://wiki.libvirt.org/page/FAQ#What_is_libvirt.3F (accessed: 22.10.2019).
[12] Goto Y. Kernel-based Virtual Machine Technology. FUJITSU Sci. Tech. J. 2011. Vol. 47. P. 362–368. URL:https://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol47-3/paper18.pdf (accessed: 21.04.2020).
[13] J. E. Smith and R. Nair, “The Architecture of Virtual Machines,” Computer (IEEE), Vol. 38, No. 5, 2005. P.32–38. DOI:10.1109/MC.2005.173. URL: https://www.scirp.org/reference/referencespapers.aspx?-referenceid=449371 (accessed: 21.04.2020).
[14] Operating Systems: Internals and Design Principles. Chapter 14. Virtual Machines. URL: https://www.unf.edu/public/cop4610/ree/Notes/PPT/PPT8E/CH14-OS8e.pdf (accessed: 18.11.2019).
[15] Ispol'zovanie pol'zovatel'skikh atributov v Vcenter Server. URL: http://it-pilot.ru/2013/10/14/atribut/ (accessed: 18.11.2019) (in Russian).
[16] Ryabov A. S., Ugarov D. V., Postoev D. A. Bezopasnost' virtual'nykh infrastruktur. Slozhnosti i nyuansy vypolneniya trebovaniy regulyatora. Kompleksnaya zashchita informatsii: materialy XXI nauchno-prakticheskoy konferentsii. 2016. S. 217–220. URL:https://www.okbsapr.ru/library/publications/ryabov_2015_2/ (accessed: 21.04.2020) (in Russian).
[17] S. Lydin Sredstva zashchity informatsii dlya infrastruktury virtualizatsii: vstroennye ili nalozhennye? URL: http://www.okbsapr.ru/lydin_kzi2017.html (accessed: 01.12.2019) (in Russian).
[18] Konyavskiy V. A., Gadasin V. A. Osnovy ponimaniya fenomena elektronnogo obmena informatsiey. Minsk: Bellitfond. 2004. S. 239–245. URL: https://www.okbsapr.ru/upload/iblock/016/osnovi_ponim_el_obmen_inf.pdf (accessed: 21.04.2020) (in Russian).
[19] Altukhov A. A. Doverennaya zagruzka i kontrol' tselostnosti arkhivirovannykh dannykh. Chast' i tseloe. Voprosy zashchity informatsii. 2016. Vyp. 2. S. 35–39. URL: https://www.okbsapr.ru/library/publications/altukhov_2016_1/ (accessed: 21.04.2020) (in Russian).
[20] Mozolina N. V. Reshenie zadachi kontrolya tselostnosti konfiguratsii, osnovannoe na atributnoy modeli kontrolya dostupa. Voprosy zashchity informatsii. 2017. Vyp. 3. S. 23–25. URL: https://www.okbsapr.ru/library/publications/mozolina_2017_2/ (accessed: 21.04.2020) (in Russian).
[21] Schiffman J., Moyer T., Shal C., Trent J., McDaniel P. Justifying Integrity Using a Virtual Machine. 25th Annual Computer Security Applications Conference (ACSAC), 2009. URL: http://www.patrickmcdaniel.org/pubs/acsac09c.pdf (accessed: 01.12.2019).
[22] Liu D. A Research on KVM-Based Virtualization Security. Applied Mechanics and Materials. 2014. Vol. 543–547. P. 3126–3129. DOI: https://doi.org/10.4028/www.scientific.net/AMM.543-547.3126 (accessed: 21.04.2020).
[23] Lee S., Yu F. Securing KVM-Based Cloud Systems via Virtualization Introspection. 2014 47th Hawaii International Conference on System Sciences. 2014. P. 5028–5037. DOI: 10.1109/HICSS.2014.617. URL: https://ieeexplore.ieee.org/document/6759220 (accessed: 21.04.2020).
[24] Avezova Ya. E., Fadin A. A. Voprosy obespecheniya doverennoy zagruzki v fizicheskikh i virtual'nykh sredakh. Voprosy kiberbezopasnosti. 2016. Vyp. 1. S. 24–30. URL: https://cyberrus.com/wp-content/uploads/2016/02/24-30-114-16_4.-Фадин.pdf (accessed: 21.04.2020) (in Russian).
[25] Shcherbakov A. Yu. Sovremennaya komp'yuternaya bezopasnost'. M.: Knizhnyy mir. 2009. – 352 s. URL: https://computer-museum.ru/books/computer_safety.pdf (accessed: 21.04.2020) (in Russian).
[26] Zubarev I. V., Radin P. K. Osnovnye ugrozy bezopasnosti informatsii v virtual'nykh sredakh i oblachnykh platformakh. Voprosy kiberbezopasnosti. 2014. Vyp. 2 (3). S. 40–45. URL: https://cyberrus.com/wp-content/uploads/2014/07/vkb_03_06.pdf (accessed: 21.04.2020) (in Russian).
[27] Mozolina N. V. Neobkhodimo i dostatochno, ili kontrol' tselostnosti virtual'nykh mashin s pomoshch'yu Akkord-KVM. Information Security/Informatsionnaya bezopasnost'. 2018. Vyp. 5. S. 33. URL: https://www.okbsapr.ru/library/publications/mozolina_2018_1/ (accessed: 21.04.2020) (in Russian).

Поступила в редакцию – 09 мая 2020 г. Окончательный вариант – 06 июня 2020 г.
Received – May 09, 2020. The final version – June 6, 2020.