4. *Han W., Lei C.* A Survey on Policy Languages in Network and Security Management // Computer Networks. 2012. № 56. P. 477–489.
5. *Preda S., Cuppens-Boulahia N., Cuppens F., Toutain L.* Architecture-Aware Adaptive Deployment of Contextual Security Policies// Proceedings of International Conference on Availability, Reliability, and Security. 2010. P. 87–95.
6. *Rahman M., Al-Shaer E.* A Formal Framework for Network Security Design Synthesis // Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems. 2013. P. 560–570.

## REFERENCES:

1. *Rees J., Bandyopadhyay S., Spafford E. H.* PFIRES: A Policy Framework for Information Security // Communications of the ACM. 2003. Vol. 46. Issue 7. P. 101–106.
2. *Chapple M. J., D'Arcy J., Striegel A.* An Analysis of Firewall Rulebase (Mis)Management Practices // ISSA Journal. February. 2009. P. 12–18.
3. Ponemon Institute. Perceptions about Network Security. Survey of IT & IT security practitioners in the U.S. 2011.
4. *Han W., Lei C.* A Survey on Policy Languages in Network and Security Management // Computer Networks. 2012. № 56. P. 477–489.
5. *Preda S., Cuppens-Boulahia N., Cuppens F., Toutain L.* Architecture-Aware Adaptive Deployment of Contextual Security Policies// Proceedings of International Conference on Availability, Reliability, and Security. 2010. P. 87–95.
6. *Rahman M., Al-Shaer E.* A Formal Framework for Network Security Design Synthesis // Proceedings of the 2013 IEEE 33rd International Conference on Distributed Computing Systems. 2013. P. 560–570.

*A. V. Iuzbashev*

**Detecting System of Nested Hardware Virtual Machine Monitor**

*Key words: hypervisor, hardware virtual machine monitor, Intel Vt-x*

Method of nested hardware virtual machine monitor detection was proposed in this work. The method is based on HVM timing attack. In case of HVM presence in system, the number of different instruction sequences execution time values will increase. We used this property as indicator in our detection system.

*А. В. Юзбашев*

## СПОСОБ ОБНАРУЖЕНИЯ ВЛОЖЕННЫХ МОНИТОРОВ ВИРТУАЛЬНЫХ МАШИН

Технология аппаратной виртуализации за последние годы получила широкое распространение. Большинство выпускаемых процессоров компаний Intel и AMD поддерживают данную технологию [1].

Монитор виртуальных машин (МВМ) представляет собой программную прослойку, работающую между аппаратной частью и ядром операционной системы (ОС) и контролирующую все ресурсы операционной системы. Стоит отметить, что не существует штатных средств обнаружения МВМ. Также важным моментом является то, что угроза внедрения МВМ может исходить как от поставщиков оборудования, так и от поставщиков программного обеспечения. Основной особенностью внедрения МВМ является тот факт, что возможна установка без изменения микропрограммы BIOS и загрузочной записи жесткого диска, посредством установки драйвера ОС, который позволяет перевести систему в гостевой режим работы [2].

Были проанализированы особенности времени выполнения последовательности инструкций. Пусть в случае присутствия MBM в системе количество различных возможных значений времени выполнения одной перехватываемой инструкции составляет $n_1$, в случае отсутствия MBM данная величина составляет $n_2$, где $n_1 >> n_2$. В трассе выполняются последовательно k перехватываемых инструкций. Следовательно, возможно $n_1^k$ и $n_2^k$ значений времени выполнения соответственно в случае присутствия и отсутствия в системе MBM. Рассмотрим случай, если длина трассы перехватываемых инструкций увеличится на одну инструкцию, получим число возможных вариантов значений времени выполнения $n_1^{k+1}$ и $n_2^{k+1}$. Можно подсчитать возможное количество значений, на которое произошло увеличение в случае увеличения трассы перехватываемых инструкций. Получаем $\Delta n_1 = n_1^k(n_1 - 1)$ и $\Delta n_2 = n_2^k(n_2 - 1)$, учитывая факт, что $n_1 >> n_2$, получаем, что в случае присутствия MBM при увеличении длины трассы перехватываемых инструкций происходит увеличение количества принимаемых значений на большую величину, нежели в случае отсутствия MBM в системе.

Выявленный признак позволил создать систему обнаружения вложенных мониторов виртуальных машин, использующих технологию аппаратной виртуализации компании Intel. Система обнаружения производит измерение времени выполнения последовательности инструкций при помощи счетчика системных тактов, командой RDTSC. Последовательность команд формируется из инструкций CPUID, которой вызывают безусловный перехват управления на обработчик MBM [3].

Разработанная система позволяет обнаружить как один MBM, так и несколько вложенных MBM в случае противодействия обнаружения со стороны MBM изменением счетчика системных тактов.

## СПИСОК ЛИТЕРАТУРЫ:

1. *Силаков Д. В.* Использование аппаратной виртуализации в контексте информационной безопасности. URL: http://www.ispras.ru/ru/proceedings/docs/2011/20/isp_20_2011_25.pdf. (Дата обращения 02.02.2015)
2. *Коркин И. Ю.* Методика обнаружения нелегитимного программного обеспечения, использующего технологию аппаратной виртуализации. Автореф. дисс. … канд. техн. наук. 2011, М.
3. Intel® 64 and IA-32 Architectures Application Note TLBs, Paging-Structure Caches, and Their Invalidation. URL: http://www.intel.com/products/processor/manuals. (Дата обращения 23.01.2015)

## REFERENCES:

1. *Silakov D. V.* Ispolzovanie apparatnoy virtualizatsii v kontekste informatsionnoy bezopasnosti. URL: http://www.ispras.ru/ru/proceedings/docs/2011/20/isp_20_2011_25.pdf. (02.02.2015)
2. *Korkin I. Yu.* Metodika obnaruzheniya nelegitimnogo programmnogo obespecheniya, ispolzuyushchego tehnologiyu apparatnoy virtualizatsii. Avtoref. diss. … kand. tehn. nauk. 2011, M.
3. Intel® 64 and IA-32 Architectures Application Note TLBs, Paging-Structure Caches, and Their Invalidation. URL: http://www.intel.com/products/processor/manuals. (23.01.2015)