выявления угрозы ИБ локально ОЦИБ анализирует идентификатор нарушителя, связанный с обнаруженной угрозой на предмет соответствия БДР (рис. 1). В случае если прослеживается взаимосвязь, например IP-адрес источника пакетов был уже неоднократно замечен при проведении атак в Интернете, рейтинг значимости выявленного события ИБ значительно повышается.

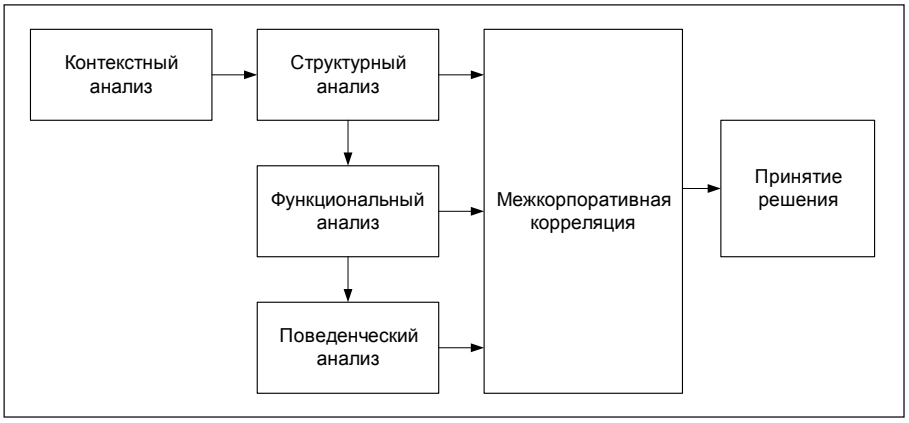

Рис. 1. Схема работы межкорпоративной корреляции

Рейтинг значимости события ИБ в рамках ОЦИБ отражает важность связанной с этим событием угрозы ИБ. Чем более важное событие ИБ, тем меньше вероятность того, что имеет место ложное срабатывание. Таким образом, на основании результатов межкорпоративной корреляции в значительной степени снижается количество ложных срабатываний и, как следствие, повышается эффективность работы ОЦИБ.

И. Ю. Коркин, Т. В. Петрова, А. Ю. Тихонов

МЕТОД ОБНАРУЖЕНИЯ АППАРАТНОЙ ВИРТУАЛИЗАЦИИ В КОМПЬЮТЕРНЫХ СИСТЕМАХ

Новые процессоры Intel и AMD поддерживают режим аппаратной виртуализации (AB) [1, 2], расширяющий возможности как для легитимного, так и для вредоносного программного обеспечения (ВПО).

Необходимо отметить, что штатные средства обнаружения запущенного монитора виртуальных машин (МВМ) в компьютерных системах отсутствуют, а опубликованные — непереносимы [3], уязвимы к обнаружению и, как следствие, неэффективны.

В связи с этим авторами решалась задача разработки средств, позволяющих с достаточной степенью надежности решать вопрос о присутствии или отсутствии МВМ в компьютерных системах даже в случае активного противодействия их обнаружению.

Поскольку упомянутые процессоры обрабатывают возникающие в ОС события с помощью монитора виртуальных машин, с целью обнаружения факта их работы в современных компьютерах был разработан специальный метод, основанный на динамическом изменении числа выполняемых инструкций.

Суть метода состоит в определении времени выполнения набора инструкций, число которых изменяется в ходе работы программы. Измерение времени выполняется для каждого набора с заданным числом инструкций, после чего выполняется блокировка программы на некоторое время.

После выполнения заданного количества инструкций вычисляется среднее время выполнения одной инструкции.

Операция повторяется заданное число итераций, при этом каждый раз после подсчета времени выполняется блокирование программы на некоторое время с целью разделения во времени наборов инструкций.

После завершения всех итераций определяется среднее время выполнения инструкции по всем измеренным наборам.

Процедура повторяется для различных комбинаций значений числа инструкций, задержек времени и числа итераций, в результате получается четырехмерный массив.

Массив включает следующие числа: количество инструкций, время задержки, число итераций и среднее время выполнения инструкции.

Путем анализа полученного массива делается заключение о присутствии МВМ в системе.

В качестве инструкций, исполняющихся циклически, предлагается использовать такие, которые потенциально могут привести к передаче управления МВМ.

При реализации метода в целях обеспечения безопасности использовался ряд различных программно-аппаратных счетчиков [4—14].

Достоинства метода:

- метод универсален, поскольку не зависит от ошибок в работе процессоров и МВМ и основан на особенностях технологии АВ [15—16];
- метод устойчив, поскольку для подсчета задержки используется ряд счетчиков, одновременная компрометация которых трудоемка [16—17].

Метод обнаружения работы МВМ позволит выявлять АВ в компьютерных системах даже в случае присутствия различных средств противодействия обнаружению, в том числе и технологии BlueChicken [18—22].

СПИСОК ЛИТЕРАТУРЫ:

1. Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide.
2. AMD64 Architecture Programmer's Manual Volume 2: System Programming.
3. IA-32 architecture — CPUID. URL: http://www.sandpile.org/ia32/cpuid.htm.
4. Time Stamp Counter. URL: http://www.intel.com/software/products/documentation/vlin/mergedprojects/analyzer_ec/merged-projects/reference_olh/mergedProjects/instructions/instruct32_hh/vc275.htm.
5. Real Time Clock. URL: http://www.insidepro.com/kk/030/030r.shtml.
6. ACPI. URL: http://www.acpi.info/spec30b.htm.
7. HPET Timer. URL: http://www.intel.com/hardwaredesign/hpetspec_1.pdf.
8. The PIT: A System Clock. URL: http://www.osdever.net/bkerndev/Docs/pit.htm.
9. PIC. URL: http://www.osdever.net/bkerndev/Docs/irqs.htm.
10. APIC. URL: http://www.insidepro.com/kk/030/030r.shtml.
11. LAPIC. URL: www.mcst.ru/doc/semenih_090610.doc.
12. MSR — Архитектурный мониторинг производительности процессоров Intel. URL: http://parallel.ru/russia/MSU-Intel/archmon.html.
13. Network Time Protocol. URL: http://www.ntp.org.
14. Precision Time Protocol. URL: http://www.engineering.zhaw.ch/en/engineering/ines/ieee-1588/documents.html.
15. Hypervisor Malware. Honors Thesis. Fionnbharr Davies. October 2007. Supervisor: Richard Buckland. THE UNIVERSITY OF NEW SOUTH WALES.
16. Hagen Fritsch Analysis and detection of virtualization-based rootkits. Fakultдt fьr Informatik. Bachelorarbeit in Informatik.
17. *Douglas P.* Medley, Captain, USAF, Virtualization Technology. Applied to Rootkit Defense. THESIS. AFIT/GCE/ENG/07-08. DEPARTMENT OF THE AIR FORCE.
18. Detection of an HVM rootkit (aka) Blue Pill, Desnos, Filiol.

19. CPU side-channels vs. virtualization rootkits: the good, the bad, or the ugly [ToorCon Seattle 2008].
21. Beyond The CPU: Defeating Hardware Based RAM Acquisition, Joanna RutkowskaCOSEINC Advanced Malware Labs.

С. С. Корт

ПЕРЕОБУЧЕНИЕ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Во многих существующих подходах к построению системы обнаружения вторжений обычно используется статическое обучение классификатора [1]. Однако данный процесс подразумевает стационарность распознаваемого процесса. В общем случае данные, поступающие в систему, не являются стационарными по следующим причинам:

– изменение поведения пользователя с течением времени;

– изменение окружения (появление в сети новых компьютеров и сервисов);

– изменение природы атак на систему;

– зависимость поведения пользователя в определенный момент времени от его поведения до этого момента времени.

Список этих причин можно расширить, но уже он показывает, что говорить о стационарности соответствующих процессов в системе некорректно. Как следствие этого, в идеальном случае обучение должно быть динамическим, а система должна быть адаптивной, периодически переобучаясь. Можно выделить три подхода к переобучению системы:

1) Переобучение системы инициируется пользователем. В том случае, если в системе отсутствует переобучение, она может оценивать поведение пользователя, которое можно охарактеризовать как стационарное.

2) Система обучается постоянно. Каждое новое событие вызывает переобучение. Переобучение при поступлении каждого события приводит к необходимости построения самообновляющегося профиля. К достоинствам данного подхода можно отнести преодоление проблемы адекватного описания модели системы, включающего условие стационарности. К недостаткам подхода можно отнести подверженность атакам на механизм обучения и неэффективность подхода, выражающуюся в слишком частой необходимости перестроения базы знаний.

3) Система переобучается при наступлении какого-то особого события. Данный подход к переобучению является адаптивным и устраняющим недостатки предыдущих двух подходов. Переобучение в данном случае выполняется при наступлении некоторого события, после возникновения которого соответствующий процесс не может оцениваться как стационарный и появляется необходимость перестроения базы знаний системы.

Вне зависимости от типа информации, на основании которой обучается система (по нормальному поведению или по данным об атаках), можно выделить три типа операций: разрешенные политикой безопасности, запрещенные политикой безопасности и неопределенные.

С точки зрения системы обнаружения вторжений, неопределенной является операция, которая может быть оценена как принадлежащая и безопасному поведению и поведению нарушителя. Неопределенная операция может выполняться как нарушителем, так и авторизованным пользователем, как безопасной программой, так и разрушающим программным обеспечением. Необходимо отметить, что четко определенная политика безопасности системы должна уменьшать количество неопределенных операций.