УДК 004.056
DOI: 10.15827/0236-235X.121.128-133

Дата подачи статьи: 07.09.17
2018. Т. 31. № 1. С. 128–133

ОЦЕНКА СТЕПЕНИ ВЛИЯНИЯ СРЕДСТВ РАЗГРАНИЧЕНИЯ ДОСТУПА НА ПРОИЗВОДИТЕЛЬНОСТЬ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Е.Б. Дроботун¹, к.т.н., докторант, drobotun@xakep.ru

¹ Военная академия воздушно-космической обороны им. Маршала Советского Союза Г.К. Жукова, ул. Жигарева, 50, г. Тверь, 170100, Россия

Эффективность средств защиты от информационно-технических воздействий и компьютерных атак играет огромную роль в адекватном функционировании системы. Помимо требований к эффективности выполнения защитных функций, к системам предъявляется требование минимизации влияния атак на процессы функционирования защищаемой системы, в частности, на ее производительность.

В идеальном случае система (или средства) защиты не должна оказывать влияние на производительность защищаемой информационно-вычислительной системы. Вместе с тем следует отметить, что создать какую-либо систему защиты (в том числе и систему разграничения и управления доступом), которая не влияла бы отрицательно на защищаемый объект, практически невозможно. Исходя из этого, одним из критериев выбора системы (или средств) разграничения доступа для защиты какой-либо информационно-вычислительной системы должен быть критерий минимума влияния выбранной системы разграничения доступа на производительность защищаемой информационно-вычислительной системы.

В статье предлагается методика оценки снижения производительности информационно-вычислительной системы при введении в нее средств разграничения доступа, которые являются одной из важнейших составляющих системы защиты информации для информационно-вычислительных систем, а также описываются программные средства, с помощью которых может быть реализована предложенная методика.

Ключевые слова: средства разграничения доступа, ресурсоемкость системы защиты от компьютерных атак, снижение производительности.

Из всех существующих моделей разграничения доступа наибольшее распространение получили две основные модели:

- избирательное (дискреционное) разграничение доступа;
- полномочное (мандатное) разграничение доступа.

Применение данных моделей в информационно-вычислительных системах и средствах вычислительной техники регламентируется руководящими документами ФСТЭК РФ, при этом в информационно-вычислительных системах и средствах вычислительной техники должна быть реализована как минимум дискреционная модель разграничения доступа, а в случае обработки информации, имеющей конфиденциальный характер, обязательно должна быть реализована мандатная модель разграничения доступа.

При дискреционном разграничении доступа определенные операции над конкретным ресурсом запрещаются или разрешаются субъектам или группам субъектов. Практически все существующие операционные системы реализуют именно такую модель доступа по умолчанию.

Мандатное разграничение доступа заключается в том, что все объекты доступа имеют определенные уровни секретности, а все субъекты делятся на группы, образующие иерархию в соответствии с уровнем допуска к информации. Для реализации мандатной модели разграничения доступа, как правило, необходимы дополнительные программные

средства, встраиваемые в операционную систему. При этом следует отметить, что мандатная модель разграничения доступа при реализации ее в операционных системах не заменяет дискреционную модель разграничения доступа, а дополняет ее, внося дополнительный уровень проверки прав доступа субъектов, при их запросах на какие-либо действия с объектами.

Реализация мандатного доступа в различных операционных системах (Linux, Windows), как правило, осуществляется путем встраивания в операционную систему дополнительных программных компонентов, осуществляющих перехват API-функций операционной системы, которые предназначены для осуществления каких-либо действий с объектами доступа (файлами, участками памяти, сетевыми соединениями, интерфейсами ввода/вывода внешних устройств и т.п.) (рис. 1, 2).

Рис. 1. Реализация функции действий с объектами доступа без ее перехвата

Fig. 1. The implementation of the access object action function without interception

Способ перехвата указанных системных функций зависит от типа операционной системы, для которой предназначена система (или средство) разграничения доступа. Для операционных систем семейства Windows перехват функций может быть осуществлен либо внесением изменений в таблицу дескрипторов системных вызовов SDT [1, 2], либо с использованием драйвера-фильтра файловой системы [2]. Для операционных систем семейства Linux перехват указанных функций осуществляется с применением механизма LSM [3–5].

Особенности реализации мандатной модели разграничения доступа в информационно-вычислительных системах различного назначения приводят к существенному снижению производительности защищаемой системы, поскольку права доступа к объекту должны проверяться не только при открытии объекта, но и при проведении любой операции с объектом (чтение, запись, перемещение).

Исходя из этого при построении или при выборе средств разграничения доступа для различных информационно-вычислительных систем необходимо иметь возможность оценивать уровень влияния этих средств на производительность защищаемой системы для выбора наиболее приемлемого варианта реализации мандатной модели разграничения доступа в информационно-вычислительных системах.

Известные методики оценивания влияния средств защиты на качество функционирования и на производительность информационно-вычислительных систем либо не позволяют производить оценку степени влияния средств разграничения доступа [6–9], либо не имеют возможности произвести оценку степени влияния средств разграничения доступа на производительность информационно-вычислительной системы в целом [10]. Помимо этого, в большинстве методик при свертке частных показателей, характеризующих степень влияния средств защиты на производительность защищаемой системы, используется экспертный метод оценки весовых коэффициентов значимости того

или иного частного показателя, что снижает объективность оценки и усложняет проведение самой процедуры оценки.

Таким образом, возникает необходимость в разработке методики оценки степени влияния средств разграничения доступа на производительность защищаемой системы. Методика должна отвечать следующим требованиям:

- иметь возможность оценивать влияние средств разграничения доступа на производительность информационно-вычислительной системы в целом с учетом особенностей ее функционирования и решения возложенных на нее задач;
- весовые коэффициенты, необходимые для свертки частных показателей степени снижения производительности в интегральный, должны определяться без использования методов экспертных оценок;
- в качестве частных показателей, характеризующих степень влияния средств разграничения доступа на производительность защищаемой системы, должны использоваться коэффициенты замедления скорости выполнения системных функций операционной системы, реализующие все контролируемые средством разграничения доступа операции с объектами доступа.

Методика оценки степени влияния средств разграничения доступа на производительность защищаемой информационно-вычислительной системы должна включать в себя следующие этапы (рис. 3):

- определение перечня системных функций операционной системы, подлежащих контролю;
- определение весовых коэффициентов значимости (для выполнения возложенных на защищаемую систему задач) системных функций;
- определение коэффициентов замедления выполнения системных функций операционной системы, подлежащих оценке;
- определение общего (интегрального) показателя снижения производительности защищаемой системы вследствие влияния средств разграничения доступа.

Определение перечня системных функций операционной системы, подлежащих контролю.

На данном этапе выявляются контролируемые средствами разграничения доступа системные функции. Как правило, в большинстве случаев средствами разграничения доступа контролируются системные функции, которые выполняют следующее:

- все файловые операции (открытие файла, чтение из файла, запись в файл, удаление файла, перемещение файла, изменение атрибутов файла, а также аналогичные операции с каталогами);
- операции с памятью (выделение участков памяти, чтение из памяти процесса, запись в память процесса, изменение атрибутов участков памяти, высвобождение участков памяти);
- операции сетевого обмена;
- операции обмена с внешними устройствами.

Определение весовых коэффициентов значимости системных функций. Как было отмечено, при определении весовых коэффициентов значимости системных функций необходимо избегать использования экспертных методов оценивания. В связи с этим предлагается оценивать значимость системной функции исходя из частоты ее вызова при функционировании информационно-вычисли-

тельной системы. Для этого необходимо отслеживать и подсчитывать все вызовы системных функций при штатном функционировании информационно-вычислительной системы за определенный промежуток времени. Для отслеживания вызовов системных функций возможно использование существующих программных средств (в частности, Process Monitor (рис. 4), API-Monitor для операционных систем семейства Windows или утилиты ptrace (strace) из состава систем Linux).

Тогда весовой коэффициент значимости i -й системной функции будет определяться

как $w_i = \frac{x_i}{X}$, где x_i – количество

вызовов i -й системной функции за контролируемый промежуток времени ($i = \overline{1; N}$; N – число системных функций, подлежащих оценке); X – общее количество всех системных вызовов за контролируемый промежуток времени, определяемое как $X = \sum_{i=1}^N x_i$.

Определение коэффициентов замедления выполнения системных функций операционной системы, подлежащих оценке. Данный коэффициент рассчитывается как отношение среднего времени выполнения оцениваемой системной функции без контроля средствами разграничения доступа к среднему времени выполнения оценива-

Рис. 4. Программа Process Monitor из комплекта утилит Sysinternals (показана фиксация числа вызовов API-функции CreateFile)

Fig. 4. The Process Monitor program from Sysinternals set of utilities (fixing the number of API CreateFile function calls)

Рис. 5. Программа оценки времени выполнения API-функций (показана оценка времени выполнения API-функции CreateFile)

Fig. 5. Runtime assessment program for API-functions (estimated API functions CreateFile runtime)

емой системной функции под контролем средств разграничения доступа: $k_i = \frac{\bar{t}_i^*}{\bar{t}_i}$, где $\bar{t}_i$ – среднее

время выполнения i -й системной функции без контроля ее средствами разграничения доступа; $\bar{t}_i^*$ – среднее время выполнения i -й системной функции под контролем ее средствами разграничения доступа.

Для оценки среднего времени выполнения системной функции возможны два способа:

– фиксировать время выполнения каждого вызова системной функции и определять среднее время выполнения следующим образом:

$$\bar{t} = \frac{\sum_{k=1}^n t_k}{n}, \quad (1)$$

где n – число вызовов системной функции; t_k – время выполнения функции в ходе k -го вызова;

– произвести нужное количество вызовов системной функции, зафиксировать общее время выполнения всех вызовов функции, после чего среднее время определить следующим образом:

$$\bar{t} = \frac{T}{n}, \quad \text{где } n \text{ – число вызовов си-}$$

стемной функции; T – общее время выполнения всех вызовов системной функции.

Определение интегрального показателя снижения производительности защищаемой системы вследствие влияния средств разграничения доступа. Данный показатель определяется как аддитивная свертка

коэффициентов замедления работы каждой системной функции с учетом весового коэффициента значимости этой функции: $K = \sum_{i=1}^N k_i w_i$.

Программная реализация. Для оценки времени выполнения была разработана программа оценки времени выполнения API-функций (рис. 5).

Для фиксации временных промежутков выполнения API-функций используется подсчет тактов процессора с помощью API-функций Windows QueryPerformanceCounter и QueryPerformanceFrequency [11]. Первая API-функция возвращает количество импульсов счетчика высокого разрешения, вторая возвращает текущую частоту работы счетчика высокого разрешения (число импульсов за одну секунду). Разделив значение, полученное с помощью функции QueryPerformanceCounter, на значение, полученное с помощью функции QueryPerformanceFrequency, получим время работы вычислительной системы от момента начала работы до момента вызова функции QueryPerformanceCounter. Таким образом, разница замеренных значений времени перед вызовом необходимой API-функции и времени после завершения ее выполнения покажет искомое время выполнения этой API-функции.

Общий алгоритм работы программы приведен на рисунке 6.

В программе организуется цикл необходимого числа повторений вызова оцениваемой API-функции с замером временных промежутков вы-

Рис. 6. Блок-схема алгоритма работы программы

Fig. 6. Flowchart of a program work algorithm

полнения каждой итерации и последующим вычислением среднего времени выполнения оцениваемой функции по формуле (1). Также в программе реализованы оценка доверительного интервала среднего значения времени выполнения API-функции и графическое представление результатов измерений временных промежутков. Программа разработана в среде Delphi 7. Исходный текст доступен по адресу https://github.com/drobotun/API_Performance.

Применение предлагаемой методики. В качестве примера оцениваемых функций были выбраны API-функции Windows, предназначенные для работы с файлами (CreateFile, CopyFile, DeleteFile, MoveFile, ReadFile и WriteFile). Средством разграничения доступа выступал программный комплекс защиты информации Secret Net с настроенными политиками мандатного доступа к объектам файловой системы.

Этап 1. Для выбранных API-функций при выполнении задач общего назначения (редактирование текстовых документов и электронных таблиц, работа с архивами документов и т.п.) в течение 7 минут был осуществлен подсчет количества вызовов этих функций (использовалась программная утилита Process Monitor). Результаты данного эксперимента приведены в таблице 1.

Таблица 1

Результаты оценки весовых коэффициентов значимости системных функций

Table 1

The results of evaluating weight coefficients of importance of six Windows system functions when a data processing system runs general tasks

Наименование API-функции	Количество вызовов API-функции	Весовой коэффициент значимости
CreateFile	452	0,204
CopyFile	84	0,038
DeleteFile	24	0,011
MoveFile	19	0,009
ReadFile	895	0,405
WriteFile	734	0,332

Этап 2. Для выбранных API-функций были произведены оценки времени выполнения функций без контроля их средствами разграничения доступа и под контролем средств разграничения доступа – по 30 вызовов каждой функции с фиксацией времени выполнения каждого вызова. Результаты измерений среднего времени выполнения системных функций, а также рассчитанные коэффициенты замедлений работы системных функций приведены в таблице 2.

Этап 3. С учетом данных, полученных на этапах 1 и 2, общий коэффициент замедления работы системы при введении в ее состав средств разграничения доступа из состава комплекса средств информационной безопасности Secret Net будет со-

ставлять $K = \sum_{i=1}^6 k_i w_i = 1,502 \times 0,204 + 1,231 \times 0,038 + 2,219 \times 0,011 + 2,211 \times 0,009 + 1,768 \times 0,405 + 1,792 \times 0,332 = 1,707$.

Таблица 2

Результаты измерений времени работы API-функций и оценки коэффициента замедления работы функций

Table 2

The results of measuring API-functions execution time and evaluating the deceleration rate of functions work

Наименование API-функции	Время работы функции без контроля t_i (сек.)	Время работы функции под контролем t^*_i (сек.)	Коэффициент замедления работы функции k_i
CreateFile	0,000816	0,001225	1,502
CopyFile	0,002614	0,003219	1,231
DeleteFile	0,000429	0,000952	2,219
MoveFile	0,000440	0,000973	2,211
ReadFile	0,000082	0,000145	1,768
WriteFile	0,000207	0,000371	1,792

Таким образом, предлагаемая методика и ее программная реализация позволяют оценить влияние системы разграничения доступа к объектам доступа на производительность информационно-вычислительной системы и при этом при оценке весовых коэффициентов значимости частных показателей использовать расчетный метод без применения экспертных оценок.

Литература

1. Руссинович М., Соломон Д. Внутреннее устройство Microsoft Windows. СПб: Питер, 2013. 800 с.
2. Зайцев О.В. Rootkits, Spyware/Adware, Keyloggers & Backdoors: обнаружение и защита. СПб: БХВ-Петербург, 2006. 304 с.
3. Corbet J., Rubini A., Kroah-Hartman G. Linux Device Drivers. O'ReillyMedia, Inc., 2005, 636 p.
4. Матвейчиков И.В. Особенности встраивания в ключевые механизмы ядра Linux с использованием LSM. URL: <https://habrahabr.ru/post/196372/> (дата обращения: 3.09.2017).
5. Матвейчиков И.В. Обзор методов динамического встраивания в ядро операционной системы (на примере Linux) // Безопасность информационных технологий. 2014. № 4. С. 75–82.
6. Картавенко М. Тест антивирусов на быстроедействие. 2012. URL: http://www.anti-malware.ru/antivirus_test_performance_2012 (дата обращения: 3.09.2017).
7. Картавенко М. Тест корпоративных антивирусов на быстроедействие. 2012. URL: http://www.anti-malware.ru/corporate_antivirus_test_performance_2012 (дата обращения: 3.09.2017).
8. Шабанов И. Сравнение скорости работы ведущих антивирусов. 2005. URL: http://www.anti-malware.ru/comparisons/compare_speed_antivirus (дата обращения: 3.09.2017).
9. Дроботун Е.Б., Козлов Д.В. Оценка степени влияния антивирусных средств на качество функционирования информационно-вычислительных систем // Программные продукты и системы. 2016. № 4. С. 129–134.
10. Шумилин А.В. Подход к оцениванию влияния средств разграничения доступа к данным на производительность реляционных СУБД // Программная инженерия. 2013. № 4. С. 29–33.
11. Саймон Р. Microsoft Windows API. Справочник системного программиста. Киев: ТИД «ДС», 2004. 1216 с.

**EVALUATION OF A DEGREE OF IMPACT OF ACCESS CONTROL MEANS
ON INFORMATION SYSTEM EFFICIENCY****E.B. Drobotun**¹, Ph.D. (Engineering), Doctoral Student, drobotun@xakep.ru¹ Military Academy of the Aerospace Defense, Zhigareva St. 50, Tver, 170100, Russian Federation

Abstract. There are requirements to effective performance of protective functions by protection systems against information and technical impacts and computer attacks. In addition, such systems are required to have minimal influence on protected system functioning and, in particular, on its performance.

Ideally, a security system (or means) should not have any effect on the performance of a protected information system. At the same time, it should be noted that it is practically impossible to create a protection system (including the system of access control and delimitation) that would not have any negative effect on a protected object. Therefore, one of the criteria for choosing a system (or means) for access control to protect any information and computing system should be the criterion of the minimum effect of the chosen access control system on the performance of a protected information system.

The article proposes the evaluation technique of information system performance degradation at introduction of access control means to it, which are one of the most important components of an information security system for information systems. The paper also describes software that helps implementing the offered technique.

Keywords: means of access control, resource capacity of the computer attack protection system, performance degradation.

References

1. Russinovich M., Solomon D. *Vnutrennee ustroystvo Microsoft Windows* [Microsoft Windows Internals]. St. Petersburg, Piter Publ., 2013, 800 p.
2. Zaytsev O.V. *Rootkits, Spyware/Adware, Keyloggers & Backdoors: obnaruzhenie i zaschita* [Rootkits, Spyware/Adware, Keyloggers & Backdoors: Detection and Protection]. St. Petersburg, BHV-Peterburg Publ., 2006, 304 p.
3. Corbet J., Rubini A., Kroah-Hartman G. *Linux Device Drivers*. O'ReillyMedia, Inc. Publ., 2005, 236 p.
4. Matveychikov I.V. *Osobennosti vstraivaniya v klyuchevye mekhanizmy yadra Linux s ispolzovaniem LSM* [Features embedded in the key mechanisms of the Linux kernel using the LSM]. Available at: <https://habrahabr.ru/post/196372/> (accessed September 3, 2017).
5. Matveychikov I.V. Review of methods for dynamic integration into an operating system kernel (for example Linux). *Bezopasnost informatsionnykh tekhnologiy* [IT Security]. 2014, no. 4, pp. 75–82 (in Russ.).
6. Kartavenko M. *Test antivirusov na bystrodeystvie* [Operation Speed Test of Antiviruses]. 2012. Available at: http://www.anti-malware.ru/antivirus_test_performace_2012 (accessed September 3, 2016).
7. Kartavenko M. *Test korporativnykh antivirusov na bystrodeystvie* [Operation Speed Test of Corporate Antiviruses]. 2012. Available at: http://www.anti-malware.ru/corporate_antivirus_test_performance_2012 (accessed September 3, 2017).
8. Shabanov I. *Sravnenie skorosti raboty vedushchikh antivirusov* [Comparison of Leading Antiviruses Operation Speed]. 2005. Available at: http://www.anti-malware.ru/comparisons/compare_speed_antivirus (accessed September 3, 2017).
9. Drobotun E.B., Kozlov D.V. Assessment of influence of anti-virus software on quality of information-computing system. *Programmnye produkty i sistemy* [Software & Systems]. 2016, no. 4, pp. 129–134 (in Russ.).
10. Shumilin A.V. Approach to the evaluation of the impact of the means of restricting access to data on the performance of a relational DBMS. *Programmnaya inzheneriya* [Software Engineering]. 2013, no. 4, pp. 29–33 (in Russ.).
11. Saymon R. *Microsoft Windows API. Spravochnik sistemnogo programmista* [Microsoft Windows API. Reference Book of a Systemic Programmer]. Kiev, TID DS Publ., 2004, 1216 p.

Примеры библиографического описания статьи

1. Дроботун Е.Б. Оценка степени влияния средств разграничения доступа на производительность информационно-вычислительной системы // Программные продукты и системы. 2018. Т. 31. № 1. С. 128–133. DOI: 10.15827/0236-235X.121.128-133.
2. Drobotun E.B. Evaluation of a degree of impact of access control means on information system efficiency. *Programmnye produkty i sistemy* [Software & Systems]. 2018, vol. 31, no. 1, pp. 128–133 (in Russ.). DOI: 10.15827/0236-235X.121.128-133.