УПРАВЛЕНИЕ ДОСТУПОМ

УДК 004.056

Моделирование и верификация подсистемы управления доступом средства защиты информации Аккорд-X

[1]*А. М. Каннер*; [1, 2] *Т. М. Каннер*
[1] ЗАО «ОКБ САПР», Москва, Россия
[2] Московский физико-технический институт (национальный исследовательский университет), г. Долгопрудный, Московская обл., Россия

Описана модель управления доступом средства защиты информации Аккорд-X, позволяющего реализовать мандатное управление доступом в операционных системах семейства Linux. В модели учитывается дискреционное управление доступом. Моделирование и верификация модели управления доступом произведены на языке темпоральной логики действий Лэмпорта и соответствующих инструментальных средств верификации.

Ключевые слова: подсистема управления доступом, Аккорд-X, модель безопасности, верификация, темпоральная логика.

Подсистема управления доступом операционной системы представляет собой средство защиты информации (СЗИ), содержащее большой объем исполняемого кода. По мере появления новых требований к таким СЗИ их сложность и объемы исполняемого кода возрастают. Это приводит к появлению новых угроз безопасности и уязвимостей, которые чаще всего связаны с ошибками проектирования средств защиты. Во избежание данной проблемы при разработке подсистем управления доступом любой операционной системы (ОС), в том числе ОС Linux, необходимо предварительно проводить моделирование механизмов защиты. Для этого разрабатывают модели безопасности компьютерных систем [1—3].

Требования к разработке моделей безопасности регламентированы рядом нормативных документов [4—8]. Эти требования предназначены для компаний-разработчиков средств защиты информации. Их выполнение является обязательным при проведении работ по сертификации СЗИ в ФСТЭК России. В соответствии с требованиями нормативных документов в разрабатываемой модели безопасности средств управления доступом должны быть отражены реализуемые политики управления доступом и фильтрации информационных потоков. При этом для подтверждения корректности модели безопасности и тем самым корректности работы СЗИ в части отсутствия логических ошибок в разграничении доступа необходимо подтверждать соответствие модели заявленным требованиям. Наиболее объективным путем подтверждения соответствия модели требованиям является ее верификация с применением специальных инструментальных средств.

Рассмотрим подход к верификации модели безопасности для подсистемы управления доступом, входящей в состав программно-аппаратного комплекса "Аккорд-X" [9, 10] производства компании "ОКБ САПР", а также некоторые особенности процесса верификации.

Материалы и методы

Существует множество подходов к моделированию произвольных систем или алгоритмов в целях их верификации на соответствие некоторым формально описанным свойствам [1—3, 11]. Выбран подход к верификации с использованием темпоральной логики действия Лэмпорта (TLA — Temporal Logic of Actions) и метода Model Checking [1, 2, 12, 13] как наиболее доступный для понимания подход, позволяющий в рамках формальной нотации на языке TLA+ описать все необходимые сущности и операции системы, а также свойства безопасности, необходимые для проверки во всех ее состояниях.

Моделирование на языке темпоральной логики действий Лэмпорта позволяет описывать и в дальнейшем верифицировать в автоматическом режиме системы, заданные в виде конечных автоматов [14]. Вместе с тем при использовании данного

Каннер Андрей Михайлович, программист группы программирования ПО для СЗИ отдела программирования СЗИ.
E-mail: kanner@okbsapr.ru
Каннер Татьяна Михайловна, руководитель учебного центра ЗАО «ОКБ САПР», ведущий инженер лаборатории "Прикладные исследования МФТИ-Сбербанк".
E-mail: tatianash@okbsapr.ru

Статья поступила в редакцию 9 сентября 2020 г.

подхода существуют некоторые ограничения на возможность верификации системы, так как метод Model Checking не осуществляет ее полноценную формальную верификацию. В соответствии с этим в рамках нотации модели безопасности TLA+ введены модельные значения для некоторых сущностей системы (количества субъектов, объектов и пользователей). Такие ограничения на сущности системы, заданные модельными значениями, с одной стороны, не могут повлиять на успешность или неуспешность верификации, а с другой — позволяют методу Model Checking завершить верификацию на этапе, когда для системы не существует новых состояний, т. е. когда все достижимые состояния для данного количества сущностей системы уже проверены. Итак:

* *IDs*
* Множества идентификаторов:
$UserIDs \triangleq 0..2$
$SubjectIDs \triangleq 0..2$
$ObjectIDs \triangleq 0..4$

В модели безопасности необходимо задать переменные — те сущности, которые будут изменяться при выполнении операций и изменение которых влечет за собой изменение состояния системы: A — множество произошедших доступов (вспомогательная переменная [15]); O — множество объектов; S — множество субъектов; U — множество учетных записей пользователей:

VARIABLES A, O, S, U
* Переменные модели:
$vars \triangleq (A, O, S, U)$

Любое изменение приведенных переменных модели переводит систему из одного состояния в другое. Например, во множество произошедших доступов можно добавить новый элемент-кортеж — доступ субъекта с идентификатором *s.sid* к объекту с идентификатором *o.oid* по методу доступа *r* из множества *Accesses*:

$$Accesses \triangleq \left\{ \begin{array}{l} \text{"read", "write", "list\_files", "append",} \\ \text{"lookup", "rename\_obj", "rename\_cont",} \\ \text{"ucreate", "udelete", "change\_user\_perm",} \\ \text{"change\_ext\_attr", "change\_cl", "screate",} \\ \text{"sdelete", "delete\_object","create\_object"} \end{array} \right\}$$

$$A' = A \cup \{< s.sid,\ o.oid,\ r >\}$$

Изменение тех или иных переменных модели необходимо описывать в операциях, которые можно выполнить в системе, соответствующих методам доступа *Accesses*: чтение, запись, получение содержимого каталога, дозапись, поиск объекта, переименование объектов и контейнеров, создание и удаление пользователей, изменение прав пользователей, изменение атрибутов и уровня конфиденциальности объектов доступа, создание и удаление субъектов (процессов пользователей), удаление и создание объектов доступа.

Все операции в нотации TLA+ описывают в виде предикатов пред- и постусловий выполнения операции, например:

* Read
* Операция чтения:
$Read\ (s,o) \triangleq$
$\quad \wedge A' = A \cup \{\langle s.sid,\ o.oid,\ \text{"read"}\rangle\}$
$\quad \wedge$ UNCHANGED $\langle S, O, U\rangle$
$Read\ D \triangleq$
$\exists s \in S:$
$\exists o \in O:$

* Проверка прав
$\wedge \vee\ IsUserAdmin(s)$
* *DAC*
$\vee \wedge\ DAC\_may\_do\ (s, o, \text{"read"})$
* *MAC*
$\quad \wedge MAC\_may\_read\ (s, o)$
* Lookup
$\wedge \langle s.sid, o.oid, \text{"}lookup\text{"}\rangle \in A$
* Постусловия
$\wedge\ Read\ (s,o)$

В предусловии *ReadD* описано несколько предикатов, соединенных операторами конъюнкции (логическое И) и дизъюнкции (логическое ИЛИ). Смысл данного предусловия можно трактовать следующим образом: должны существовать субъект и объект, такие, что одновременно выполняются условия:

- либо субъект является администратором, либо должны одновременно быть выполнены свойства дискреционной и мандатной политик управления доступом (предикаты *DAC_may_do* и *MAC_may_read*);
- субъектом должен быть предварительно выполнен поиск объекта доступа.

В постусловии *Read (s, o)* изменяется переменная с произошедшими доступами системы, а остальные переменные модели остаются неизменными.

Предикаты *isUserAdmin*, *DAC_may_do* и *MAC_may_read* реализованы следующим образом.

* Предикаты проверки дискреционного
* управления доступом:

$$IsUser\ Admin(s) \triangleq$$
$$\wedge\ SelectUser(s.uid).is\_admin = \text{TRUE}$$
$$\backslash^*\ a \in Permissions$$
$$DAC\_may\_do(s, o, a) \triangleq$$
$$\wedge\ \exists\ r \in SelectUser(s.uid).acls:$$
$$\wedge\ r[1] = o.oid$$
$$\wedge\ a \in r[2]$$

* Предикаты проверки мандатного
* управления доступом:

$$MAC\_may\_read(s, o) \triangleq$$
$$\vee\ o.cl \le s.cl$$
$$\vee\ \text{"ccnr"} \in o.ext\_attr$$

Здесь *cl* — уровень доступа субъекта или конфиденциальности объекта;

ccnr — специальный атрибут, позволяющий для объекта делать исключение в рамках мандатной политики управления доступом [3].

Аналогичным образом в нотации модели безопасности TLA+ описаны все остальные операции, соответствующие методам доступа *Accesses*, с учетом характерных для них ограничений, в том числе дискреционного и мандатного управления доступом.

Начальное состояние системы описывается с помощью следующего предиката:

* *Init*
* Инициализация:

$$Init \triangleq\ \wedge A = \{\ \}$$
$$\wedge S = \{s0, s1\}$$
$$\wedge O = \{o0,\ o1,\ o2\}$$
$$\wedge U = \{u0,\ u1\}$$

Множество произошедших доступов инициализируется пустым. В системе изначально существуют две учетные записи пользователей: *u*0 — администратор с максимальным уровнем конфиденциальности и *u*1 — модельный пользователь с минимальным уровнем конфиденциальности, а также соответствующие им субъекты-процессы *s*0 и *s*1 и объекты — корневой каталог файловой системы *o*0, вложенный контейнер *o*1 и файл этого контейнера *o*2. Данные модельные сущности выбраны для ускорения процесса верификации. Множество объектов или субъектов может изначально быть пустым, но это приведет к значительно большему количеству состояний системы.

Условия или свойства безопасности, которые необходимо описывать и проверять в рамках нотации TLA+, представляют в виде инвариантов или темпоральных свойств [12, 13]. При этом в рамках данной модели безопасности за счет использования переменной истории (*history variable* [15]) для множества всех совершенных доступов нет необходимости использовать темпоральные свойства, которые, в отличие от инвариантов, зависят от фактора времени и событий в прошлом или будущем. Все свойства безопасности описаны как предикаты, истинность которых проверяется в каждом возможном состоянии системы.

Для модели описаны инварианты: проверяющие правильность функционирования системы и инварианты безопасности. Первой группе принадлежат инварианты *TypeInv* (консистентность типов сущностей системы), *OneAdminExists* (существование в любой момент администратора системы), *NoCyclesInContainers* (отсутствие циклов в контейнерах). К инвариантам безопасности относятся *MacSafety* (невозможность существования вложенного в контейнер объекта с большим уровнем конфиденциальности), *IntegrityInv* (невозможность запуска измененных исполняемых файлов), *LinksSafety* (наследование уровня конфиденциальности всех ссылок от уровня объекта доступа).

* *MACSafety*
* Инвариант безопасности мандатного
* управления доступом для иерархии объектов
* в контейнере:

$$MACSafety \triangleq$$
$$\forall\ o \in O:$$
$$\vee\ \wedge\ o.type \in Containers$$
$$\wedge\ \forall\ ch \in SelectAllChilds(o):$$
$$\wedge\ \vee\ ch.cl \le o.cl$$
$$\vee\ \text{"ccnr"} \in o.ext\_attr$$
$$\vee\ \neg\ o.type \in Containers$$

* *IntegrityInv*
* Инвариант динамического контроля
* целостности:

$$IntegrityInv \triangleq \forall\ e \in Select\ Executables:$$
$$\left(SubjectIDs \times \{e.oid\} \times \{\text{"write", "append"}\}\right) \cap A = \{\}$$

Модель безопасности в нотации TLA+ описана с помощью спецификации *Spec*, для которой зада-

но начальное состояние *Init* и в дальнейшем могут выполняться различные действия из *Next*.

* *Spec*
* Спецификация модели:

$$Spec \triangleq Init \wedge \Box [Next]_{vars}$$

* Invariants
* Теорема, учитывающая все инварианты
* (доказывается в процессе верификации):

$$\begin{aligned} \text{THEOREM } Spec \Rightarrow & \wedge \Box\, TypeInv \\ & \wedge \Box\, OneAdminExists \\ & \wedge \Box\, MACSafety \\ & \wedge \Box\, NoCyclesInContainers \\ & \wedge \Box\, IntegrityInv \\ & \wedge \Box\, LinksSafety \end{aligned}$$

Формальное доказательство отсутствия противоречий в модели безопасности осуществляют с помощью проверки в автоматическом режиме рассмотренных инвариантов в каждом возможном состоянии для спецификации *Spec*.

Результаты

Верификацию разработанной модели проводили с помощью инструментального средства TLC2 v2.15 на СВТ с Intel Core i5-9400 (3,80 ГГц) и объемом оперативной памяти 16 ГБ в 64-разрядной ОС Linux с ядром v5.4.38. Время, затраченное на верификацию с использованием 6 отдельных потоков, составляет от 12 мин до 24 ч в зависимости от выставленных опций верификации — проверки модели с итеративным углублением, начиная с заданной глубины (опция *dfid* от 6 до 8). Общее количество различных проанализированных состояний 2 776 895.

Результаты верификации описанной спецификации относительно заданных инвариантов позволяют сделать вывод о выполнении требований безопасности для всех возможных состояний модели управления доступом.

Верификация модели позволила выявить и устранить недостатки в реализации комплекса "Аккорд-X", а также провести дальнейший анализ возможных скрытых каналов утечки информации при реализации мандатного управления доступом.

Заключение

Рассмотрен подход, использованный в процессе верификации модели безопасности для подсистемы управления доступом комплекса "Аккорд-X". Данный подход не только позволил выполнить формальные требования существующих нормативных документов в области защиты информации, но и способствовал выявлению логических ошибок в работе подсистемы разграничения доступа, которые могли привести к нарушению конфиденциальности или целостности защищаемых данных.

Литература

1. *Козачок А. В.* Спецификация модели управления доступом к разнокатегорийным ресурсам компьютерных систем // Вопросы кибербезопасности. 2018. № 4(28). С. 2—8.
2. *Козачок А. В.* Спецификация модели управления доступом на языке темпоральной логики действий Лэмпорта // Тр. Института системного программирования РАН. 2018. Т. 30. № 5. С. 147—162.
3. *Девянин П. Н. и др.* Моделирование и верификация политик безопасности управления доступом в операционных системах. — М.: Горячая линия-Телеком, 2019. — 212 с.
4. *Мозолина Н. В.* Формальное моделирование политики безопасности: к вопросу о стандартизации процесса // Комплексная защита информации. 2019. С. 96—99.
5. Требования по безопасности информации, устанавливающие уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий (выписка). [Приказ ФСТЭК России от 30 июля 2018 г. № 131]. — М.: ФСТЭК России, 2018. — 17 с.
6. ГОСТ Р ИСО/МЭК 15408-3-2013 Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Ч. 3. Компоненты доверия к безопасности. — М.: Стандартинформ, 2014. — 152 с.
7. ГОСТ Р (проект) Защита информации. Формальное моделирование политики безопасности. Ч. 1. Формальная модель управления доступом. — М.: Стандартинформ, 201х. — 36 с.
8. ГОСТ Р (проект) Защита информации. Формальное моделирование политики безопасности. Ч. 2. Верификация формальной модели управления доступом. — М.: Стандартинформ, 201х. — 36 с.
9. *Каннер А. М., Ухлинов Л. М.* Управление доступом в ОС GNU/Linux // Вопросы защиты информации. 2012. № 3. С. 35—38.
10. *Каннер А. М.* Linux: о жизненном цикле процессов и разграничении доступа // Вопросы защиты информации. 2014. № 4. С. 37—40.
11. *Klein G. et al.* seL4: formal verification of an OS kernel // Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles. 2009. P. 207—220. DOI: https://doi.org/10.1145/1629575.1629596
12. *Lamport L.* The Temporal Logic of Actions // ACM Trans. Program. Lang. Syst. 1994. V. 16. № 3. P. 872—923. DOI: http://doi.acm.org/10.1145/177492.177726
13. *Lamport L. et al.* Specifying and verifying systems with TLA+ // Proceedings of the ACM SIGOPS 10th workshop. 2002. P. 45—48.
14. *Kanner A. M., Kanner T. M.* Testing Software and Hardware Data Security Tools Using the Automata Theory and the Graph Theory // Proceedings of Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology. 2020. P. 615—618.
15. *Lamport L., Merz S.* Auxiliary variables in TLA+ [Электронный ресурс]. URL: https://arxiv.org/pdf/1703.05121.pdf (дата обращения: 01.09.2020).

Modeling and verification of the access control subsystem of Accord-X data security tool

[1]*A. M. Kanner,* [1, 2] *T. M. Kanner*
[1] JSC "OKB SAPR", Moscow, Russia
[2] Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Moscow region, Russia

The article describes the access control model of the Accord-X data security tool, which makes it possible to implement mandatory access control in the Linux operating systems. The model also takes into account discretionary access control. Modeling and verification of the access control model was carried out in the language of Lamport's temporal logic of actions and the corresponding verification tools.

Keywords: access control subsystem, Accord-X, security model, verification, temporal logic.

Bibliography — 15 references. *Received September 9, 2020*