Формальная верификация модели мандатного контроля целостности в KasperskyOS

Владимир Буренков
Разработчик-исследователь

kaspersky

Актуальность задачи

Разработка формальной модели

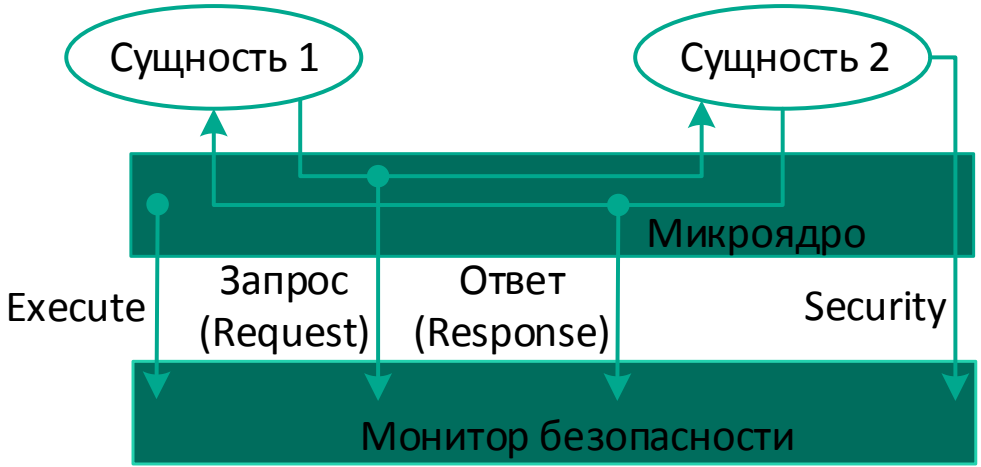
Доказательство свойств модели

Повышение доверия к механизмам безопасности, реализующим модель

Выполнение требований регуляторов (например, ФСТЭК России)

Операционная система KasperskyOS

Все взаимодействия контролируются монитором безопасности



Идеи по разработке и использованию модели

Спроектировать множество правил

Их достаточное количество для сопоставления с взаимодействиями

Монитор безопасности реализует модель

Разрешение или запрет взаимодействий в соответствии с семантикой правил

Контекст Event-B модели

Неизменное состояние модели

Универсум сущностей и объектов	@Entities_and_objects_universe_finite finite(Entities_and_objects_universe)
Виды доступов и информационных потоков	@Access_types partition(Access_types, {read_a}, {write_a})
Множество уровней целостности и свойства отношения частичного порядка на нем	@Integrity_levels_finite finite(Integrity_levels) @le_il_type le_il ⊆ Integrity_levels × Integrity_levels

Машина Event-B модели

Динамическая часть модели

Сущности и объекты

@Entities_type
$Entities \subseteq Entities\_and\_objects\_universe$

Доступы и информационные потоки

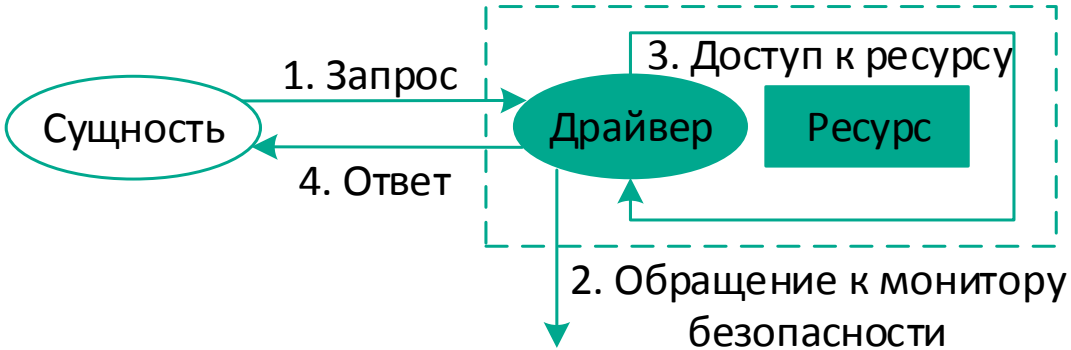
@Accesses_type
$Accesses \subseteq Entities \times Objects \times Access\_types$

Другие отношения на множествах сущностей и объектов

@integrity_level_type
$integrity\_level \in (Entities \cup Objects) \rightarrow Integrity\_levels$

Учет особенностей KasperskyOS в правилах модели

Доступ к объектам возможен только посредством их драйверов



Правила (события) Event-B модели

Правила учитывают возможность компрометации сущностей

Получение **сущностью** (entity) доступа к **объекту** (object) **посредством драйвера** (driver).

event access_write (entity, driver ∈ Entities; object ∈ Objects) where

Resource_driver(object) = driver

driver ∉ Entities_compromised ⇒

integrity_level(object) ↦ integrity_level(entity) ∈ le_il

integrity_level(object) ↦ integrity_level(driver) ∈ le_il

then

Accesses ≔ Accesses ∪ {entity ↦ object ↦ write_a}

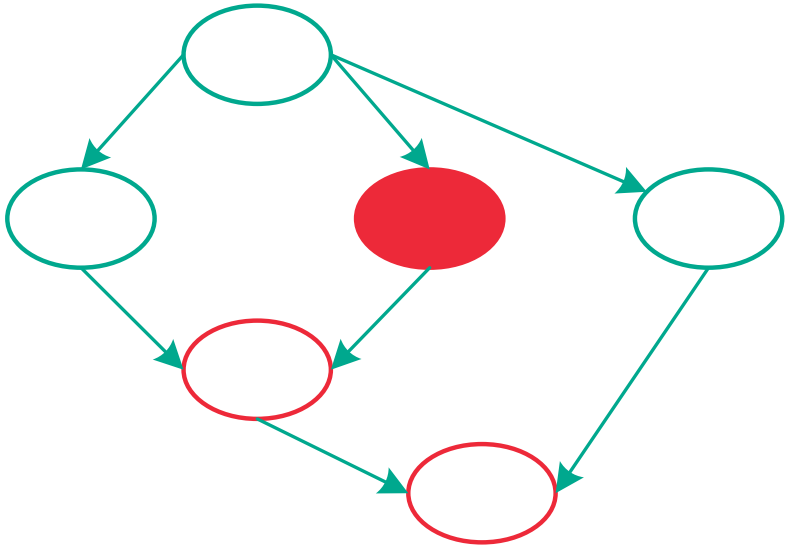
Flows :| (Flows' = Flows ∪ {entity ↦ object ↦ write_m} ∧ **driver ∉ Entities_compromised**) ∨

(Flows' = Flows ∪ {entity ↦ object ↦ write_m, entity ↦ driver ↦ write_m} ∧

driver ∈ Entities_compromised)

Основное свойство безопасности

Модель позволяет ограничить зону захвата контроля



$$\forall u, v \cdot u \in Entities \cup Objects \wedge$$
$$v \in Entities \cup Objects \wedge$$
$$v \mapsto u \mapsto write\_m \in Flows \Rightarrow$$
$$integrity\_level(u) \mapsto integrity\_level(v) \in le\_il$$

$$\vee (\exists w \cdot w \in Entities\_compromised \wedge$$
$$integrity\_level(u) \mapsto integrity\_level(w) \in le\_il)$$

Верификация модели в системе Rodin

Сгенерировано около 300 теорем
Доказательство 54% из них потребовало личного вмешательства (ручной работы)

Проведено множество экспериментов по модификации формальной модели и исследованию ее свойств

Исправлен ряд неточностей модели на естественном и математическом языках

Спасибо за внимание!

kaspersky