# ADV_SPM — Формальные модели политики безопасности на практике

[1,2,3,4]*А.В. Хорошилов <khoroshilov@ispras.ru>,*
[1]*И.В. Щепетков <shchepetkov@ispras.ru>,*
[1]*ИСП РАН, 109004, Россия, г. Москва, ул. А. Солженицына, дом 25*
[2]*ВМК МГУ, 119991 ГСП-1 Москва, Ленинские горы,*
[3]*Московский физико-технический институт,*
*141700, Московская область, г. Долгопрудный, Институтский пер., 9*
[4]*НИУ ВШЭ, Россия, Москва, 101000, ул. Мясницкая, д. 20*

**Аннотация.** В статье рассматривается семейство требований доверия к безопасности ADV_SPM «Моделирование политики безопасности», которое определяется стандартом ГОСТ Р ИСО/МЭК 15408-3-2013 «Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности». Обсуждаются задачи, решаемые этим семейством, и вопросы, которые возникают при попытке интерпретировать его требования. На простом примере представляется подход к формализации политик безопасности при помощи языка формальных спецификаций Event-B и инструментов платформы Rodin.

**Ключевые слова:** информационная безопасность; политики безопасности; формальные модели.

**DOI:** 10.15514/ISPRAS-2017-29(3)-4

**Для цитирования:** Хорошилов А.В., Щепетков И.В. ADV_SPM — Формальные модели политики безопасности на практике. Труды ИСП РАН, том 29, вып. 3, 2017 г., стр. 43-56. DOI: 10.15514/ISPRAS-2017-29(3)-4

## 1. Введение

Стандарт ГОСТ Р ИСО/МЭК 15408 [1-3] применяется в качестве основы при проведении сертификации программного обеспечения, ответственного с точки зрения информационной безопасности. Программное обеспечение, подлежащее оценке, в стандарте обозначается как объект оценки (ОО).

Стандарт состоит из трёх частей:

- часть 1 описывает основные принципы обеспечения безопасности, предлагаемые стандартом;
- часть 2 содержит библиотеку функциональных требований безопасности;





- часть 3 содержит каталог требований доверия, которые определяют набор мероприятий, которые должны быть проведены в ходе разработки и оценки ОО, а также набор документов, которые должны быть сформированы в результате этих мероприятий.

Всего в третьей части стандарта ГОСТ Р ИСО/МЭК 15408-3-2013 [3] представлено 39 семейств требований доверия к безопасности, объединённых в 8 классов доверия.

## 2. Моделирование политики безопасности

В рамках данной статьи основным объектом рассмотрения является семейство требований доверия к безопасности ADV_SPM "Моделирование политики безопасности", являющаяся одним из элементов класса доверия ADV "Разработка" (рис. 1). Также для рассмотрения ADV_SPM нам потребуются элементы класса доверия ASE "Задание по безопасности", поэтому остановимся на этом классе поподробнее.

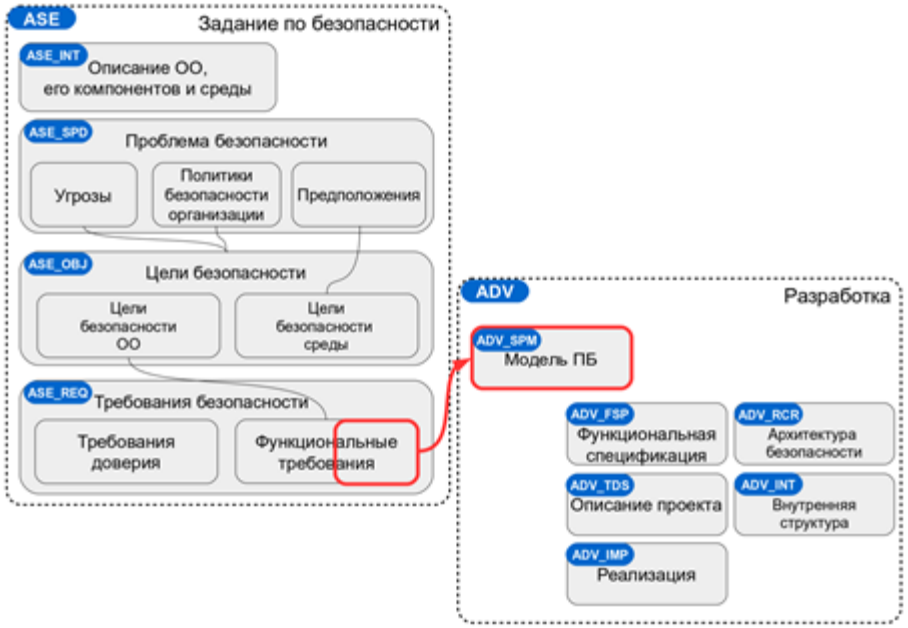


*Рис. 1. Место семейства требований ADV_SPM*
*Fig. 1. Location of the ADV_SPM requirements family*

Семейство требований ASE_INT "Введение в задание по безопасности" требует подготовки документа, содержащего описание ОО, его основных компонентов и среды функционирования.

Семейство ASE_SPD "Определение проблемы безопасности" требует явной формулировки проблемы безопасности, включающей в себя:

- описание угроз, с выделением:
  - источника угрозы,
  - способа реализации угрозы,
  - активов, подвергаемых опасности,
  - нарушаемых свойств безопасности (целостность, доступность, конфиденциальность);
- описание политик безопасности организации (ПБОр) — дополнительных требований, специфичных для организации или области применения ОО;
- предположения безопасности, т. е. предположения о среде функционирования ОО, существенные с точки зрения выполнения функций безопасности.

Семейство ASE_OBJ "Цели безопасности" требует определение целей безопасности с разделением их на два класса: отнесённых к ОО и отнесённых к среде. Для каждой цели требуется обоснование, которое сопоставляет её с идентифицированными угрозами, которым будет противостоять ОО, или с политикой безопасности организации, которая будет выполняться ОО. Цели безопасности среды также могут быть сопоставлены с предположениями безопасности. Также семейство ASE_OBJ требует проведение обобщающего логического анализа совокупности сформулированных целей безопасности, демонстрирующего способность противостоять всем идентифицированным угрозам безопасности и выполнять все установленные положения политики безопасности организации.

Семейство ASE_REQ "Требования безопасности" говорит о необходимости описания требований безопасности к ОО, разделяя их на два вида:

- функциональные требования безопасности ОО, которые, в частности, могут использовать компоненты функциональных требований, представленные во второй части стандарта;
- требования доверия к ОО, которые, как правило, составляются из компонентов требований доверия, описанных в третьей части стандарта.

При этом набор сформулированных функциональных требований должен обеспечить достижение всех целей безопасности, отнесённых к ОО. Для семейства требований доверия к безопасности ADV_SPM "Моделирование политики безопасности", функциональные требования безопасности ASE_REQ играют ключевую роль.

Хотя исторические корни термина "политика безопасности" находятся в политиках управления доступом, и формальные модели политик безопасности





появились именно как модели политик управления доступом, в определении семейства ADV_SPM стандарт использует этот термин в более широкой трактовке, которая предлагает рассматривать "политику безопасности" как произвольное множество логически связанных функциональных требований безопасности.

Таким образом, для одного объекта оценки может быть определено множество политик безопасности и, кроме того, одна формальная модель политики безопасности может описывать несколько политик безопасности. При этом формальная модель политики безопасности строится не только посредством формализации функциональных требований безопасности, но и при помощи представления в модели некоторых деталей реализации, необходимых для придания полноты описываемой картине на соответствующем уровне абстракции.

Целью семейства ADV_SPM стандарт заявляет приобретение дополнительного доверия посредством разработки формальной модели политики безопасности и установления соответствия между функциональной спецификацией ОО и этой моделью политики безопасности.

Для достижения этой цели стандарт требует от разработчика выполнения следующих мероприятий с использованием формальной модели политики безопасности (модели ПБ):

- формальное доказательство отсутствия внутренних противоречий в модели ПБ;
- определение понятия небезопасного состояния для каждой политики безопасности и формальное доказательство недостижимости небезопасных состояний;
- формальное доказательство соответствия между формальной функциональной спецификацией и моделью ПБ;
- демонстрация непротиворечивости и полноты функциональной спецификации относительно модели ПБ, а также относительно моделируемых политик безопасности.

Доказательство отсутствия внутренних противоречий помогает выявить неоднозначные, внутренне противоречивые и противоречащие друг другу элементы функциональных требований безопасности, а также позволяет повысить уровень доверия к корректности представления модели, поскольку ошибки и опечатки в описании сложных моделей не так легко выявить при помощи ручного анализа [4].

Доказательство недостижимости небезопасных состояний предназначено для демонстрации того, что моделируемый подход к реализации функциональных требований безопасности удовлетворяет этим требованиям.

Функциональная спецификация определяется в стандарте как описание интерфейса функций безопасности ОО, которое включает:

- описание всех способов, которыми пользователи могут вызвать ту или иную функцию безопасности;
- описание реакций на запросы пользователя;

но не включает описание реализации этих функций.

Демонстрация соответствия между формальной функциональной спецификацией и моделью ПБ должна показать, что функциональная спецификация является непротиворечивой и полной относительно модели ПБ.

## *3. Формальная модель политики безопасности*

Наш практический опыт [5,6] в первую очередь приходится на построение формальной модели политики безопасности операционной системы специального назначения Astra Linux Special Edition, основанной на мандатной сущностно-ролевой ДП-модели управления доступом и информационными потоками (МРОСЛ ДП-модели [7]), и решении с помощью формальной модели первых двух задач: доказательства отсутствия внутренних противоречий и доказательства недостижимости небезопасных состояний.

Для этой цели использовался язык формальной спецификации Event-B [8], так как он позволяет описывать систему при помощи определения возможных наблюдаемых событий вне зависимости от того, где эти события происходят: внутри специфицируемой системы, вне её, или на границе между целевой системой и её окружением, в отличии от многих других языков формальной спецификации.

В соответствии с предложенных подходом формальная модель ПБ описывается в виде спецификации на Event-B, состоящей из следующих элементов:

- Состояние модели — набор переменных, отражающих текущее состояние как объекта оценки, так и его окружения. Состояние описывается как множество значений переменных модели.
- События модели — формальное представление значимых событий, происходящих как в объекте оценки, так и в его окружении. Семантика событий определяется как трансформация состояния модели. События формально описываются при помощи параметров, предусловий и постусловий.
- Инварианты консистентности — предикат над состоянием модели, описывающий требования к внутренней согласованности значений переменных состояния.
- Инварианты безопасности — предикат над состоянием модели, описывающий недостижимость небезопасного состояния для данной политики безопасности.

Верификация модели ПБО заключается в формальном доказательстве её консистентности и недостижимости небезопасных состояний, которое





выполняется при помощи доказательства сохранения всех инвариантов при любом событии модели.

Рассмотрим, как выглядит на примере модель ПБ на языке Event-B, формализующая функциональное требование безопасности FRU_PRS "Приоритет обслуживания", описанное во второй части стандарта (рис. 2).

**15.2 Приоритет обслуживания (FRU_PRS)**

**15.2.1 Характеристика семейства**

Требования семейства FRU_PRS позволяют ФБО управлять использованием находящихся под контролем ФБО ресурсов пользователями и субъектами так, что высокоприоритетные операции под контролем ФБО всегда будут выполняться без препятствий или задержек со стороны операций с более низким приоритетом.

**15.2.2 Ранжирование компонентов**

FRU_PRS.1 «Ограниченный приоритет обслуживания» предоставляет приоритеты для использования субъектами подмножества ресурсов под контролем ФБО.

FRU_PRS.2 «Полный приоритет обслуживания» предоставляет приоритеты для использования субъектами всех ресурсов под контролем ФБО.

**15.2.3 Управление: FRU_PRS.1, FRU_PRS.2**

Для функций управления из класса FMT могут рассматриваться следующие действия:

а) назначение приоритетов каждому субъекту в ФБО.

**15.2.4 Аудит: FRU_PRS.1, FRU_PRS.2**

Если в ПЗ/ЗБ включено семейство FAU_GEN «Генерация данных аудита безопасности», то следует предусмотреть возможность аудита следующих действий:

а) Минимальный: отклонение операции на основании использования приоритета при распределении ресурса.

b) Базовый: все попытки использования функции распределения ресурсов с учетом приоритетности обслуживания.

**15.2.5 FRU_PRS.1 Ограниченный приоритет обслуживания**

Иерархический для: Нет подчиненных компонентов.

Зависимости: отсутствуют

**15.2.5.1 FRU_PRS.1.1**

ФБО должны установить приоритет каждому субъекту в ФБО.

**15.2.5.2 FRU_PRS.1.2**

**ФБО должны обеспечить доступ к [назначение: *управляемые ресурсы*] на основе приоритетов, назначенных субъектам.**

**15.2.6 FRU_PRS.2 Полный приоритет обслуживания**

Иерархический для: FRU_PRS.1 Ограниченный приоритет обслуживания

Зависимости: отсутствуют.

**15.2.6.1 FRU_PRS.2.1**

ФБО должны установить приоритет каждому субъекту в ФБО.

**15.2.6.2 FRU_PRS.2.2**

ФБО должны обеспечить доступ ко **всем совместно используемым ресурсам** на основе приоритетов, назначенных субъектам.

*Рис. 2. Требование безопасности FRU_PRS "Приоритет обслуживания"*
*Fig. 2. The requirements of the FRU_PRS "Priority of service" family*

Модели (или спецификации) на языке Event-B состоят из компонентов двух типов: контекстов и машин. Контексты содержат статическую, неизменяемую часть модели: определения множеств и констант, а также аксиом, которые используются для описания свойств множеств и констант.

Контекст модели FRU_PRS (рис. 3) содержит определение множества приоритетов *P*, которое с помощью аксиом задаётся как подмножество множества натуральных чисел, состоящее из двух элементов — *High*, или

высокий приоритет, и *Low*, низкий приоритет. В свою очередь, *High* и *Low* определяются как константы, числа 1 и 0 соответственно.

```
CONTEXT
    C0  ›
CONSTANTS
○   P     ›Множество приоритетов
○   High      ›Высокий приоритет
○   Low   ›Низкий приоритет
AXIOMS
○   axm1:   P≔ℕ not theorem ›
○   axm2:   High=1 not theorem ›
○   axm3:   Low=0 not theorem ›
○   axm4:   P={High, Low} not theorem ›
END
```

*Рис. 3. Контекст модели*
*Fig. 3. The model context*

В отличие от контекстов, машины содержат динамическую часть модели. При этом машины могут использовать определённые в контекстах константы и множества. В машинах определяются переменные, инварианты, события. Значения переменных формируют текущее состояние модели, а инварианты ограничивают его.

```
MACHINE
    M0  ›
SEES
○    C0
VARIABLES
○   S     ›Субъекты
○   SP    ›Функция приоритетов субъектов
○   O     ›Объекты
○   R     ›Текущие активные доступы
○   Q     ›Очередь
INVARIANTS
○   inv1:   S≔ℕ not theorem ›
○   inv2:   SP∈S→P not theorem ›
○   inv3:   O≔ℕ not theorem ›
○   inv4:   R∈S↔O not theorem ›
○   inv5:   Q∈S↔O not theorem ›

○   inv6:   ∀s,o·s∈S ∧ o∈O ∧ s↦o∈R ⇒ s↦o∉Q not theorem ›
○   inv7:   ∀s,o·s∈S ∧ o∈O ∧ s↦o∈Q ⇒ s↦o∉R not theorem ›
○   inv8:   ∀s1,s2,o·s1∈S ∧ s2∈S ∧ s1≠s2 ∧ o∈O ∧ s1↦o∈R ⇒ s2↦o∉R not theorem ›
○   inv9:   ∀sr,sq,o·sr∈S ∧ sq∈S ∧ o∈O ∧ sr↦o∈R ∧ sq↦o∈Q ⇒ SP(sr)≥SP(sq) not theorem ›
```

*Рис. 4. Переменные и инварианты модели*
*Fig. 4. Variables and invariants of the model*





В машине модели FRU_PRS (рис. 4) определены 5 переменных: множество субъектов *S*; функция приоритетов *SP*, которая ставит в соответствие каждому субъекту его приоритет; множество объектов *O*; множество пар *R* вида "субъект-объект", описывающее текущие активные доступы субъектов к объектам; множество пар *Q*, описывающее очередь на получение доступа. Типы данных переменных задаются в первых пяти инвариантах.

Остальные инварианты описывают следующие требования:

- inv6: каждая пара вида "субъект-объект", которая принадлежит множеству текущих активных доступов *R*, не может одновременно находиться в очереди на получение доступа *Q*;
- inv7: каждая пара вида "субъект-объект", которая находится в очереди *Q*, не может одновременно принадлежать множеству текущих активных доступов *R*;
- inv8: никакие два субъекта не могут одновременно иметь доступ к одному и тому же объекту;
- inv9: приоритет любого субъекта *sr*, который обладает текущим активным доступом к некоторому объекту *o*, должен быть выше приоритета любого субъекта *sq*, который находится в очереди на получение доступа к тому же объекту.

Текущее состояние модели может быть изменено событием. Каждое событие атомарно и обычно состоит из имени, параметров, охранных условий, и действий. Охранные условия являются набором предикатов, которые ограничивают набор возможных состояний модели, при которых данное событие может произойти. Типы параметров события также задаются в блоке охранных условий. Действия изменяют текущее состояние за счёт модификации значений переменных модели.

В модели FRU_PRS определяются четыре события. Рассмотрим подробнее каждое из них.

Событие *change_priority* (рис. 5) моделирует изменение приоритета субъекта. У события два параметра: субъект *s* и новый приоритет *p*. Охранные условия *grd1* и *grd2* задают типы параметров, а условие *grd3* требует, чтобы новый приоритет *p* отличался от старого. Приоритет субъекта изменяется в действии *act1*, причём данное изменение может нарушить некоторые из определённых инвариантов. Например, если приоритет субъекта повышается, и он находится в очереди на получение доступа к некоторому объекту, текущий активный доступ к которому имеется у другого субъекта, приоритет которого меньше *p*, то будет нарушен инвариант *inv9*. Он будет нарушен и в том случае, когда приоритет субъекта уменьшается, и он обладает текущим активным доступом к некоторому объекту, в очереди на получение доступа к которому находится другой субъект, приоритет которого больше *p*. Чтобы избежать этого, в действии *act2* забирается доступ у тех пар "субъект-объект", которые нарушат

инварианты после изменения приоритета. В действии *act3* те же самые пары добавляются в очередь.

```
change_priority:     not extended ordinary ›
ANY
○  s    ›
○  p    ›
WHERE
○  grd1:   s∈S not theorem ›
○  grd2:   p∈P not theorem ›
○  grd3:   p≠SP(s) not theorem ›
THEN
○  act1:   SP(s) ≔ p ›
○  act2:   R ≔ {x↦y|x↦y∈R ∧ (s↦y∈Q ⇒ SP(x)≥p) ∧ (x≠s ∨ (x=s ∧ (∀z·z∈S ∧ z↦y∈Q ⇒ p≥SP(z))))} ›
○  act3:   Q ≔ {x↦y|x↦y∈Q ∨ (x↦y∈R ∧ ((x≠s ∧ s↦y∈Q ∧ SP(x)<p) ∨ (x=s ∧ (∃z·z∈S ∧ z↦y∈Q ∧ p<SP(z)))))
END
```

*Рис. 5. Событие «change_priority»*
*Fig. 5. «Change_priority» event*

Событие *unsuccessful_access* (рис. 6) описывает ситуацию, когда субъект хочет получить доступ к объекту, но его приоритет меньше приоритета субъекта, который уже осуществляет доступ к этому объекту (*grd3*). В этом случае пара "субъект-объект" добавляется в очередь (*act1*).

```
unsuccessful_access:     not extended ordinary ›
ANY
○  s    ›
○  o    ›
WHERE
○  grd1:   s∈S not theorem ›
○  grd2:   o∈O not theorem ›
○  grd3:   ∃x·x∈S ∧ x↦o∈R ∧ SP(s)≤SP(x) not theorem ›
THEN
○  act1:   Q ≔ Q ∪ {s↦o} ›
END
```

*Рис. 6. Событие unsuccessful_access*
*Fig. 6. «Unsuccessful_access» event*

Событие *access* (рис. 7) описывает успешное получение доступа субъекта *s* к объекту *o*, которые, как и в предыдущих событиях, заданы в виде параметров. Если существует активный доступ к объекту *o* от некоторого другого субъекта, то его приоритет должен быть строго ниже (*grd3*) приоритета *s*. Если существует субъект, который находится в очереди на получение доступа к объекту *o*, то его приоритет должен также должен быть ниже (*grd4*) приоритета *s*. В действии *act1* ко множеству текущих активных доступов *R* добавляется новый доступ от субъекта *s* к объекту *o*, но при этом если какой-то субъект уже обладает доступом к объекту *o*, то его доступ удаляется из *R* и переносится в очередь *Q* (*act2*). Если субъект *s* находился в очереди на получение доступа к объекту *o*, то после успешного получения доступа он будет удалён из очереди (*act2*).





```
access:  not extended ordinary ›
ANY
○  s    ›
○  o    ›
WHERE
○  grd1:   s∈S not theorem ›
○  grd2:   o∈O not theorem ›
○  grd3:   ∀x·x∈S ∧ x↦o∈R ⇒ SP(s)>SP(x) not theorem ›
○  grd4:   ∀x·x∈S ∧ x↦o∈Q ⇒ SP(s)≥SP(x) not theorem ›
THEN
○  act1:   R ≔ {x↦y|(x↦y∈R ∧ y≠o) ∨ (x=s ∧ y=o)} ›
○  act2:   Q ≔ {x↦y|(x↦y∈Q ∧ (x≠s ∨ y≠o)) ∨ (x↦y∈R ∧ y=o)} ›
END
```

*Рис. 7. Событие «access»*
*Fig. 7. «Access» event*

Последнее событие называется *free* (рис. 8) и моделирует удаление доступа из множества текущих активных доступов.

```
free:    not extended ordinary ›
ANY
○  s    ›
○  o    ›
WHERE
○  grd1:   s∈S not theorem ›
○  grd2:   o∈O not theorem ›
○  grd3:   s↦o∈R not theorem ›
THEN
○  act1:   R ≔ R\{s↦o} ›
END
```

*Рис. 8. Событие «free»*
*Fig. 8. «Free» event*

Корректность изменения состояния модели, которое происходит после каждой модификации значений переменных в результате выполнения события, необходимо доказывать, так как при этом могут нарушиться определённые инварианты. Для этого платформа Rodin [9], которая используется для разработки и верификации моделей на Event-B, генерирует специальные утверждения для доказательства, которые могут быть доказаны формальным образом либо автоматическими инструментами, либо в интерактивном редакторе доказательств (рис. 9). При условии адекватности сформулированных в виде инвариантов требований к модели доказательство всех сгенерированных утверждений подтверждает её консистентность и корректность.


*Рис. 9. Формализация и верификация модели FRU_PRS*
*Fig. 9. Formalizaion and verification of the FRU_PRS model*

В рассмотренном примере инвариантом безопасности является *inv9*. Доказательство его сохранности при любом событии означает, что все состояния модели безопасны, и, следовательно, небезопасные состояния недостижимы.

При этом остаётся открытым вопрос о том, насколько корректно ОО будет реализовать формализованную модель ПБ. Для ответа на этот вопрос стандартом ГОСТ Р ИСО/МЭК 15408 предусмотрен анализ соответствия по цепочке модель ПБ — функциональная спецификация — описание проекта — реализация (рис. 1). Но детальное рассмотрение этой цепочки находится вне рамок данной статьи.

## *4. Заключение*

В настоящей работе рассмотрен подход к формализации модели ПБ на языке Event-B, позволяющий решать задачи доказательства отсутствия внутренних противоречий и доказательства недостижимости небезопасных состояний при инструментальном контроле со стороны системы верификации Rodin. Наиболее значимым примером применения этого подхода является формализация и верификация политики безопасности МРОСЛ-ДП, реализованной операционной системе специального назначения Astra Linux Special Edition. Разработанная в рамках данного проекта формальная модель ПБ на языке Event-B обладает следующими количественными характеристиками:

- кол-во констант: 34;
- кол-во аксиом: 30;
- кол-во переменных состояния: 60;
- кол-во инвариантов: 248;
- кол-во событий: 75;
- число уровней уточнения: 4;
- размер спецификации на Event-B: 4393 строк;
- кол-во утверждений для доказательства: 2962.





Таким образом, можно сделать вывод, что рассмотренный подход применим для решения практически значимых задач и может применяться для выполнения требований семейства доверия ADV_SPM "Моделирование политики безопасности", определяемого стандартом ГОСТ Р ИСО/МЭК 15408-3-2013.

## Список литературы

[1]. ГОСТ Р ИСО/МЭК 15408-1-2012 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 1. Введение и общая модель.

[2]. ГОСТ Р ИСО/МЭК 15408-2-2013 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2. Функциональные компоненты безопасности.

[3]. ГОСТ Р ИСО/МЭК 15408-3-2013 Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3. Компоненты доверия к безопасности

[4]. Huynh, N., Frappier, M., Mammar, A., Laleau, R., Desharnais, J.: Validating the RBAC ANSI 2012 standard using B. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. (2014) 255–270

[5]. Devyanin P.N., Khoroshilov A.V., Kuliamin V.V., Petrenko A.K., Shchepetkov I.V. Formal Verification of OS Security Model with Alloy and Event-B. In A. Yamine and K.-D. Schewe, eds. Abstract State Machines, Alloy, B, TLA, VDM, and Z, LNCS 8477:309-313, Proceedings of ABZ-2014, Toulouse, France, June 2-6, 2014, pp. 309-313. DOI: 10.1007/978-3-662-43652-3_30.

[6]. Devyanin P.N., Khoroshilov A.V., Kuliamin V.V., Petrenko A.K., Shchepetkov I.V. Comparison of Specification Decomposition Methods in Event-B. Programming and Computer Software, 2016, Vol. 42, No. 4, pp. 198–205. DOI: 10.1134/S0361768816040022.

[7]. Буренин П.В., Девянин П.Н., Лебеденко Е.В., Проскурин В.Г., Цибуля А.Н. Безопасность операционной системы специального назначения Astra Linux Special Edition. Учебное пособие для вузов. 2-е изд. М.: Горячая линия — Телеком, 2016, 312 с.

[8]. Abrial J.-R. Modeling in Event-B: System and Software Engineering. Cambridge University Press, 2010.

[9]. Abrial J.-R., Butler M., Hallerstede S., Hoang T. S., Mehta F., Voisin L. Rodin: An Open Toolset for Modelling and Reasoning in Event-B. International Journal on Software Tools for Technology Transfer, 12(6), pp. 447-466, 2010.

# ADV_SPM — Formal security policy models in practice

[1,2,3,4]*A.V. Khoroshilov <khoroshilov@ispras.ru>,*
[1]*I.V. Shchepetkov <shchepetkov@ispras.ru>,*
[1]*ISP RAS, 25 Alexander Solzhenitsyn Str., Moscow, 109004, Russia*
[2]*CMC MSU, CMC faculty, 2 educational building,*
*MSU, Leninskie gory str., Moscow 119991, Russia*
[3]*Moscow Institute of Physics and Technology, 9 Institutskiy per.,*
*Dolgoprudny, Moscow Region, 141700, Russia*
[4]*FCS NRU HSE, 20 Myasnitskaya str., Moscow 101000, Russia*

**Abstract**. The paper examines the ADV_SPM "Security policy modelling" assurance family, which is part of the ADV "Development" assurance class and defined by the ISO/IEC 15408-3-2013 "Information technology – Security techniques – Evaluation criteria for IT security – Part 3: Security assurance components" standard. We discuss the objective set by this family, which is to provide additional assurance from the development of a formal security policy model of the target of evaluation security functionality and establishing a correspondence between the functional specification and this security policy model by means of a mathematical proof. We propose an approach to the formalization and verification of security policies using the Event-B modelling notation and the Rodin platform, whose rigour is used to obtain the desired security properties by means of formal machine-checkable proofs. The approach helps to identify and eliminate ambiguous, inconsistent, contradictory, or unenforceable security policy elements. We illustrate this approach with a simplified example of a FRU_PRS "Priority of service" model (defined in the second part of the standard) in which we provide a formal proof that the model contains no inconsistencies, and that an insecure state cannot be reached. We conclude that the approach is applicable for solving practical problems and it can be used to fulfil the requirements of the ADV_SPM assurance family.

**Keywords:** information security; security policy; formal models

**DOI:** 10.15514/ISPRAS-2017-29(3)-4

**For citation:** Khoroshilov A.V., Shchepetkov I.V. ADV_SPM — Formal security policy models in practice. Trudy ISP RAN/Proc. ISP RAS, vol. 29, issue 3, 2017. pp. 43-56 (in Russian). DOI: 10.15514/ISPRAS-2017-29(3)-4

## References

[1]. ISO/IEC 15408-1:2012 Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model (in Russian).
[2]. ISO/IEC 15408-2:2013 Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional requirements (in Russian).
[3]. ISO/IEC 15408-3:2013 Information technology - Security techniques - Evaluation criteria for IT security - Part 3: Security assurance requirements (in Russian).





[4]. Huynh, N., Frappier, M., Mammar, A., Laleau, R., Desharnais, J.: Validating the RBAC ANSI 2012 standard using B. In: Abstract State Machines, Alloy, B, TLA, VDM, and Z. (2014) 255–270
[5]. Devyanin P.N., Khoroshilov A.V., Kuliamin V.V., Petrenko A.K., Shchepetkov I.V. Formal Verification of OS Security Model with Alloy and Event-B. In A. Yamine and K.-D. Schewe, eds. Abstract State Machines, Alloy, B, TLA, VDM, and Z, LNCS 8477:309-313, Proceedings of ABZ-2014, Toulouse, France, June 2-6, 2014, pp. 309-313. DOI: 10.1007/978-3-662-43652-3_30.
[6]. Devyanin P.N., Khoroshilov A.V., Kuliamin V.V., Petrenko A.K., Shchepetkov I.V. Comparison of Specification Decomposition Methods in Event-B. Programming and Computer Software, 2016, Vol. 42, No. 4, pp. 198–205. DOI: 10.1134/S0361768816040022.
[7]. Burenin P.V., Devyanin P.N., Lebedenko E.V., Proskurin V.G., Cibulya A.N. Security of the special purpose Astra Linux Special Edition operating system. Textbook for high schools. 2nd ed. Hot line - Telecom, Moscow [Uchebnoe posobie dlya vuzov. 2-e izd. M.: Goryachaya liniya — Telekom], 2016, 312 p. (in Russian)
[8]. Abrial J.-R. Modeling in Event-B: System and Software Engineering. Cambridge University Press, 2010.
[9]. Abrial J.-R., Butler M., Hallerstede S., Hoang T. S., Mehta F., Voisin L. Rodin: An Open Toolset for Modelling and Reasoning in Event-B. International Journal on Software Tools for Technology Transfer, 12(6), pp. 447-466, 2010.