I. Burdonov, A. Kosachev, V. Kulyamin
Computer security, verification, and conformance theory
MABIT 2006. Вторая международная научная конференция по проблемам безопасности и противодействия терроризму.
33 слайда

Безопасность, верификация и теория соответствия.

1) Представление. 

2) Содержание доклада.

После введения, в котором мы введем термины и сформулируем задачу, мы подробнее рассмотрим предлагаемое отношение конформности, приведем примеры нарушения конформности при композиции и определим преобразование спецификаций для этого отношения конформности.

3) Основные термины.

Безопасность. Будем говорить, что система безопасна, если при любом взаимодействии с ней она не может себя вести недекларированным образом.

Декларированность поведения определяется требованиями к системе.

Соответствие (или конформность) требованиям – это и есть отсутствие недекларированного поведения.

Верификация системы – это проверка её конформности требованиям.

4) Поведение.

Поведение целевой (исследуемой) системы описывается в терминах её «внешнего» наблюдаемого взаимодействия с окружением

У системы может быть и внутренняя, не проявляющаяся во внешних взаимодействиях, ненаблюдаемая активность

Мы рассматриваем широко распространенный частный случай взаимодействия – взаимодействие путем обмена дискретными порциями информации

Порции информации, передаваемые из окружения в систему, будем называть стимулами (или *инпутс*ами), а порции информации, передаваемые из системы окружению, – реакциями (или *аутпутс*ами)

5) Формальное соответствие (конформность).

Пусть есть требования и система. Как установить её конформность требованиям? Для этого надо определить способ моделирования, т.е.

отображения объектов реального мира в объекты модельного мира При этом требования отображаются в спецификацию, а целевая система в её модель – реализацию.
Между объектами-моделями задается отношение конформности, как подмножество декартового произведения. Для конкретных спецификации и реализации проверяется, принадлежит ли эта пара заданному отношению, т.е. производится верификация конформности. А что делать, если у нас нет модели реализации? В этом случае, если взаимодействие является обменом дискретными порциями информации, возможно тестирование – динамическая верификация. При этом из модели выводится набор модельных тестов, для каждого из которых определяется вердикт «прошел тест или нет», и доказывается, что если все тесты из данного набора проходят, то имеет место конформность. После этого модельные тесты транслируются в реализационное представление и пропускаются на целевой системе. Если все они проходят, то конформность доказана.

6) Композиционная система.

Целевая система может быть (и, как правило, является) не монолитной, а состоящей из компонентов. В этом случае всякое взаимодействие системы с окружением реализуется как последовательность взаимодействий компонентов между собой и с окружением. При этом взаимодействие компонентов между собой является внутренней, ненаблюдаемой извне активностью. А можно ли построить безопасную систему из условно-безопасных компонентов? Можно, если учитывать, что окружением для компонентов является не внешний мир, а другие компоненты, поведение которых не произвольное, а определяемое нашими требованиями

7) Запрещенное поведение.

Небезопасный компонент – это не любой компонент, а такой, поведение которого определено, но частично. Такой компонент будем называть условно-безопасным.

Для этого спецификации должны описывать не только декларированное поведение, но и случаи, когда поведение не декларировано или явно не разрешено. Такое поведение в спецификации будем называть «запрещенным» поведением и обозначать символом **γ.**

Композиционная система, построенная из условно-безопасных компонентов, будет безопасной, если запрещенное поведение в любом компоненте будет недостижимо при любом взаимодействии со средой

8) Примеры запрещенного поведения.

В системах динамического распределения памяти поведение процедуры освобождение памяти не декларировано для освобождения незахваченной ранее памяти.

«Кнопка запуска ракеты» в некоторых случаях явно не разрешается.

9) Системы размеченных переходов.

В качестве модельного мира мы выбираем достаточно традиционные системы размеченных переходов – LTS. Одним из обоснований такого выбора является то, что такие системы используются в алгебрах процессов и для них определены операции композиции.

LTS – это ориентированный граф, который задается множеством вершин-состояний, между которыми заданы ориентированные дуги – переходы. Каждый переход помечен меткой, в качестве которых могут быть символы стимулов, реакций, символ внутреннего действия – традиционно обозначаемый τ или символ запрещенного действия γ. Множество символов, которыми могут быть помечены переходы называются алфавитом системы. Так как внутреннее действие может быть в любой системе символ τ обычно не включают явно в алфавит. Одно из состояний выделено – это начальное состояние системы. Полезным в дальнейшем является понятие стабильного состояния – состояния, в котором нет переходов по внутренним или запрещенным действиям. Определим теперь безопасную систему в терминах этого модельного мира. Безопасная система – это система, модельное представление которой LTS без достижимых из начального состояния переходов, помеченных запрещенным действием.

10) Оператор композиции LTS.

При композиции систем некоторые стимулы одной из них являются реакциями другой, и наоборот. Такие пары назовем противоположными действиями.

Напомним, как определяется оператор композиции. Пусть даны две LTS, алфавиты которых A и B. Результатом композиции будет LTS, алфавит которой определяется следующей формулой: это объединение алфавитов, каждый алфавит уменьшается на те символы, для которых в парной LTS есть противоположные. Множество состояний композиционной LTS является декартово произведение множеств состояний LTS компонентов. Начальным состоянием – пара начальных состояний компонентов. Множество переходов определяется следующими правилами вывода.

Среди порождаемых переходов выделяются синхронные переходы по противоположным действиям: переходы взаимодействия – одна система выдает реакцию, а вторая воспринимает её как стимул. Остальные переходы – это асинхронные переходы, происходящие только в одном компоненте, состояние другого при этом не меняется. Такие переходы могут быть по действиям, связанным со взаимодействием с окружением композиции (при этом один из компонентов принимает стимул извне или выдает реакцию вовне), по внутренним действиям одного из компонентов, или по запрещенным действиям, Определенную таким образом композицию будем называть *прямой* композицией.

Эта композиция определяет поведение композиционной системы.

11) Проблема монотонности композиции.

Мы будем рассматривать отношение конформности, являющиеся предпорядком (рефлексивные и транзитивные).

Пусть нам заданы спецификации компонентов. Определим (неявно) свойства которыми должна обладать спецификация композиционной системы.

Первое свойство назовем «Условием монотонности». Будем говорить, что спецификация композиции удовлетворяет условию монотонности, если композиция любых конформных реализаций компонентов конформна этой спецификации.

Второе свойство назовем «Условием максимальности требований». Оно заключается в том, что данная спецификация системы конформна любой спецификации системы, удовлетворяющей условию монотонности.

Назовем спецификацию системы «косой» композицией спецификаций, если она удовлетворяет обоим свойствам.

Проблема заключается в том, что прямая композиция спецификаций компонентов не не является косой композицией: она не удовлетворяет условию монотонности.

В частности, может быть нарушение безопасности – прямая композиция спецификаций может быть безопасной, а композиция конформных реализаций опасна.

12) Предлагаемое решение.

Основная идея предлагаемого решения заключается в следующем явном определении косой композиции: определяется преобразование $\mathcal{T}$: LTS→LTS, называемое *монотонным преобразованием спецификации*, и косая композиция определяется как прямая композиция преобразованных спецификаций.

Это преобразование должно быть эквивалентно по отношению конформности исходной спецификации. Тем самым множество конформных реализаций сохраняется.

13) Содержание 2.

В оставшей части доклада мы определим отношение конформности $\boldsymbol{ioco}_{\beta\gamma\delta}$, приведем примеры немонотонности прямой композиции и нарушения безопасности для этого отношения. На этих примерах мы покажем основные аспекты монотонного преобразования и определим формально это преобразование.

Для определения отношения конформности мы рассмотрим трассы наблюдений, тестовые и безопасные трассы, сформулируем гипотезу о безопасности, формально определим отношение конформности $\boldsymbol{ioco}_{\beta\gamma\delta}$, и покажем связь между конформностью и безопасностью.

14) Поведение – трассы наблюдений.

Для того, чтобы иметь возможность проводить как статическую, так и динамическую верификацию, мы будем формализовать поведение как множество трасс наблюдений.

Мы не будем рассматривать отношения конформности, основанные на состояниях. Такие как изоморфизм и всевозможные виды симуляции: сильная, слабая, и т.п.

Отслеживание состояния (если оно возможно) рассматривается как достоверное сообщение о состоянии и также включается в трассы наблюдений.

Рассмотрим возможные виды наблюдений при внешних взаимодействиях. Это, во первых, наблюдаемые внешние действия – стимулы и реакции. И, во вторых, отсутствие внешних действий – множества отказов. Наблюдение множества отказов возможно только в стабильных состояниях.

Пример. Стабильное состояние LTS в заданном алфавите. И вот возможные множества отказов.

Наблюдение отказов в стабильных состояниях при помощи таймаутов возможно только, если ограничено время выполнения внешнего действия, ограничено время выполнения конечной последовательности внутренних действий и в LTS нет дивергенции – бесконечной последовательности внутренних действий. Поэтому дивергенцию мы будем считать запрещенным действием.

Добавим к видам наблюдений и запрещенные действия.

Напомним, что наша цель – уметь строить безопасные системы из возможно опасных компонентов. Это возможно, если при взаимодействии компонентов не будут наблюдаться запрещенные действия.

Возможные наблюдения зависят от наших возможностей по взаимодействию и наблюдению и от наших гипотез о реализации.

Если все виды наблюдений возможны в полном составе, то это будет семантика трасс отказов.

Мы будем рассматривать минимальные возможности и гипотезы.

15) βγδ– трассы

Конкретизируем наши возможности по взаимодействию и наблюдению:

Мы можем послать любой стимул, после этого мы можем наблюдать прием этого стимула или его блокировку.

Мы можем организовать приём реакций, после этого система, вообще говоря, недетерминировано выбирает какую реакцию ей послать, и мы можем её принять, причем ровно одну или обнаружить «молчание» системы, если ей нечего сейчас посылать.

В результате мы получаем, так называемые βγδ-трассы, т.е. последовательность стимулов, их блокировок, реакций, «молчаний» - δ и запрещенных действий – γ. Причем запрещенное действие может в трассе наблюдений быть только последним наблюдением.

16) Безопасные трассы. 29 кликов+последний

Введем обозначения. Стимул и его блокировка безопасны после некоторой трассы наблюдений, если не существует трассы σ?aγ.

Реакция и молчание безопасны после некоторой трассы наблюдений, если ни для какой реакции не существует трассы σ!xγ.

Трасса является безопасной, если любой её префикс продолжается символом, безопасным после этого префикса, и не существует трассы γ.

Рассмотрим пример. Пусть задана LTS в алфавите {?a, {?a}, !x, !y, δ}

Возмем пустую трассу. Она заканчивается в начальном состоянии. Из этого состояния выходят следующие переходы ... Следовательно, по определению, безопасными символами после пустой трассы являются следующие. Мы отмечаем серым цветом символы, безопасные после трассы, но не продолжающие её во множестве трасс LTS, а черным – продолжающие.

Рассмотрим трассу ?a.

Она заканчивается в следующем множестве состояний... Из этих состояний выходят переходы следовательно трасса ?a может продолжаться этими символами, но стимул ?a после этой трассы НЕбезопасен, пометим это красным цветом

17) Гипотеза о безопасности.

Предлагаемое отношение конформности состоит из двух условий. Первое мы называем гипотезой о безопасности или отношением «безопасен для» .

Реализация безопасна для спецификации тогда, и только тогда, когда:

Если пустая трасса безопасна в спецификации, то она безопасна и в реализации. И, второе, любая безопасная в спецификации и реализации трасса может быть безопасно продолжена в реализации любым символом, безопасным после этой трассы в спецификации.

Формально это можно записать короче с помощью тестовых трасс.

18) ***ioco***$_{βγδ}$ отношение конформности.

Теперь дадим формальное определение предлагаемому отношению конформности.

Реализация ***ioco***βγδ-конформна спецификации тогда, и только тогда, когда:

она безопасна для этой спецификации и

для каждой трассы безопасной в спецификации и реализации любой символ, безопасный в спецификации после этой трассы, возможен в реализации только если он возможен в спецификации
или формально

Множество тестовых трасс спецификации, пересеченное с множеством трасс реализации вложено во множество трасс спецификации.

А полностью формальное определение отношения выглядит так:

Множество тестовых трасс спецификации, пересеченное с множеством трасс реализации вложено в пересечение множества тестовых трасс реализации со множеством трасс спецификации.

19) Примеры.

Приведем несколько примеров. Пусть задана спецификация S.

Реализация I_1 конформна спецификации. Хотя её поведение и отличается от поведения спецификации, но это происходит на продолжениях опасных в спецификации трасс. Они выделены серым цветом.

Реализация I_2 не конформна S, так как она не безопасна для этой спецификации.

Реализации I_3 , I_4 , I_5 , I_6 ,не конформны S, так как I_3 выдает реакцию, котрой нет в спецификации.

I_4 «молчит», а спецификация выдает реакцию

I_5 принимает стимул, а не блокирует как спецификация

I_6 блокирует стимул, а не принимает как спецификация

20) Безопасность и $\boldsymbol{ioco}_{\beta\gamma\delta}$.

Если спецификация безопасна, то конформность сводится к вложенности трасс и реализация тоже безопасна.

Пример.

Если спецификация не безопасна, то реализация может быть как безопасна, так и опасна.

Пример безопасной реализации.

Пример опасной. Напомним, что отношение рефлексивно.

21) Содержание 3.

Далее

Рассмотрим примеры несохранения конформности и нарушения безопасности при композиции. Всё это нас интересует с точки зрения построения безопасных систем из условно-безопасных компонентов.

Мы рассмотрим четыре причины несохранения и четыре аспекта предлагаемой трансфрмации спецификаций для избежания несохранения.

22) Безопасные системы и условно-безопасные компоненты.

Для проверки безопасности системы, построенной из компонентов конформных их спецификациям, нам достаточно проанализировать системную спецификацию.

Системная спецификация должна быть косой композицией спецификаций компонентов (т.е. для неё должны выполняться свойства монотонности и максимальности) и она должна быть безопасна.

Прежде всего продемонстрируем возможность этого на простом примере. Вот два условно-безопасных компонента и вот их композиция.

23) Первый пример: полное стабильное состояние.

Полное стабильное состояние – это стабильное состояние, в котором определены переходы по всем стимулам и по крайней мере одной реакции, т.е в нет нет блокировок стимулов и молчания.

Рассмотрим спецификацию и конформную ей реализацию одного компонента. Они отличаются тем, что у спецификации начальное состояние нестабильное, а у реализации полное стабильное.

У второго компонента спецификация и реализация совпадают (по рефлексивности спецификация конформна самой себе).

И возьмем их композиции – спецификации со спецификацией и реализации с реализацией. При этом у композиции спецификаций будет начальное нестабильное состояние, а у композиции реализаций – Неполное стабильное.

Поэтому у композиции реализаций будет трасса, которой нет в композиции. Отсюда – неконформность композиции реализаций композиции спецификаций.

Как этого избежать? Рассмотрим преобразование, создающее в спецификации полное стабильное состояние

Таким образом преобразованная спецификация эквивалентна исходной, следовательно, реализация будет конформна ей.

В композиции преобразованных спецификаций соответствующее стабильное состояние будет неполным,как в реализации . Отсюда – выполнение условия монотонности.

24) Сингулярность.

Сингулярным состоянием мы называем стабильное состояние из которого выходит не более одного перехода, помеченного реакцией.

Рассмотрим спецификацию и конформную ей реализацию одного компонента. Они отличаются тем, что у спецификации начальное состояние несингулярное, а у реализации –сингулярное.

У второго компонента спецификация и реализация совпадают (по рефлексивности спецификация конформна самой себе).

И возьмем их композиции – спецификации со спецификацией и реализации с реализацией. При этом у композиции спецификаций будет

нестабильное состояние, а у композиции реализаций –неполное стабильное.

Поэтому у композиции реализаций будет трасса, которой нет в композиции. Отсюда – неконформность композиции реализаций композиции спецификаций.

Как этого избежать? Рассмотрим преобразование, создающее в спецификации два сингулярных состояния

Таким образом преобразованная спецификация эквивалентна исходной, следовательно, реализация будет конформна ей.

В композиции преобразованных спецификаций соответствующее состояние будет неполным сингулярным, как в реализации. Отсюда – выполнение условия монотонности.

25) Пауэр-LTS.

Еще одним аспектом трансформации является пауэризация спецификации. Состояния пауэр-LTS это множества состояний после безопасной трассы. Сингуляризация производится после пауэризации. Переходы из пауэр состояний – это объединение переходов из всех состояний, входящих в пауэр состояние. Таким образом преобразованная спецификация эквивалентна исходной. Тем самым они являются взаимными реализациями. Покажем необходимость всех пауэр состояний на примерах.

Пусть спецификация-реализация второго компонента T. Тогда при композиции исходной спецификации и преобразованной спецификации с T мы видим необходимость сохранения в преобразованной спецификации состояния 1.

Аналогично покажем необходимость состояния 2.

Теперь покажем необходимость пауэр состояния 12!y. Для этого рассмотрим реализацию I, конформную исходной и преобразованной спецификациям. При композиции этой реализации с некоторой T мы видим нарушение монотонности. Но для преобразованной спецификации монотонность сохраняется при помощи наличия состояния 12!y.

Аналогично демонстрируется необходимость состояния 12!x.

26) γ-однородность.

Состояние называется однородным, если после всех реакций следует запрещенное действие или все реакции безопасны.

Рассмотрим не однородную спецификацию и ее преобразование в эквивалентную ей однородную.

И рассмотрим конформную реализацию, которую будем компоновать с компонентом T.

Композиция реализаций оказывается опасной, а композиция спец – безопасной, нарушается безопасность. И конформностью

Зато композиция преобразованной спец тоже опасна, и конформность не нарушается.

27) Содержание

28) Монотонное преобразование.

Пусть S спецификация в алфавите C.

Преобразование не меняет алфавит, а состояния – это множества состояний после безопасных трасс + специальное Г-большое состояние.

Мы используем обозначения: мю - безопасная трасса не заканчивающаяся на отказ, а ро – трасса отказов.

29) Монотонное 2

Здесь преобразование как набор правил вывода, которые разбиваются на след группы

- Определение Г состояния.
- Определение нестабильных пауэр-состояний.
- определение сингулярных полных стабильных состояний
- определение гамма-однородных полных стабильных сост
- определение сингулярных не полных стабильных состояний
- определение гамма-однородных не полных стабильных сост
- определение молчаливых не полных стабильных сост

30) Верификация декомпозиции системы

Если заданы не только спецификации компонентов, но и спец системы, мы можем проверить декомпозицию требований. Декомпозиция корректна, если косая композиция конформна заданной спец системы

31) Системы без блокировок.

Для важного частного случая спец без блокировок достаточно гамма-однородности

Если косая композиция безопасна, то онга совпадает с прямой композицией

Поэтому для систем без блокировок косая композиция нужна только для проверки безопасности.

32) Заключение.

Вот основные результаты, представленные в докладе

Мы не упомянули об алгоритмизации и опустили доказательства всех утверждений

33) Спасибо за внимание.

Контакты