УДК 004.056

О.В. Полубелова, И.В. Котенко

ВЕРИФИКАЦИЯ ПРАВИЛ ФИЛЬТРАЦИИ С ВРЕМЕННЫМИ ХАРАКТЕРИСТИКАМИ МЕТОДОМ “ПРОВЕРКИ НА МОДЕЛИ”

Полубелова О.В., Котенко И.В. **Верификация правил фильтрации с временными характеристиками методом “проверки на модели”.**
Аннотация. В статье описывается подход к верификации правил фильтрации межсетевого экрана, в том числе представляются модели, алгоритмы и разработанный программный прототип, предназначенные для обнаружения аномалий фильтрации в спецификации политики безопасности компьютерной сети. Предлагаемый подход основан на применении метода “проверки на модели” (Model Checking) и позволяет использовать темпоральную логику для спецификации и анализа информационных процессов, протекающих в модели компьютерной сети с функционирующей системой безопасности, которые изменяются во времени и могут нарушить такие свойства безопасности, как конфиденциальность и доступность.
Ключевые слова: сетевая безопасность, верификация, проверка на модели, аномалии правил фильтрации.

Polubelova O.V., Kotenko I.V. **Verification of security policy filtering rules with temporal parameters by Model Checking.**
Abstract. The paper outlines an approach to verification of filtering rules of firewalls. The approach is intended for detection and resolution of filtering anomalies in the specification of the security policy of computer networks. It is based on Model Checking technique. The paper proposes the models of computer networks, the models of firewalls and filtering anomalies, as well as the algorithm of detection of such anomalies. The main peculiarities of the approach consist in using Model Checking exactly to detect the anomalies of filtering rules and in ability to specify temporal parameters in filtering rules.
Keywords: network security, verification, model checking, anomalies of filtering rules.

1. Введение. Верификация правил фильтрации межсетевого экрана - актуальная проблема для многих систем управления корпоративной компьютерной сетью на основе политик безопасности. С ростом размеров сети, а значит, и количества разделяемых ресурсов и пользователей, правила фильтрации множатся и плохо поддаются неавтоматизированному анализу.

Применение метода “проверки на модели” [1-5] для верификации правил фильтрации, описанного в данной статье, позволяет снизить риски нарушения таких свойств безопасности, как доступность и конфиденциальность. Процесс анализа правил становится автоматизированным и проводится с помощью программного средства SPIN [2, 6].

Ранее авторы уже публиковали серию статей по данной тематике [7-11], в которых были описаны предварительные модели

компьютерной сети и межсетевого экрана, а также предложен обобщенный алгоритм верификации правил фильтрации. Дальнейшие исследования авторов были посвящены формализации аномалий фильтрации с помощью линейной темпоральной логики (Linear Time Logic, LTL) [3]. Предложенный в данной статье подход отличается от предыдущих работ авторов и других релевантных подходов применением метода “проверки на модели” для правил фильтрации политики безопасности с учетом временных характеристик. В правила фильтрации были добавлены временные характеристики, позволяющие задавать время применения правила. Примером таких правил является правило, разрешающее доступ к сетевому принтеру в локальной сети с 9:00 до 18:00. Также была проведена модификация и развитие отдельных моделей, алгоритмов и реализации программного прототипа системы верификации правил фильтрации на основе метода “проверки на модели”.

Статья организована следующим образом. В первом разделе проводится анализ релевантных работ. Во втором разделе описана сущность метода “проверки на модели” и роль в нем темпоральной логики, а также кратко представлены разработанные модели компьютерной сети, межсетевого экрана и аномалий фильтрации и их формализация на языке линейной темпоральной логики. Третий раздел посвящен методике верификации правил фильтрации и практической реализации системы верификации. В заключении сформулированы результаты работы и основные направления дальнейших исследований.

2. Анализ релевантных работ. В ходе работы над задачей верификации правил фильтрации политики безопасности были рассмотрены релевантные работы по обнаружению конфликтов в правилах политик безопасности и использованию для такого анализа метода “проверки на модели”.

Большая часть работ, в которых исследуются конфликты и аномалии в правилах политик безопасности, посвящены преимущественно правилам авторизации. В [12] рассматривается применение метода “проверки на модели” для верификации правил авторизации политики безопасности для мобильных систем. Мобильные системы характеризуются понятием локализации (с указанием, например, сайтов, где они запущены) и способностью выполняться в различных локализациях. Это приводит к возможным проблемам с безопасностью. Авторы [12] формализовали представление отдельных компонентов мобильных систем в виде структур Крипке [5], а также предложили язык

спецификации политики безопасности, который включает в себя правила для манипулирования идентификатором локализации.

Статья [13] посвящена обнаружению конфликтов в правилах авторизации с применением подхода, основанного на построении абстрактных синтаксических деревьев. В [14] исследуются конфликты авторизации, и предлагается оригинальный подход к их обнаружению. Также представлено программное средство для разработки правил авторизации и поиска конфликтов. Работа [15] посвящена верификации правил авторизации с использованием метода “проверки на модели”.

Все работы, предметом исследований которых являются правила фильтрации, предлагают различные методы обнаружения конфликтов, однако метод “проверки на модели” в этих работах не используется. В [16] рассматривается использование аргументационной логики для спецификации и анализа правил фильтрации межсетевого экрана. В [17] предлагается подход для обнаружения аномалий фильтрации, и описывается разработанное программное средство, в котором реализован этот подход.

В процессе построения моделей для верификации правил фильтрации был рассмотрен ряд релевантных работ, посвященных моделированию. В [18] описывается разработанный язык и инструмент Rebeca, позволяющий верифицировать параллельные и событийно управляемые системы. Рассматриваются элементы моделирования верифицируемых систем, с примерами для программного средства SPIN. В этой работе не обсуждаются аспекты верификации правил политики безопасности, но описан процесс построения верифицируемой модели.

В [19] представлен подход, основанный на методе “проверки на модели”, предназначенный для формального моделирования и анализа реализации атак на компьютерную сеть. Авторы предлагают набор функций, типов данных и процессов для моделирования компьютерных атак.

В [20] описываются требования к подходу, основанному на использовании шаблонов. Этот подход применяется при разработке UML-диаграмм, где каждый шаблон имеет секцию ограничений для спецификации безопасности и других неизменяемых свойств. Авторы [20] анализируют диаграммы этих спецификаций, описанные на языке Promela и верифицируемые с помощью программного средства SPIN.

На основании рассмотренных работ в данной статье предлагаются модели компьютерной системы и межсетевого

экрана, вводится формализация аномалий фильтрации на языке линейной темпоральной логики, а также приводится методика обнаружения таких аномалий. Отличительной особенностью данной работы является применение метода “проверки на модели” для верификации правил фильтрации, содержащих временные параметры.

3. Сущность метода. Модели компьютерной сети, межсетевого экрана и аномалий.

3.1. Сущность метода “проверки на модели”. Сущность метода “проверки на модели” заключается в переборе состояний, в которые может перейти система в зависимости от запросов пользователей или других подсистем и ответов компонента, принимающего решения о разрешении или отклонении таких запросов.

Последовательность действий при переборе зависит от условий, которые сформулированы на языке линейной темпоральной логики и выражают корректные состояния системы. Состояние системы определяется набором значений переменных, а изменение состояния вызывается выполняющимися в ней параллельными процессами.

Процесс, который должен выполняться в очередной момент времени, выбирается случайно. Система рассматривает все возможные последовательности шагов для заданных процессов и сигнализирует о потенциальном некорректном состоянии. После этого пользователю выдается ”трасса”, т.е. последовательность шагов, ведущая к некорректному состоянию системы относительно заданных условий.

На рис. 1 представлена обобщенная схема выполнения метода “проверки на модели” для верификации правил фильтрации политики безопасности. На вход верификатору подается формальная спецификация системы, описанная структурами Крипке, и формальная спецификация требований, заданная формулами линейной темпоральной логики (LTL). По результатам верификации определяется, есть ли нарушения спецификации требований, т.е. удовлетворяет ли система заданной спецификации.

Метод “проверки на модели” подразумевает моделирование сущностей, значимых для решения поставленной задачи в рассматриваемой предметной области, и их поведения. Для выбора подхода к моделированию необходимо определить, к какому классу относится моделируемая система.

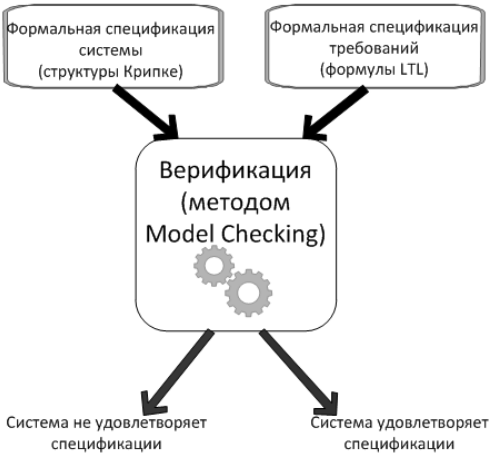

Рис.1. Обобщенная схема выполнения метода “проверки на модели”

Компьютерная система, защищенная межсетевым экраном, относится к классу реагирующих систем. Для таких систем характерна необходимость частого взаимодействия с окружающей средой и длительное время функционирования. Поэтому для адекватного моделирования их поведения нельзя ограничиться только рассмотрением отношения вход-выход.

К отличительным чертам реагирующих систем можно отнести также изменение их состояния с течением времени. Состояние системы описывает систему в определенный момент времени, используя значения множества переменных состояния. Необходимо определять изменения состояния системы в зависимости от тех или иных действий. Изменение описывается указанием состояния системы до и после выполнения действия. Таким образом, переход в системе можно задать как пару состояний до и после изменения. Вычисления реагирующей системы можно определить в терминах переходов системы. В соответствии с [28] определим вычисление как последовательность состояний, каждое из которых получено из предыдущего посредством некоторого перехода. Существуют различные подходы к описанию реагирующих систем, в том числе на основе моделей множества состояний, последовательности состояний, последовательности событий, дерева состояний и др. Для формализации спецификаций таких систем подходит модель Крипке [5]. Она состоит из множества состояний, множества переходов между состояниями и функции, которая помечает каждое

состояние набором свойств, истинных в этом состоянии. Пути в модели Крипке соответствуют вычислениям системы. Для описания параллельных систем можно воспользоваться формулами логики предикатов первого порядка, по которым строится модель Крипке, адекватно соответствующая моделируемой системе.

Задача верификации компьютерной системы, защищенной межсетевым экраном, на предмет обнаружения аномалий в правилах фильтрации политик безопасности может быть формально определена следующим образом. Модель компьютерной сети, которая представляет собой параллельную систему с конечным числом состояний, задается моделью Крипке. Спецификация моделируемой системы представляет собой набор темпоральных формул, выражающих желаемое поведение системы. Такие формулы будут выражать отсутствие аномалий в правилах фильтрации. Таким образом, задача верификации представляет собой проверку выполнимости данных формул на модели Крипке.

Далее введем определение модели компьютерной системы и межсетевого экрана через модель Крипке, а затем опишем спецификацию требований на отсутствие аномалий формулами темпоральной логики. Назовем эту спецификацию моделью аномалий фильтрации.

3.2. Модель компьютерной сети и межсетевого экрана. В процессе исследований были выделены две основные сущности, подлежащие моделированию: компьютерная сеть и межсетевой экран. Для анализа правил фильтрации на предмет аномалий достаточно рассмотреть модель системы, состоящую из этих сущностей.

Предлагаемая *модель компьютерной сети* предназначена для представления структуры сети, ее основных элементов и сетевых процессов. Она включает в себя два базовых компонента: топологию сети и генерируемый в сети трафик. Для задачи верификации правил фильтрации из общей топологии выделяется расположение хостов и межсетевых экранов в сети. При генерации трафика все адресное пространство сокращается до минимума, необходимого для выявления всех возможных аномалий, при этом рассматриваются только хосты, которые специфицированы в правилах фильтрации.

Модель межсетевого экрана предназначена для представления межсетевого экрана и алгоритмов его работы. Основными компонентами этой модели являются сетевые идентификаторы,

заданные наборы правил фильтрации, а также алгоритм обработки сетевого трафика.

Рассмотрим построение формальной спецификации этих моделей на основе [28].

Введем определение модели Крипке. Рассмотрим множество AP атомарных высказываний. Моделью Крипке M_N над множеством AP назовем четверку $M_N = (S_0, S, R, L)$, где S – конечное множество состояний, $S_0 \subseteq S$ – множество начальных состояний (в данной модели не рассматривается отдельно), $R \subseteq S \times S$ – отношение переходов, где для каждого состояния $s \in S$ должно существовать такое состояние $s' \in S$, что имеет место $R(s, s')$, $L : S \to 2^{AP}$ – функция, которая помечает каждое состояние множеством атомарных высказываний, истинных в этом состоянии.

Модель Крипке строится на основании формул $\Re$ первого порядка с учетом следующих правил:

- Множество состояний S - это множество всех оценок над множеством переменных V.
- Какова бы ни была пара состояний s и s', отношение $R(s, s')$ соблюдается в том и только в том случае, когда формула $\Re$ обращается в $True$, после того, как каждой переменной $v \in V$ присвоено значение $s(v)$ и каждой переменной $v' \in V'$ - значение $s'(v')$.

Для того, чтобы описать спецификацию, выражающую свойства моделируемой системы, необходимо определить множество атомарных высказываний. Атомарные высказывания представляют собой присваивание переменным из множества V значений из множества D, которое называется доменом.

Определим переменные моделируемой системы, которые характеризуют ее состояние: u - правило фильтрации, p - сетевой пакет.

Введем множество V, которое содержит набор переменных, характеризующих состояние моделируемой системы: $V = (u, p)$.

Введем также множество V', которое содержит штрихованные переменные для каждой переменной из множества V и представляет собой значения переменных в следующем состоянии: $V' = (u', p')$.

Такое разделение необходимо для того, чтобы определять переходы в моделируемой системе.

Переменные из множеств V и V' принимают значения из конечного множества D.

Определим домены D, используемые в модели компьютерной сети, в виде следующего множества $D = \{O, H, I, T, B, A, M_1, M_2, M_1^1, M_2^1, P, F, W, AF\}$, где N – множество натуральных чисел; Z – множество целых чисел; $M_i^j = \underbrace{M_i \times M_i \times ... \times M_i}_{j\ раз}$ – декартово произведение из j элементов; $O = [0, 255] \cap Z$ – множество октетов; $H = [0, 2^{16} - 1] \cap Z$ – множество значений приоритетов; $I = [0, 2^{32} - 1] \cap Z$ – множество значений типа integer; $T = A*$ - множество строк, состоящих из различных комбинаций символов алфавита A; $B = \{true, false\}$ – множество значений boolean; $A = O \times O \times O \times O$ – множество сетевых адресов; $M_1 = \{t_1, t_2, t_3\}$ – множество сетевых протоколов, $t_1, t_2, t_3 \in T$, $t_1 = IPv4, t_2 = udp, t_3 = tcp$; $M_2 = \{k_1, k_2\}$ – множество привилегий, $k_1, k_2 \in T$, $k_1 = Enabled, k_2 = Disabled$; M_1^1 – множество значений типа сетевого протокола для правила фильтрации; M_2^1 – множество значений привилегии для правила фильтрации; $P = I$ – множество сетевых портов; $F = B \times T \times M_1 \times M_1^1 \times A \times P \times M_2 \times M_2^1 \times H$ – множество правил фильтрации; $W = A \times P \times A \times P$ – множество сетевых пакетов; $AF = 2^{(U \times E)}$ – множество примененных правил фильтрации, $U \subset F$ – множество используемых в модели правил

фильтрации, определяющих политику фильтрации, $E \subset W$ – множество сетевых пакетов в модели.

Множество состояний в модели будет определяться следующим образом: $S = E \times U \times AF$, т.е. на каждом шаге система характеризуется сетевым пакетом, который обрабатывается межсетевым экраном, применяемым правилом межсетевого экрана, парой значений пакета и примененного к нему правила фильтрации.

Для построения системы переходов в модели Крипке выделим следующие виды возможных изменений в системе:

1. Для сетевого пакета p проверяется возможность применения правила u из множества U до тех пор, пока все правила фильтрации не будут проанализированы. Если правило можно применить к данному сетевому пакету, то пара, состоящая из пакета и примененного правила (p,u), добавляется в массив пар, включающих пакет и правило фильтрации.

2. Сетевой пакет p обработан, и обрабатывается пакет p'. К нему начинают применяться все правила фильтрации.

3. Анализируется пара, включающая сетевой пакет и примененное к нему правило (p,u). Если проанализирован весь набор примененных к одному пакету правил, то они удаляются.

Для описания системы переходов используются следующие функции:

- $'$ - переход переменной в следующее состояние;
- $\max(u)$ - функция для определения достижения максимального значения для индекса правила фильтрации;
- $apply(u, p)$ - функция для определения применимости правила фильтрации для текущего сетевого пакета;
- u_0 - первое правило фильтрации.

Отношение переходов выражается формулой:

$$\Re(p,u,p',u') \equiv u' \neq \max(u) \Rightarrow p' = p \wedge u' = next(u) \wedge$$
$$\wedge (apply(u, p) \Rightarrow (p,u) \in h')$$
$$\wedge$$
$$u = \max \Rightarrow (p \neq \max(p) \Rightarrow p' = new(p) \wedge u' = u_0) \wedge$$
$$\wedge apply(u, p) \in h'$$

Таким образом, определена модель Крипке M_N для компьютерной системы, защищенной межсетевым экраном.

3.3. Модель аномалий фильтрации. Для того чтобы задать спецификацию требований к системе, необходимо ввести определение аномалий фильтрации и их классификацию.

Для этого рассмотрим пример конфигурации моделируемой компьютерной системы и спецификации правил фильтрации, которые представлены на рис. 2.

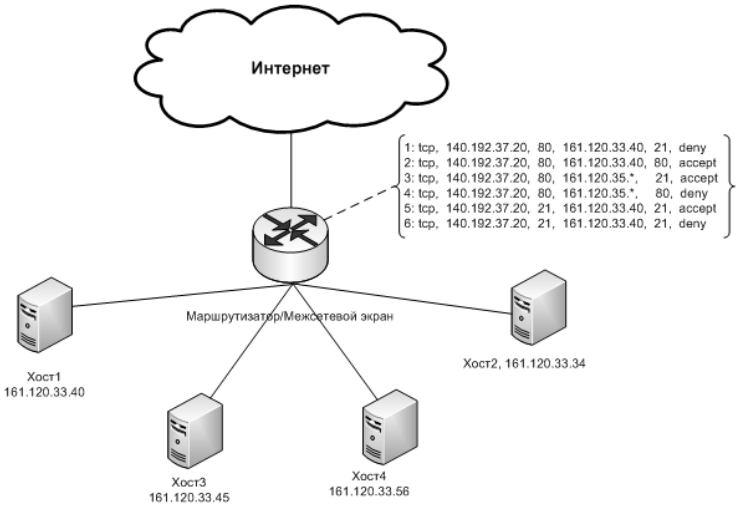

Рис.2. Пример конфигурации моделируемой компьютерной системы, защищенной межсетевым экраном

Компьютерная система состоит из четырех хостов, которые обмениваются сетевым трафиком через межсетевой экран. Фильтрация трафика осуществляется с помощью таблицы доступа.

На рис. 2 представлено шесть правил фильтрации. Каждое правило включает его номер, используемый протокол, адрес и порт отправителя и получателя, а также действие (или привилегию) правила (allow – разрешить, deny – запретить).

Моделируемая система специфицируется на основе задания свойств, которые не должны нарушаться. В данной работе к ним относятся свойства, задающие аномалии фильтрации. Такая спецификация была формализована на языке линейной

темпоральной логики, которая была разработана Амиром Пнуэли для спецификации параллельных программных систем. В темпоральной логике истинность логических формул зависит от момента времени, в котором вычисляются значения формул. В ее основе лежит простейшая логика высказываний, формулы которой строятся из конечного числа атомарных предикатов (утверждений) и булевых операций над ними. Темпоральные операторы позволяют установить истинность высказываний, упорядоченных во времени.

Чаще всего в научных исследованиях противоречия и некорректности в правилах политик безопасности определяют как конфликты. Основываясь на работе [22], введем определения конфликта и аномалии. Определим, какие противоречия приведут систему в нерабочее состояние, а какие не повлияют на стабильность работы системы, хотя и снизят ее эффективность.

Предлагается различать два вида противоречий:

- **конфликт** – это противоречие между правилами политики безопасности или между политикой и описанием сети, которое ведет к недетерминированному поведению системы защиты. Возможны две ситуации, приводящие к подобной неопределенности: (1) система защиты должна активировать два или более правил, но действия (привилегии), задаваемые этими правилами, противоречивы; (2) система защиты должна активировать правила, но действия (привилегии), задаваемые этими правилами, не могут быть выполнены;
- **аномалия** – это противоречие между правилами политики безопасности или между политикой и описанием сети, при котором одно или более правил политики никогда не будут активированы.

Таблицы доступа, которые формируются из наборов правил фильтрации, имеют защиту от возможных конфликтов, обеспечиваемую заданием приоритетов для правила.

Таким образом, при наличии в таблице доступа двух правил, заданных для одного диапазона адресов, но имеющих различные привилегии (разрешить или запретить доступ), межсетевой экран выберет правило с более высоким приоритетом. Конфликта не возникнет, но возможны различные аномалии. Они приводят к скрытому игнорированию правил, заданных системным администратором, создавая при этом видимость успешной работы межсетевого экрана.

В своей работе авторы использовали таксономию аномалий фильтрации, представленную в статье [23].

Внутри одной таблицы доступа возможны следующие аномалии:

- затенение – когда правило с большим приоритетом обрабатывает все пакеты, которые удовлетворяют условиям правила с меньшим приоритетом, и привилегии (разрешение/запрет) у этих правил – различны. В этом случае правило с меньшим приоритетом никогда не будет активировано;
- обобщение – когда множество пакетов, удовлетворяющих условию правила с большим приоритетом, является подмножеством пакетов, удовлетворяющих условию правила с меньшим приоритетом, и привилегии этих правил – различны. Правило с меньшим приоритетом никогда не будет активировано на этом подмножестве;
- корреляция – когда пересечение множеств пакетов, задаваемых условиями двух правил, - не пусто, но в то же время оно не совпадает ни с одним из этих множеств; при этом привилегии этих правил различны. В этом случае правило с меньшим приоритетом никогда не будет активировано на указанном пересечении;
- избыточность – когда пересечение множеств пакетов, задаваемых условиями двух правил, – не пусто, при этом привилегии этих правил совпадают. Возможно, такая ситуация связана с ошибкой администратора, и одно из правил можно удалить или разделить на части.

В таблице представлена таблица доступа, содержащая правила фильтрации с временными характеристиками с примерами всех возможных аномалий. В таблице введены следующие сокращения: П – приоритет правил, Пр – тип используемого протокола, По – порт отправителя, Пп – порт получателя, Пв – привилегия, Ва – время активации, Вд – время деактивации.

Правила с приоритетами 1-6 приводят к возникновению аномалий. Между правилами с приоритетами 1 и 2, а также 1 и 6 существует аномалия затенения. Между правилами 1 и 3, 1 и 5, 2 и 4, а также 3 и 5 есть аномалия избыточности. Пример аномалии корреляции показан на правилах с привилегиями 1 и 4, 4 и 5. Аномалия обобщения показана между правилами 3 и 4.

Представим формальное описание перечисленных аномалий. Согласно описанию метода “проверки на модели”, приведенному

выше, формулами линейной темпоральной логики должны быть выражены корректные состояния моделируемой системы, которые никогда не должны быть нарушены.

Таблица доступа с временными параметрами и примерами аномалий

П	Пр	IP отправителя	По	IP получателя	Пп	Пв	Ва	Вд
1	tcp	140.192.37.20	80	161.120.33.40	any	accept	9:00	18:00
2	tcp	140.192.37.20	80	161.120.33.40	80	deny	9:00	18:00
3	tcp	140.192.37.20	80	161.120.33.*	21	accept	9:00	18:00
4	tcp	140.192.37.20	80	*.*.*.*	any	deny	9:00	18:00
5	tcp	140.192.37.20	any	161.120.33.40	21	accept	9:00	18:00
6	tcp	140.192.37.20	21	161.120.33.40	80	deny	10:00	18:00
7	tcp	140.192.37.20	21	161.120.35.*	21	accept	12:20	14:00
8	tcp	140.192.37.20	21	161.120.35.*	80	deny	9:00	18:00
9	tcp	140.192.37.30	80	161.120.33.40	21	accept	17:00	22:00
10	tcp	140.192.37.30	80	161.120.33.40	80	deny	13:00	23:00
11	tcp	140.192.37.30	80	161.120.35.*	21	accept	11:00	12:00
12	tcp	140.192.37.30	80	161.120.35.*	80	deny	16:00	22:00
13	tcp	140.192.37.30	21	161.120.33.40	21	accept	9:00	18:00
14	tcp	140.192.37.30	21	161.120.33.40	80	deny	9:00	18:00
15	tcp	140.192.37.30	21	161.120.35.*	21	accept	8:00	22:00

Определим для каждой категории аномалии свою формулу.

Формализуем такие состояния системы, нарушение которых приведет к появлению аномалий фильтрации.

Сначала введем обозначения свойств и событий, описывающих такое состояние: t_{a_n} – время активации правила u_n; t_{a_m} – время активации правила u_m; t_{d_n} – время деактивации правила u_n; t_{d_m} – время деактивации правила u_m; o_{u_n} – приоритет правила u_n; o_{u_m} – приоритет правила u_m; l_{u_n} – привилегия правила u_n; l_{u_m} – привилегия правила u_m.

Затем введем следующие логические функции для обозначения состояния системы: $IsSubset(u_n, u_m)$ – адресный диапазон правила u_n является подмножеством диапазона адресов правила

u_m; $IsIntersection(u_n, u_m)$ – адресный диапазон правила u_n является пересечением диапазона адресов правила u_m; $IsEqual(u_n, u_m)$ – адресный диапазон правила u_n совпадает с диапазоном адресов правила u_m.

При формализации аномалий фильтрации используется темпоральный оператор ***G,*** обозначающий, что формальное утверждение должно выполняться в настоящее время и всегда в будущем.

Представим формальное описание аномалий:

(1) аномалия затенения:

$$f_s = G((t_{a_n} \le t_{a_m}) \wedge (t_{a_m} < t_{d_n}) \wedge (o_{u_m} > o_{u_n}) \wedge \wedge IsSubset(u_m, u_n) \Rightarrow (l_{u_n} \neg = l_{u_m}));$$

(2) аномалия обобщения:

$$f_g = G((t_{a_n} \le t_{a_m}) \wedge (t_{a_m} < t_{d_n}) \wedge (o_{u_m} > o_{u_n}) \wedge \wedge IsSubset(u_n, u_m) \Rightarrow (l_{u_n} \neg = l_{u_m}));$$

(3) аномалия корреляции:

$$f_c = G((t_{a_n} \le t_{a_m}) \wedge (t_{a_m} < t_{d_n}) \wedge (o_{u_m} > o_{u_n}) \wedge \wedge IsInter\,sec\,tion(u_n, u_m) \wedge \neg(IsEqual(u_n, u_m) \Rightarrow (l_{u_n} \neg = l_{u_m}));$$

(4) аномалия избыточности:

$$f_r = G((t_{a_n} \le t_{a_m}) \wedge (t_{a_m} < t_{d_n}) \wedge (o_{u_m} > o_{u_n}) \wedge \wedge IsIntersection(u_n, u_m) \Rightarrow (l_{u_n} = l_{u_m})).$$

Итак, задача верификации компьютерной системы, защищенной межсетевым экраном, на предмет аномалий фильтрации представляет собой проверку выполнимости формул f_s, f_g, f_c, f_r на модели Крипке M_N.

4. Методика верификации правил фильтрации методом "проверки на модели". Основные этапы предлагаемой методики верификации правил фильтрации методом "проверки на модели" представлены на рис. 3 [24].

Входными данными предлагаемой методики являются описания правил фильтрации политики на языке описания политики (ЯОП) и конфигурации компьютерной сети на языке описания системы (ЯОС), а также выявляемые аномалии фильтрации. ЯОП и

ЯОС – это XML-подобные языки, созданные на основе Common Information Model (CIM) [25].

На первом шаге методики входные данные, включающие в себя описание политики, компьютерной системы и аномалий, преобразуются во внутренний формат системы верификации.

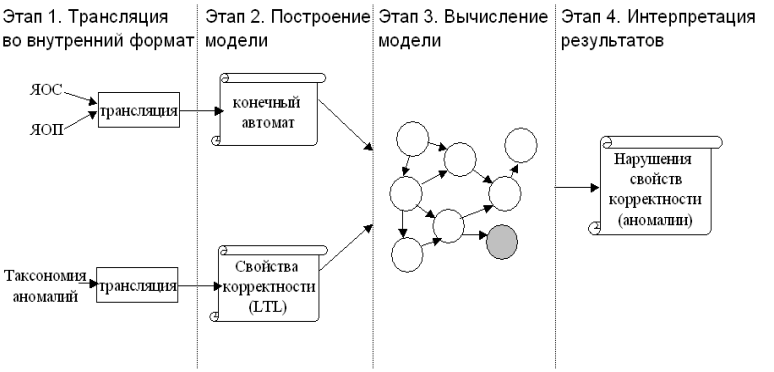

Рис.3. Основные этапы методики верификации правил фильтрации

Затем строится общая модель для верификации правил фильтрации, представленная в виде конечного автомата. Эта модель включает в себя модели компьютерной системы, межсетевого экрана и аномалий. Она инициализируется входными данными во внутреннем формате.

Данная модель должна обеспечивать: отображение топологии сети, генерацию необходимого набора значений сетевых адресов, позволяющего обработать все задаваемые на входе правила; передачу сетевого трафика по заданному диапазону адресов; обработку сетевого трафика межсетевыми экранами с накоплением статистики по применяемым правилам; анализ статистики работы межсетевых экранов на предмет существующих аномалий.

Аномалии в модели выражены формальными утверждениями. Для метода “проверки на модели” эти формальные утверждения будут являться свойствами корректности, нарушение которых приводит исследуемую систему в некорректное состояние.

На третьем этапе общая модель для верификации правил фильтрации верифицируется специальными программными средствами, реализующими метод “проверки на модели”, например, SPIN [2, 6], SMV [26], MOCHA [27] и т.п. В процессе верификации

выявляются все некорректные состояния системы. Серым цветом на рисунке 3 помечено одно из таких некорректных состояний.

На завершающем этапе полученные результаты верификации интерпретируются. Если были обнаружены аномалии, то пользователь получает адреса межсетевых экранов и правила, приводящие к возникновению аномалий, а также тип аномалий.

Для реализации методики верификации в работе использовалось программное средство SPIN.

Пример схемы реализации обнаружения аномалий в правилах фильтрации представлен на рис. 4.

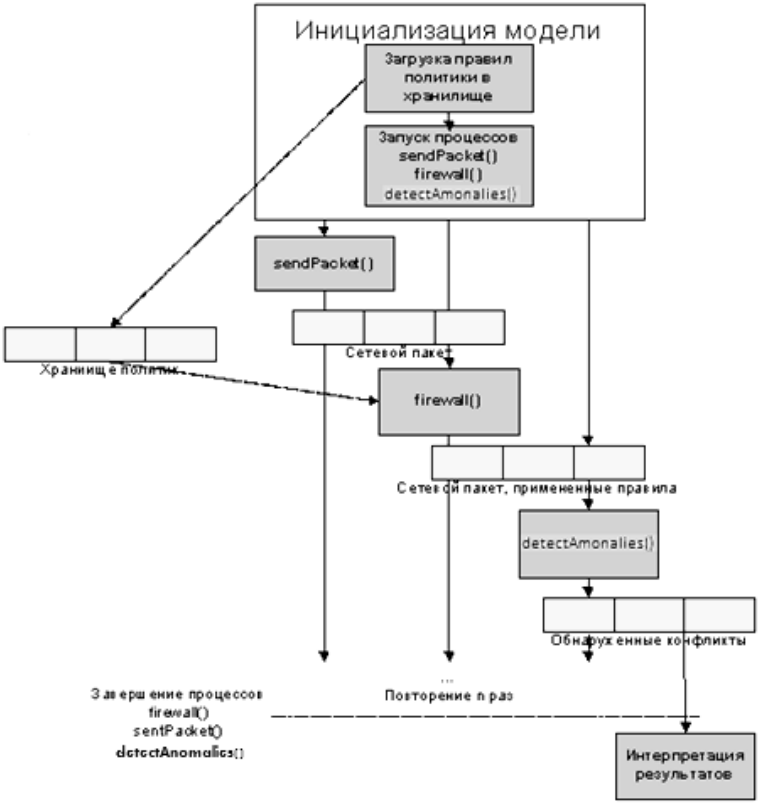

Рис. 4. Пример схемы реализации обнаружения аномалий в правилах фильтрации

В SPIN внутренним языком является Promela. Основные сущности, такие как правило политики, сетевой пакет, конфликт, отображаются в виде структур данных на языке Promela.

Приведем пример представления некоторых структур данных на языке Promela.

Представление правила фильтрации имеет следующий вид:

```
typedef FilteringRule {
  bool action = true; /*true – allow, false – deny*/
  mtype ruleName;
  mtype = {Ipv4, tcp, udp};
  chan trafficType = [1] of {mtype};
  Address sourceAddress;
  Port sourcePort;
  Address destinationAddress;
  Port destinationPort;
  mtype = {Enabled, Disabled};
  chan en = [1] of {mtype};
  short prior;
}
```

Эта структура данных позволяет хранить в себе действие, которое задает правило для сетевых пакетов (allow, deny), имя правила фильтрации, тип протокола, адрес и порт источника, а также адрес и порт получателя, статус правила (enabled, disabled) и приоритет правила в таблице доступа.

Описание политики безопасности задается следующим образом:

```
typedef Policy {
  mtype policyName;
  mtype = {filteringPolicy};
  chan policyType = [1] of {mtype};
  chan filtreringRules = [10] of {FilteringRule};
}
```

Структура политики безопасности Policy включает в себя название политики, тип политики (авторизация, фильтрация, аутентификация или операционные правила) и набор правил политики.

Ниже представлен код для описания хранилища политики:

```
/*Security policy storage*/
typedef Storage {
  chan policy = [50] of {Policy};
}
```

Описание хранилища политики включает в себя одну из базовых структур Promela – буферизированный канал сообщений, представляющий собой аналог массива для хранения политик.

Хранилище правил также представлено буферизированным каналом сообщений:

```
chan policyRules = [50] of {FilteringRule}; /*policyRules*/
```

Обмен данными между процессами в языке Promela также осуществляется через каналы сообщений.

Ниже описаны разделяемые между различными процессами программы ресурсы:

Структура storage предназначена для хранения политик:

```
/*SHARED DATA*/
Storage storage; /*policy storage*/
```

Через канал сообщений sentPackets процесс генерации пакетов и процесс межсетевого экрана обмениваются сетевыми пакетами:

```
chan sentPackets = [1] of {NetPacket}; /*sent packets channel*/
```

Канал для хранения правил фильтрации, примененных к сетевому пакету, и сам пакет задается следующим образом:

```
chan affectedRulePacket = [100] of { FilteringRule, NetPacket};
/*applied rules channel*/
```

Канал используется процессами межсетевого экрана и обнаружения аномалий.

Для формул линейной темпоральной логики в языке Promela используется специальное ключевое слово assert (формальное утверждение). В процессе верификации любое нарушение этих формальных утверждений будет выявлено, и построен контрпример, показывающий, как система пришла в некорректное состояние. Таким образом, все правила, образующие аномалии будут выявлены.

Представим фрагмент процесса инициализации верифицируемой модели:

```
/**********MODEL INITIALIZATION**********/
mtype = {filtering_rule_1, filtering_rule_2, filtering_rule_3, filtering_rule_4,
        filtering_rule_5, filtering_rule_6, filtering_rule_7, filtering_rule_8,
        filtering_rule_9, filtering_rule_10};
mtype = {upstream, downstream};
```

```
proctype initModel () {

/***Creation rules***/
  FilteringRule fRule1;
  FilteringRule fRule2;
/***Init rules***/
  fRule1.ruleName = filtering_rule_1;
  fRule1.trafficType! tcp;
  fRule1.sourceAddress.octet1=140;
  fRule1.sourceAddress.octet2=192;
  fRule1.sourceAddress.octet3=37;
  fRule1.sourceAddress.octet4=20;
  fRule1.sourcePort.portNum=80;
  fRule1.destinationAddress.octet1=161;
  fRule1.destinationAddress.octet2=120;
  fRule1.destinationAddress.octet3=33;
  fRule1.destinationAddress.octet4=40;
  fRule1.destinationPort.portNum=21;
  fRule1.action = true;
  fRule1.en!Enabled;
  fRule1.prior = 1;
  fRule1.firewallId = 1;
/***Create firewalls***/
  Firewall firewall1;
  Firewall firewall2;
  firewall1.firewallName = upstream;
  firewall1.firewallId = 1;
  firewall1.firewallOrder = 1;
  firewall2.firewallName = downstream;
  firewall2.firewallId = 2;
  firewall1.firewallOrder = 2;

  firewalls!firewall1;
  firewalls!firewall2;
  HIERARCHYLEVEL=2

/***Init policy storage***/
  storage.policyRules!fRule1;
  storage.policyRules!fRule2;
```

При инициализации задается политика фильтрации, создаются правила таблицы доступа, происходит загрузка этих данных в хранилище. Далее запускаются основные процессы, такие как процесс генерации трафика (sendNetPacket), процесс межсетевого экрана (firewall) и процесс обнаружения аномалий (detectFilteringConflicts):

```
proctype startProcesses() {
  run generatePackets();
  run sendNetPacket();
```

```
  run firewall();
  run detectFilteringConflicts();
}
```

Код ниже показывает запуск верификации модели:

```
init {
  atom {  run initModel(); }
  run startProcesses();
}
```

После того, как будет обработан весь диапазон сетевых адресов, верификация завершается. Верификатор выдает все нарушения формальных утверждений, т. е. обнаруженные аномалии фильтрации.

Результаты проведенных экспериментов представлены на рис. 5. Вычисления производились на 12 выборках, в которых изменялось число правил и число аномалий. Число правил увеличивалось с 5 до 50, а число аномалий с 0 до 15. По результатам экспериментов все аномалии были обнаружены.

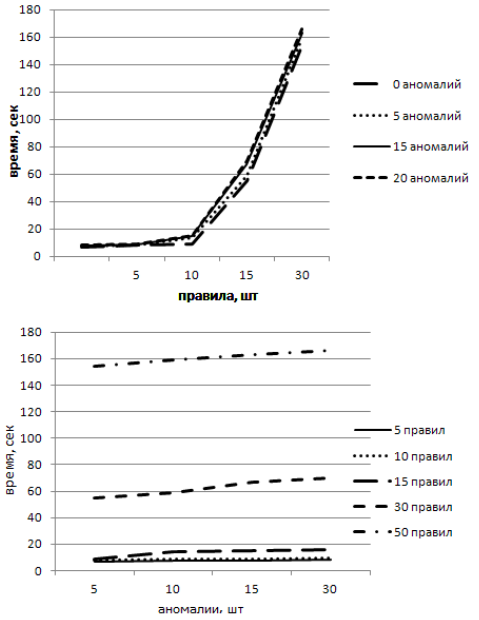

Рис. 5. Результаты экспериментов обнаружения аномалий фильтрации

Согласно полученным данным время поиска аномалий в зависимости от числа правил растет экспоненциально. Также выявлена линейная зависимость времени поиска от количества аномалий в правилах.

Проведенные эксперименты показали, что предлагаемый подход позволяет выявлять все аномалии правил фильтрации межсетевых экранов, но может эффективно использоваться до определенного количества правил. Введение временных характеристик в правила фильтрации не изменили существенно времени верификации.

В ходе дальнейших экспериментов планируется уточнить данное ограничение. Преимуществами данного подхода являются его высокий уровень абстракции при представлении данных и возможность исследовать динамическое поведение системы. К недостаткам относится вычислительная сложность подхода.

5. Заключение, В настоящей статье предложен общий подход к верификации правил фильтрации межсетевого экрана, модели компьютерной сети, межсетевого экрана и аномалий фильтрации, а также методика верификации правил фильтрации методом “проверки на модели”.

Методика верификации правил фильтрации реализована с использованием программного средства SPIN. Представлены результаты проведенных экспериментов.

В предлагаемом подходе к верификации методом “проверки на модели” осуществляется моделирование передачи ограниченного набора адресов и их обработка с помощью модели межсетевого экрана. Состояние системы характеризуется накопленной статистикой по обработке сетевого трафика. Статистика анализируется на предмет аномалий фильтрации, заданных формальными утверждениями.

По результатам проведенных экспериментов было показано, что предлагаемый подход позволяет выявлять все аномалии в правилах фильтрации политики безопасности, однако имеет экспоненциальную вычислительную сложность в зависимости от количества верифицируемых правил.

Для оценки эффективности реализации метода верификации были проведены тесты на различных наборах правил фильтрации с вычислением различных метрик, таких как время вычисления, полнота функциональности и потребление ресурсов. Тесты показали, что система выполняет обнаружение аномалий фильтрации в приемлемый период времени с использованием

заданных ресурсов. Таким образом, можно заключить, что предложенная методика может быть применима для малых и средних компьютерных сетей.

В настоящее время продолжается реализация компонентов верификации, служащих для выявления внешних аномалий при работе сети с несколькими межсетевыми экранами. Кроме того, авторы планируют расширить верифицируемую модель другими категориями политик безопасности, такими как авторизация и защита каналов. Также планируется дальнейшее совершенствование системы верификации на основе разработки интерфейса загрузки первоначальных данных и реализации механизмов интерпретации результатов с использованием графических средств.

Литература

1. *Baier C., Katoen J.-P.* Principles of Model Checking // The MIT Press, 2008. 984 p.
2. *Holzmann G.* The Spin Model Checker Primer and Reference Manual // Addison-Wesley, 2003. 608 p.
3. *Manna Z., Pnueli A.* Temporal Verification of Reactive Systems: Safety // Springer-Verlag, New York, 1995. 530 p.
4. *Карпов Ю.Г.* Model checking. Верификация параллельных и распределенных программных систем // СПб.: БХВ, 2009. 560 с.
5. *Миронов А.М.* Математическая теория программных систем. http://intsys.msu.ru/staff/mironov/
6. On-The-Fly, LTL Model Checking with SPIN. http://netlib.bell-labs.com/netlib/spin/whatispin.html
7. *Тишков А.В., Котенко И.В., Черватюк О.В., Лакомов Д.П., Резник С.А., Сидельникова Е.В.* Обнаружение и разрешение конфликтов в политиках безопасности компьютерных сетей // Труды СПИИРАН, Выпуск 3, Том 2. СПб.: Наука, 2006. С.102-114.
8. *Котенко И.В., Тишков А.В., Черватюк О.В., Лакомов Д.П.* Поиск конфликтов в политиках безопасности // Изв. Вузов. Приборостроение, Т.49, № 11, 2006, С.45-49.
9. *Котенко И.В., Тишков А.В., Черватюк О.В., Резник С.А., Сидельникова Е.В.* Система верификации политики безопасности компьютерной сети // Вестник компьютерных и информационных технологий, № 11, 2007. С.48-56.
10. *Котенко И.В., Тишков А.В., Сидельникова Е.В., Черватюк О.В.* Проверка правил политики безопасности для корпоративных компьютерных сетей // Защита информации. Инсайд, № 5, 2007. С.46-49; № 6, 2007. С.52-59.
11. *Черватюк О.В., Котенко И.В.* Верификация правил фильтрации политики безопасности методом “проверки на модели” // Изв. Вузов. Приборостроение, Т.51, № 12, 2008, С.44-49. ISSN 0021-3454.
12. *Braghin C., Sharygina N., Barone-Adesi K.* A Model Checking-based Approach for Security Policy Verification of Mobile Systems // Formal Aspects of Computing Journal, Springer, 2010. Vol. 23, No. 5, P.627-648.
13. *Pham H.T., Truong N.-T., Nguyen V.-H.* Analyzing RBAC Security Policy of Implementation Using AST // Proceedings of the 2009 International Conference on Knowledge and Systems Engineering (KSE ‘09), 2009. P. 215-219.

14. *Jajodia S., Samarati P., Sapino M.L., Subrahmanian V.S.* Flexible support for multiple access control policies // ACM Transaction Database Systems. Vol.26, No.2, 2001. P.214-260.
15. *Zhang N., Ryan M. D. Guelev D.* Evaluating Access Control Policies Through Model Checking // The 8th Information Security Conference (ISC'05). Lecture Notes in Computer Science. Springer-Verlag. Vol. 3650, 2005. P.446–460.
16. *Bandara A.K., Kakas A.S., Lupu E.C., Russo A.* Using Argumentation Logic for Firewall Policy Specification and Analysis // 17th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management. DSOM 2006, 2006. P.185-196.
17. *Al-Shaer E., Hamed H., Boutaba R., Hasan M.* Conflict classification and analysis of distributed firewall policies // IEEE Journal on Selected Areas in Communications, Vol.23, No.10, 2005. P.2069-2084.
18. *Sirjani M.* Formal specification and verification of concurrent and reactive systems // Ph.D. Thesis. Sharif University of Technology, 2004. 175 p.
19. *Rothmaier G., Kneiphoff T., Krumm H.* Using SPIN and Eclipse for Optimized High-Level Modeling and Analysis of Computer Network Attack Models // Lecture Notes in Computer Science, Springer-Verlag, 2005. Vol. 3639. P.236–250.
20. *Conrad S., Campbell L.A., Cheng B.H.C., Deng M.* A requirements patterns-driven approach to specify systems and check properties // 18th International SPIN Workshop, Snowbird, UT, USA. Lecture Notes in Computer Science, Vol. 2648. Springer-Verlag, 2003. P.18-33.
21. *Benthem J.M., Meulen A.T. (Eds.)* Handbook of Logic and Language // North-Holland, 1997.
22. *Westerinen A., Strassner J., Scherling M., Quinn B., Herzog S., Huynh A., Carlson M., Perry J., Waldbusser S.* Terminology for Policy-Based Management (RFC 3198). http://www.rfc-archive.org/getrfc.php?rfc=3198.
23. *Hamed H., Al-Shaer E.* Taxonomy of Conflicts in Network Security Policies // IEEE Communications Magazine, March 2006, Vol. 44, No. 3. P.134-141.
24. *Kotenko I., Polubelova O.* Verification of Security Policy Filtering Rules by Model Checking // Proceedings of IEEE Fourth International Workshop on "Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications" (IDAACS'2011). Prague, Czech Republic, 15-17 September 2011. P. 706-710.
25. Common Information Model (CIM) Standards. http://www.dmtf.org/standards/cim
26. *McMillan K.* The SMV System. http://www.cs.cmu.edu/_modelcheck/smv.html
27. *Alur R., Anand H., Grosu R., Ivancic F., etc.* Mocha User Manual. Jmocha Version 2.0. http://embedded.eecs.berkeley.edu/research/mocha/doc/j-doc/.
28. *Кларк Э.М., Грамберг О., Пелед Д.* Верификация моделей программ. Model Checking. М.: МЦНМО. 2002.

Полубелова Ольга Витальевна — младший научный сотрудник лаборатории проблем компьютерной безопасности СПИИРАН. Область научных интересов: безопасность компьютерных сетей, включая управление политиками безопасности, верификация протоколов безопасности и систем безопасности, использование методов проверки на модели для обнаружения и разрешения конфликтов в политиках; онтологии в информационной безопасности, дескрипционные логики, СИЕМ-системы. Число научных публикаций — 25. ovp@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450. Научный руководитель — Котенко И.В.

Polubelova Olga Vitalievna — junior researcher of Laboratory of Computer Security Problems, SPIIRAS. Research interests: computer network security, including policy management, verification of security protocols and security systems, model checking techniques for policy conflicts detection and resolution, ontology, description logic, SIEM systems. The number of publications — 25. ovp@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450. Scientific leader — I.V. Kotenko.

Котенко Игорь Витальевич — д.т.н., проф.; заведующий лабораторией проблем компьютерной безопасности СПИИРАН. Область научных интересов: безопасность компьютерных сетей, в том числе управление политиками безопасности, разграничение доступа, аутентификация, анализ защищенности, обнаружение компьютерных атак, межсетевые экраны, защита от вирусов и сетевых червей, анализ и верификация протоколов безопасности и систем защиты информации, защита программного обеспечения от взлома и управление цифровыми правами, технологии моделирования и визуализации для противодействия кибер-терроризму, искусственный интеллект, в том числе многоагентные системы, мягкие и эволюционные вычисления, машинное обучение, извлечение знаний, анализ и объединение данных, интеллектуальные системы поддержки принятия решений, телекоммуникационные системы, в том числе поддержка принятия решений и планирование для систем связи. Число научных публикаций — более 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; СПИИРАН, 14-я линия В.О., д.39, Санкт-Петербург, 199178, РФ; р.т. +7(812)328-2642, факс +7(812)328-4450.

Kotenko Igor Vitalievich — Prof. of Computer Science; head of Laboratory of Computer Security Problems, SPIIRAS. Research interests: computer network security, including security policy management, access control, authentication, network security analysis, intrusion detection, firewalls, deception systems, malware protection, verification of security systems, digital right management, modeling, simulation and visualization technologies for counteraction to cyber terrorism, artificial intelligence, including multi-agent frameworks and systems, agent-based modeling and simulation, soft and evolutionary computing, machine learning, data mining, data and information fusion, telecommunications, including decision making and planning for telecommunication systems. The number of publications — 450. ivkote@comsec.spb.ru, www.comsec.spb.ru; SPIIRAS, 39, 14-th Line V.O., St. Petersburg, 199178, Russia; office phone +7(812) 328-2642, fax +7(812)328-4450.

Поддержка исследований. Работа выполняется при финансовой поддержке Министерства образования и науки Российской Федерации (государственный контракт 11.519.11.4008), РФФИ (проект №10-01-00826-а), программы фундаментальных исследований ОНИТ РАН (проект №2.2), проектов Евросоюза SecFutur и MASSIF и ряда других проектов.

Рекомендовано СПИИРАН, лабораторией проблем компьютерной безопасности, заведующий лабораторией Котенко И.В., д-р техн. наук, проф.
Статья поступила в редакцию 3.04.2012.

РЕФЕРАТ

Полубелова О.В., Котенко И.В. **Верификация правил фильтрации с временными характеристиками методом “проверки на модели”.**

В работе предлагается подход к верификации правил фильтрации, предназначенный для обнаружения и разрешения аномалий фильтрации в спецификациях политики безопасности компьютерных сетей.

Задача верификации правил фильтрации межсетевых экранов актуальна для любой системы управления корпоративной компьютерной сетью на основе политик безопасности. Формирование набора правил фильтрации, не содержащих аномалий, позволяет обеспечить работоспособность и безопасность компьютерных сетей. По мере увеличения числа правил фильтрации задача их верификации становится трудновыполнимой без автоматизации. Предлагаемые в работе модели, алгоритмы и методика верификации базируются на использовании метода “проверки на модели” (model checking).

В работе рассмотрены модели компьютерной системы, межсетевого экрана и аномалий фильтрации, служащие для верификации правил фильтрации, а также алгоритмы выявления таких аномалий.

Предложена методика верификации правил фильтрации политик безопасности на базе этих моделей. Основными входными данными в методике являются правила фильтрации политики и конфигурации компьютерной сети, а также выявляемые аномалии фильтрации. Аномалии в модели выражены формальными утверждениями. Для метода проверки на модели эти формальные утверждения будут являться свойствами, нарушение которых приводит исследуемую систему в некорректное состояние. Специфика данной работы в области верификации правил фильтрации состоит в том, что возможна верификация правил с временными параметрами, которые задают ограничения на прохождение трафика по сети с учетом времени суток, дня недели и т. д.

В реализации предложенной методики верификации с заданием временных параметров была использована утилита ipTables, которая является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) netfilter для ядер Linux.

Для анализа эффективности предложенной методики была проведена серия экспериментов, в которых поиск аномалий осуществлялся в одной таблице доступа одного межсетевого экрана для двух сегментов сети. Проведенные эксперименты показали, что предлагаемый подход позволяет выявлять все аномалии правил фильтрации межсетевых экранов, но может эффективно использоваться до определенного количества правил. В ходе дальнейших экспериментов планируется уточнить данное ограничение. Преимуществами данного подхода являются его высокий уровень абстракции при представлении данных и возможность исследовать динамическое поведение системы. К недостаткам относится вычислительная сложность подхода.

SUMMARY

Polubelova O.V., Kotenko I.V. **Verification of security policy filtering rules with temporal parameters by Model Checking.**

The paper outlines an approach to verification of filtering rules of firewalls. The approach is intended for detection and resolution of filtering anomalies in the specification of security policy of computer networks.

One of the very important tasks a computer network administrator has to fulfill under constructing a (distributed) firewall security policy is to guarantee the absence of inconsistencies (or anomalies) and possibility to implement the policy in the given network configuration. Verification of firewall filtering rules is an actual problem for any management system of corporative computer network based on security policies.

The paper proposes the models of computer networks, the models of firewalls and filtering anomalies, as well as an algorithm of detection of such anomalies. The main peculiarities of the approach consist in using Model Checking exactly to detect the anomalies of filtering rules and in ability to specify temporal parameters in filtering rules.

We suggest also a method for verification of filtering rules based on the mentioned models. This model should provide: mapping of the computer network; generating the set of network address; network traffic in the defined range of network addresses; network data processing with accumulation of statistics; network statistics analyzed in order to detect anomalies. Generated network addresses should cover the complete range of network addresses defined in the rules. Anomalies in the model are expressed as assertions. Their violations lead to an incorrect state. The peculiarity of this work is the ability to use temporal logic to specify temporal parameters in filtering rules which set constrains to network traffic taking into account time, day of week, etc.

Implementation of the proposed method of filtering rules verification with temporal parameters uses the utility named ipTables. It is a standard interface of firewall management for Linux cores.

To analyze efficiency of verification modules we executed several experiments. In the experiments the inconsistencies were searched for filtering anomalies inside the access table of single firewall separating two network segments. The results of experiments, carried out, show that the proposed approach lets to detect all possible anomalies in filtering rules. But the approach has complexity exponentially dependent on number of verified rules.

In order to analyze the effectiveness of implementation of the method a number of tests using different metrics have been performed. The metrics used are the duration of computation, completeness of functionality, consumption of resources. Thus, we can conclude that the method proposed can be applicable for small and medium-sized computer networks. The advantage of this approach is a high level of abstraction in the presentation of data. This approach also let us to investigate the dynamic behavior of the system. The disadvantage is the computational complexity of the approach.