МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра общей информатики

Направление подготовки: 230100 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ БАКАЛАВРСКАЯ РАБОТА

ПРИМЕНЕНИЕ МЕТОДОЛОГИИ АНАЛИЗА ФОРМАЛЬНЫХ ПОНЯТИЙ ДЛЯ АВТОМАТИЧЕСКОГО ИЗВЛЕЧЕНИЯ ПРИЧИННО-СЛЕДСТВЕННЫХ СВЯЗЕЙ ИЗ ПРЕЦЕДЕНТОВ КОМПЬЮТЕРНЫХ АТАК

Буркова Надежда Сергеевна

«К защите допущена»

Зав. Кафедрой

Пальчунов Д.Е.,

д.ф.-м.н., доцент

……/……..

(фамилия , И., О.) / (подпись, МП)

«……»……20…г.

Научный руководитель

Яхъяева Г.Э.,

к.ф.-м.н., доцент НГУ

……/……..

(фамилия , И., О.) / (подпись, МП)

«……»……20…г.

Дата защиты: «……»……20…г.

Автор……/……..

(фамилия , И., О.) / (подпись)

Новосибирск, 2014г.

МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ, НГУ)

Факультет информационных технологий

Кафедра общей информатики

Направление подготовки: 230100 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

УТВЕРЖДАЮ

Зав. кафедрой Пальчунов Д.Е.

…….

(подпись, МП)

«……»……20…г.

ЗАДАНИЕ

НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ БАКАЛАВРСКУЮ РАБОТУ

Студентке Бурковой Надежде Сергеевне

Тема: Применение методологии анализа формальных понятий для автоматического извлечения причинно-следственных связей из прецедентов компьютерных атак

Цель работы: разработка модуля автоматического извлечения причинно-следственных связей из разных прецедентов компьютерных атак

Структурные части работы: работа включает обзор языков описания онтологий, обзор существующих алгоритмов нахождения базиса импликаций, описание функциональности разработанного модуля.

Научный руководитель
Яхъяева Г.Э.,
к.ф.-м.н., доцент

……/……..
(фамилия , И., О.) / (подпись)
«…»……20…г.

Задание принял к исполнению
Буркова Н.С./……..
(ФИО студента) / (подпись)

«…»……20…г.

Содержание

ВВЕДЕНИЕ

В настоящее время роль компьютера в жизни каждого человека возрастает. Информационные технологии давно занимают важное место в повседневной жизни человека как на работе, так и дома, практически у всех уже есть доступ в Интернет. Вместе с растущим числом пользователей компьютеров постоянно возрастает число угроз, подстерегающих человека при работе с компьютером, например, вирусы, черви, трояны, из-за которых пользователь может потерять важную информацию, деньги с аккаунтов, работоспособность компьютера может быть снижена, также злоумышленники могут использовать компьютер пользователя и его доступ в Интернет для заражения других машин и сокрытия своего истинного местоположения, создания ботнет-сети и рассылки спама. Практически каждому приходилось сталкиваться с угрозами информационной безопасности. По оценкам некоторых исследований ежесекундно в мире жертвами злоумышленников становятся 12 человек, и эта цифра с каждым годом возрастает [12]. Для отдельного пользователя последствия могут быть не очень критичными, если, конечно, злоумышленники не получили доступ к его конфиденциальным данным, как пин-код к кредитке, но для крупных компаний ущерб обычно бывает куда более значительным. Многие компании несут серьезные убытки вследствие киберпреступлений (кражи ключей, ценных данных, Ddos-атаки, итд). В России за 2012 год более 31.4 миллионов человек стали жертвами киберпреступников. В денежном выражении ущерб составил порядка 2 миллиардов долларов США [13, 14]. Компьютерные преступления в России с каждым годом совершаются все чаще, по оценке МВД за 2013-й их число увеличилось на 8,6% [12]. Именно поэтому необходимо предотвращать прецеденты компьютерных атак как можно раньше, а для этого необходимо анализировать большое количество информации о прецедентах и об их признаках, чтобы была возможность быстро отреагировать на проблему и принять правильные контрмеры.

Применение онтологий в качестве компонента информационной системы, которая нацелена на решение проблемы с использованием знаний, в настоящее время является одним из самых перспективных методов. Онтология – это формализация некоторой предметной области. Она позволяет организовать знания, то есть определить понятия, отношения, описывающие задачи в выбранной предметной области, также возможно детализировать описание так глубоко, как это необходимо. Онтология, содержащая еще и

экземпляры объектов, является ничем иным, как базой знаний, и может служить для представления иерархии понятий и их отношений

Данная работа посвящена описанию модуля, который автоматически извлекает связи между признаками прецедентов компьютерных атак. Модуль строит набор импликаций, описывающих закономерности предметной области. Для достижения этой цели используется анализ формальных понятий, являющийся мощным инструментом для представления знаний.

ГЛАВА 1 Цель и задачи дипломной работы

Цель данной дипломной работы – разработка модуля автоматического извлечения причинно-следственных связей из разных прецедентов компьютерных атак.

Задачи дипломной работы:

- провести анализ существующих языков описания онтологий и выбрать наиболее подходящий для работы;

- построить онтологию предметной области компьютерной безопасности на выбранном языке;

- заполнить базу знаний прецедентами, используя данные из базы, реализованной в системе RiskPanel [8, 9];

- провести анализ существующих алгоритмов построения базиса импликаций и выбрать наиболее подходящий;

- программно реализовать модуль извлечения закономерностей между признаками прецедентов компьютерных атак. К модулю предъявляются следующие функциональные требования: пользователь должен иметь возможность выбрать категории признаков, которые интересны ему в качестве посылки и заключения импликаций, в результате он получает импликации, отфильтрованные по выбранным им категориям.

ГЛАВА 2 Обзор предметной области

2.1 Описание предметной области информационной безопасности

Как известно, прецедент – это случай или событие, имевшее место в прошлом, и служащее примером или основанием для аналогичных действий в настоящем. Однако, в данной работе под прецедентом понимается модель некоторой компьютерной атаки. Каждый прецедент компьютерной атаки описывается алгебраической системой $\mathfrak{A} = \langle A, \sigma \rangle$, где A – основное множество алгебраической системы, а σ – её сигнатура. Сигнатура σ – это множество понятий, с их помощью описывается предметная область информационной безопасности. Данная сигнатура имеет следующие понятия: множества различных уязвимостей, угроз, контрмер, последствий, потерь, симптомов [9].

Каждый прецедент может описываться следующими признаками:

1) Симптомы – какие-либо происшествия, изменения в системе, встречающихся при тех или иных атаках
2) Угрозы – различные ситуации и сбои в работе, происходящие во время атаки вируса или злоумышленника
3) Уязвимости – недостатки и слабые места в системе, использование которых может позволить злоумышленнику получить доступ к компьютеру, нарушить целостность данных и вызвать неправильную работу системы
4) Последствия – последствия для системы и пользователя, которые может нанести та или иная атака
5) Возможные потери – потери информации, контроля над системой, которые могут произойти вследствие атаки
6) Контрмеры – возможные решения проблемы, например, обновление вирусных баз, шифрование данных и т.д.

2.2 Анализ формальных понятий

Анализ формальных понятий (АФП) дает возможность построить полную решётку по любому бинарному отношению и формализовать описания понятия в виде пары <объём, содержание> (Здесь объем – некоторое множество объектов, а содержание – общие для них признаки). Решетки формальных понятий основываются на соответствии Галуа, задаваемом на множестве объектов и признаков и определяющим известную из

философии связь между объемом и содержанием понятия: с ростом содержания объем уменьшается. [2]

Определение. ***Формальный контекст*** *K = (G, M, I)* состоит из двух множеств *G* и *M* и $I \subset G \times M$ – отношения обладания признаком. Элементы множества *G* называются ***(формальными) объектами***, а элементы множества *M* – ***(формальными) признаками***.

G / M	m_1	m_2	m_3	m_4
g_1	×	×		×
g_2		×	×	×
g_3			×	×

Пример 1. Формальный контекст удобно представлять в виде объектно-признаковой таблицы. При этом запись *gIm* читается как «объект *g* обладает признаком *m*» [15].

В определении формального понятия используются операторы Галуа.

Определение. ***Операторы Галуа:***

Пусть A⊆G и B⊆M, тогда:

A' = {m∈M | ∀ g ∈ A : gIm}, т.е A' – множество признаков, которыми обладают все объекты из множества A.

B' = {g∈G | ∀ m ∈ M : gIm}, т.е. B' – множество объектов, которые обладают всеми признаками из множества B.

Пример 2. Для контекста из примера 1 $\{g_1, g_2\}' = \{m_2, m_4\}$, а $\{m_2, m_3\}' = \{g_2\}$ [15].

Определение. ***Формальным понятием*** контекста *K = (G, M, I)* называется пара вида *(A, B)*, где $A \subset G$, $B \subset M$, $A' = B$ и $B' = A$. Множество *A* называется ***объемом*** понятия, а множество *B* его ***содержанием***.

Пример 3. Для контекста из примера 1 пара ($\{g_1, g_2\}$, $\{m_2, m_4\}$) является формальным понятием, а пара ($\{m_2, m_3\}$ $\{g_2\}$ нет [15].

Определение. Множество всех понятий контекста $K = (G, M, I)$, упорядоченных по вложению объемов, т.е. $(A_1, B_1) \geq (A_2, B_2) \leftrightarrow A_2 \subseteq A_1$, обозначается $B(G, M, I)$ и называется ***решеткой понятий***.

Множество всех понятий формального контекста образует полную решетку [1].

Оператор $(\cdot)''$ является ***оператором замыкания*** (двукратное применение оператора $'$), т.е. он идемпотентен ($X'''' = X''$), экстенсивен ($X \subseteq X''$) и монотонен ($X \subseteq Y \Rightarrow X'' \subseteq Y''$). Множества $A \subseteq G, B \subseteq M$ называются замкнутыми, если $A'' = A$ и $B'' = B$. Объемы и содержания замкнуты. Множество всех замкнутых

множеств относительно данного оператора замыкания называется системой замыканий [1].

Определение. ***Признаковая импликация*** – для контекста $K = (G, M, I)$ признаковая зависимость $A \rightarrow B$ $(A, B \subseteq M)$ называется ***признаковой импликацией*** тогда и только тогда, когда $A' \subseteq B'$ (т.е. все объекты, обладающие всеми признаками из A, обладают всеми признаками из B). Аналогичным образом определяются объектные импликации.

Формально: $A \subseteq \{g\}'$ означает, что $B \subseteq \{g\}'$ для любого $g \in G$.

Пример 4:

K	Жидкий/изменчивый	Сухой	Влажный	Теплый	Холодный
Земля		X			X
Вода	X		X		X
Воздух	X		X	X	
Огонь	X	X		X	

{влажный} → {жидкий/изменчивый};

{жидкий/изменчивый, сухой} → {теплый};

{сухой, влажный} → {холодный}.

Импликации формального контекста удовлетворяют аксиомам Армстронга [1]:

1. если $Y \subseteq X$, то $X \rightarrow Y$ – рефлексивность;
2. если $X \rightarrow Y$, то $X \cup Z \rightarrow Y$ - пополнение;
3. если $X \rightarrow Y$ и $Y \cup Z \rightarrow W$, то $X \cup Z \rightarrow W$ - транзитивность .

Определение. ***Базис импликаций*** – набор импликаций, из которого все остальные импликации выводятся по правилам вывода Армстронга.

Импликативные признаковые зависимости помогают лучше понимать суть скрытых в данных закономерностей.

Понятие зависимости между признаками основано на следующей идее: если для всех объектов контекста $K = (G, M, I)$, для которых является справедливым некоторое свойство A, некоторое свойство B тоже будет справедливым, то является истинной импликация $A \rightarrow B$. Более точно, импликация $A \rightarrow B$ верна для контекста $K = (G, M, I)$, где $A \subseteq M$ и $B \subseteq M$, если для $g \in G$ выполняется следующее условие: если каждый признак из

посылки импликации A может применяться к объекту g, то каждый признак из заключения импликации B тоже может применяться к g.

Мы хотим извлечь «импликативные» знания из формального контекста. Можно воспользоваться «наивным» подходом: перенумеровать все $2^{2|M|}$ импликации и сверить с контекстом. Однако это займет слишком много времени, и созданное множество импликаций будет избыточно.

Методы анализа формальных понятий позволяют вместо всех возможных импликаций брать в рассмотрение только минимальное подмножество импликаций, причем все остальные импликации выводятся из этого подмножества по правилам вывода Армстронга [1]. Одним из минимальных базисов импликаций является базис Дюкена-Гига (канонический базис). Дюкен и Гиг установили, что базис для всех импликаций, верных в формальном контексте, задается как $\{P \to P'' \mid P - \text{псевдосодержание}\}$[1].

Определение. Множество $P \subseteq M$ называется псевдосодержанием, если $P \neq P''$ и $Q'' \subset P$ для любого псевдосодержания $Q \subset P$.

Определение. Канонический базис импликаций – $\{P \to (P''\backslash P) | P - \text{псевдосодержание}\}$.

Существует множество различных алгоритмов для построения базиса импликаций, например, алгоритм Гантера, алгоритм Объедкова-Дюкена или алгоритм Бабина для нахождения приближенного базиса.

Работу алгоритма Гантера для построения базиса импликаций можно описать следующим образом [6]:

1. Если существуют признаки, имеющиеся у всех объектов, то поместить в множество импликаций импликацию $\emptyset \to G'$

2. Множество признаков $=$ G'

3. Пока не перечислены все множества, которые могут порождать импликации

3.1. Сгенерировать следующее множество признаков P

3.2. Если $P \neq P''$, то вычислить его замыкание P* относительно ранее найденных импликаций

3.3. Если $P * \neq P''$, то добавить $P * \to P''$ в множество импликаций

4. После вычисления всех импликаций исключить из множества импликаций избыточные.

Для того, чтобы сгенерировать множества признаков, используется обход решетки понятий поиском в глубину.

Генерации неизбыточных множеств признаков возможна, если использовать такое свойство: при осуществлении перехода от одного понятия решетки к нижележащему понятию импликации образуются только добавлением признаков, имеющихся у объектов, которые лежат в объеме предыдущего понятия и не лежат в объеме текущего понятия.

Алгоритм Объедкова-Дюкена основывается на идее, похожей с алгоритмом Гантера. Этот алгоритм добавляет по одному новые признаки, причем начинает с пустого множества, затем пересчитывает текущее множество псевдосодержаний совместно с множеством содержаний [3].

Алгоритм Бабина позволяет найти приближенный базис импликаций.[7]

Определение. Набор импликаций $J\varepsilon$ называется приближенным базисом формального контекста K = (G,M, I), если для случайно и равномерно выбранного подмножества $X \subseteq M$ выполняется условие

$Pr(X'' = X^{J_\varepsilon}) > 1 - \varepsilon$, $0 < \varepsilon < 1$.

Таким образом $1 - \varepsilon$ соответствует точности приближения.

Оценка времени работы алгоритма Гантера и алгоритма Объедкова-Дюкена совпадает и равна $O\big(|M|(|J| + |G|)(|J| + |B(G,M,I)|)\big)$, где $|J|$ – размер минимального базиса, а $|B(G,M,I)|$ – количество замкнутых множеств [4, 5].

Время работы алгоритма нахождения приближенного базиса до достижения точности $(1-\varepsilon)$ для контекста равно $O(|J||M|(|G||M| + |J||M|)(\varepsilon - 1))$, где J – минимальный базис импликаций [7].

В данной работе для нахождения базиса импликаций было решено использовать точную модель, несмотря на такие преимущества приближенного базиса, как более быстрое время построения базиса, так как точный базис содержит большее количество импликаций, что является важным в данной работе для установления большего числа зависимостей. Для нахождения базиса был выбран алгоритм Гантера, так как он проще в реализации, чем алгоритм Объедкова-Дюкена, но при этом имеет такую оценку времени работы.

ГЛАВА 3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МОДУЛЯ

3.1 Описание онтологии и хранение данных

Первым этапом работы было построение онтологии для описания предметной области компьютерной безопасности. Напомним, что онтология – это формализация некоторой области знаний с помощью концептуальной схемы. Онтология должна содержать индивиды и классы предметной области, признаки объектов и отношения между ними. Индивиды (или экземпляры) – это основные компоненты онтологии. Классификация таких объектов является одной из важнейших целей онтологии. Классы (или понятия) – это некоторые абстрактные группы индивидов, которые могут включать в себя индивиды, другие классы или сочетать эти варианты. Индивиды в онтологии могут иметь признаки. Признаки имеют имя и значение и необходимы для хранения информации, специфичной для индивида. Основная их роль состоит в том, чтобы определять зависимости между индивидами онтологии. Отношением является признак, значением которого является другой индивид.

Также онтология должна иметь формат, понятный для компьютера.

Существует множество языков для описания онтологий [11]:

- KIF (Knowledge Interchange Format) – язык обмена знаниями, основанный на формальной логике и имеющий декларативную семантику. Разрабатывался для описания общего формата представления знаний, независящего от конкретной системы.
- CycL – онтологический язык, используемый в проекте Cyc, основанный на исчислении предикатов с некоторыми расширениями более высокого порядка. Он может различать такие сущности, как индивиды, классы, предикаты и функции. Его словарь состоит из термов, множество которых можно разделить на константы, неатомарные термы и переменные. Термы используются при составлении значащих выражений CycL, из которых формируются суждения, из них в свою очередь состоит база знаний.
- RDF – язык, разработанный в рамках проекта семантической паутины (Semantic Web), основным предназначением которого является описание метаданных документов, размещаемых в Интернет. RDF использует базовую модель представления данных "субъект - предикат - объект", называемую триплетом. RDF является не форматом файла, а абстрактной моделью. Для записи и

передачи RDF может использоваться RDF/XML – запись в виде XML-документа. RDF Schema –специальный словарь для RDF.

- DAML+OIL – семантический язык разметки Web-ресурсов, который расширяет стандарты RDF и RDF Schema за счет более полных примитивов моделирования. В последнюю версию DAML+OIL включен набор дополнительных конструкций для создания онтологий и разметки информации в легко интерпретируемом машиной виде.
- OWL (Web Ontology Language) – язык представления онтологий следующего поколения после DAML+OIL. Он обладает более богатым набором возможностей и обеспечивает большую выразительность при сохранении полноты вычислений и разрешаемости по сравнению с XML, RDF, RDF Schema и DAML+OIL.

В данной работе был выбран OWL – язык описания онтологий для Semantic Web. Он является рекомендованным консорциумом Всемирной паутины (W3C). В качестве своего синтаксиса OWL использует язык XML. Основными элементами языка являются свойства, классы и ограничения. Эти элементы позволяют реализовать представление о мире, как о множестве объектов, описываемых некоторым набором свойств. Эти объекты связаны между собой определенными отношениями, а также могут быть объединены по определенным свойствам в классы.

В языке OWL свойства бывают двух типов: DatatypeProperty, т.е. свойства-характеристики и ObjectProperty, т.е. свойства-связи. Первые характеризует объекты (классы), в качестве значений они принимают данные определенных типов, например, String и другие. Вторые ассоциирует объекты (классы) друг с другом, таким образом их значениями могут быть объекты (классы) [16].

В качестве редактора онтологии и фреймворка для построения базы прецедентов использовался Protégé, совместимый со стандартом W3C. Protégé ― это свободная, исходно-открытая платформа, которая обеспечивает пользователей набором инструментов для создания, визуализации и манипуляции онтологий в различных форматах представления. Встроенный решатель Fact++ позволил проверить онтологию на непротиворечивость.

В построенной онтологии предметной области компьютерной безопасности имеются 24 класса.

На рисунке 1 показана иерархия классов онтологии, на рисунке 2 представлен онтограф построенной онтологии.

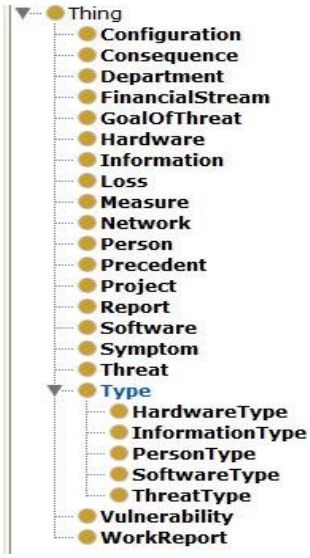

Рис.1 Иерархия классов онтологии

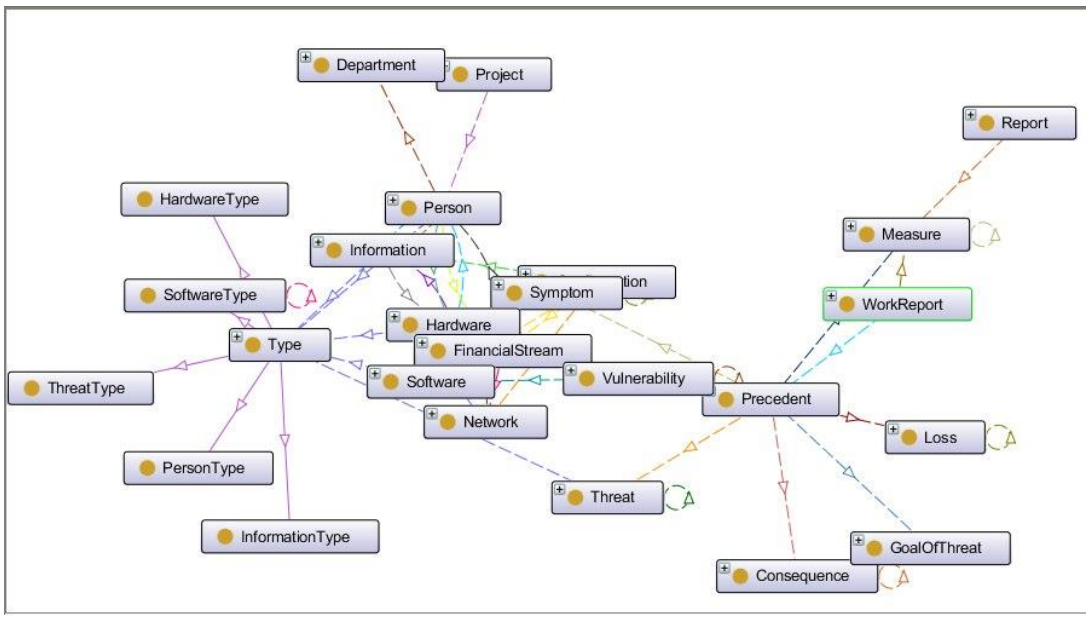

Рис.2 Онтограф онтологии

Следующим этапом работы было занесение в базу знаний данных о прецедентах компьютерных атак. Для этого был реализован программный модуль, который автоматически портировал данные из базы прецедентов формата .mvx, реализованной в системе RiskPanel [8, 9] с использованием технологии OntoBox, в формат .owl.

3.2 Средства разработки модуля

Для разработки модуля использовался объектно-ориентированный язык Java. Для разработки графического интерфейса модуля была выбрана технология JavaFX. Для работы с базой знаний формата .owl использовалась библиотека OWL API согласно ее

лицензии. Для работы с базой знаний Ontobox использовалась библиотека Ontobox-storage.

3.3 Архитектура программы

На рисунке 3 представлена аналитическая диаграмма программы. На рисунке 4 представлена диаграмма классов модуля. Архитектура программы была построена, опираясь на шаблон проектирования Model-View-Controller. Для реализации Model-View-Controller был использован фреймворк JavaFX. Разметка элементов интерфейса была описана с помощью fxml, и был автоматически сгенерирован графический пользовательский интерфейс с помощью этой разметки. Класс RMOntology реализует методы, которые выгружают данные из базы и строят базис импликаций.

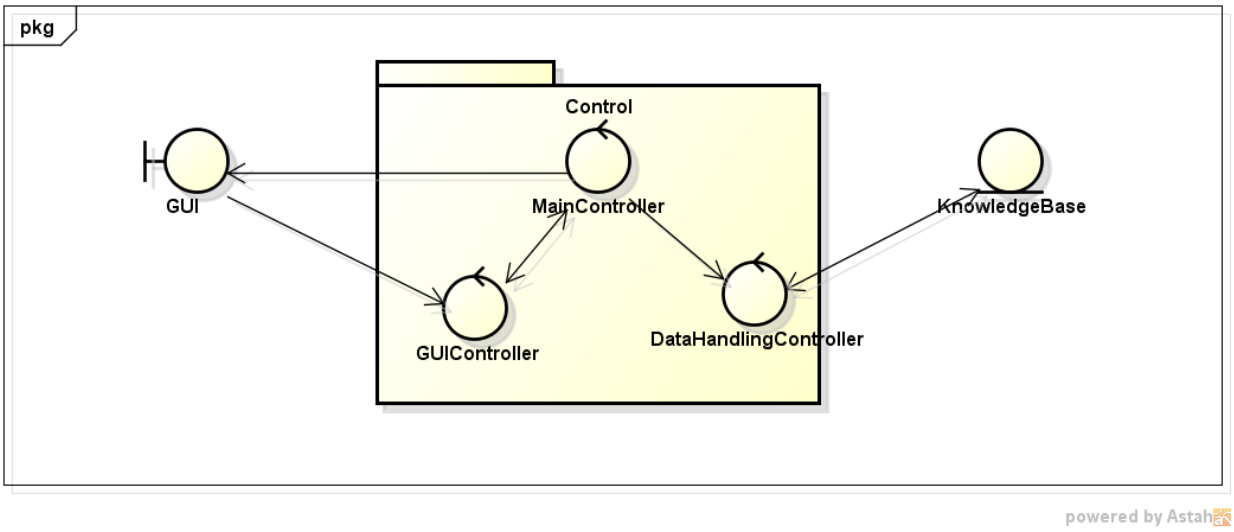

Рис.3 Аналитическая диаграмма

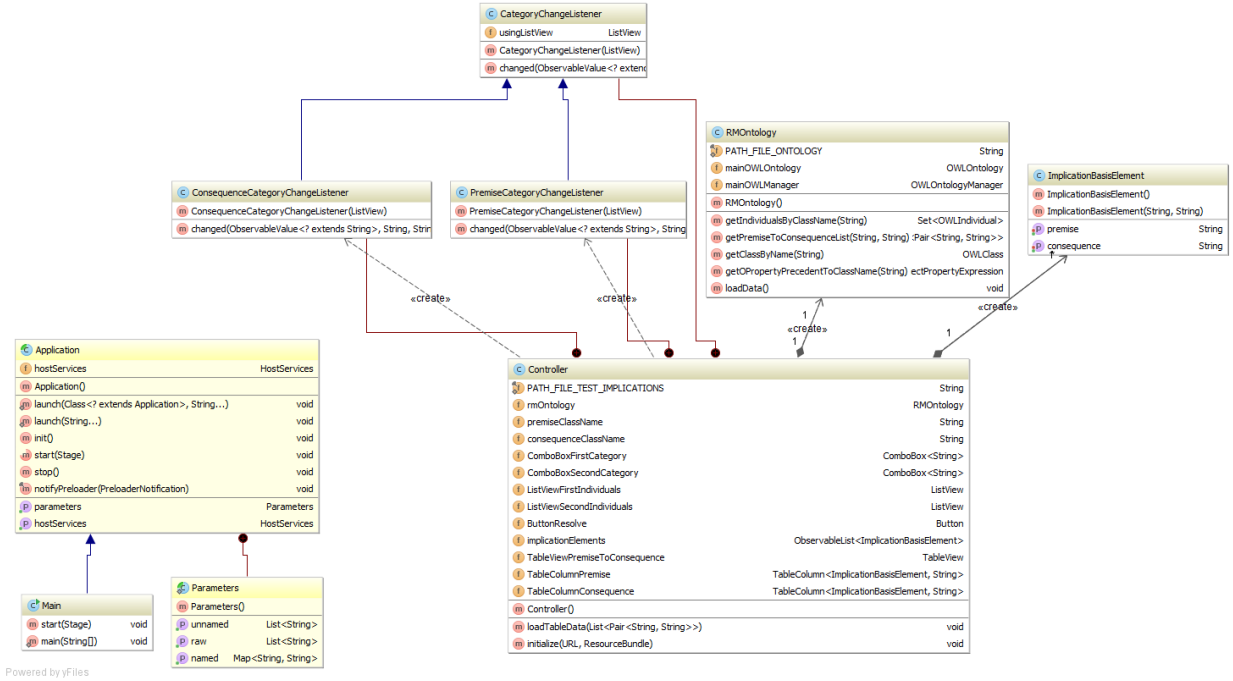

Рис. 4 Диаграмма классов

3.4 Функциональные возможности модуля

Целью пользователя является не получение базиса импликаций, а наглядное представление зависимостей, поэтому пользователю необходима возможность отфильтровать результаты, например, выбирать в качестве посылки категорию признаков «Симптомы», а в качестве заключения – «Меры» или «Уязвимости» → «Последствия» и т.п. В программе была реализована такая возможность для выбора посылки и заключения, чтобы пользователь мог отфильтровать интересные ему результаты. На рисунке 5 представлена диаграмма функциональных требований.

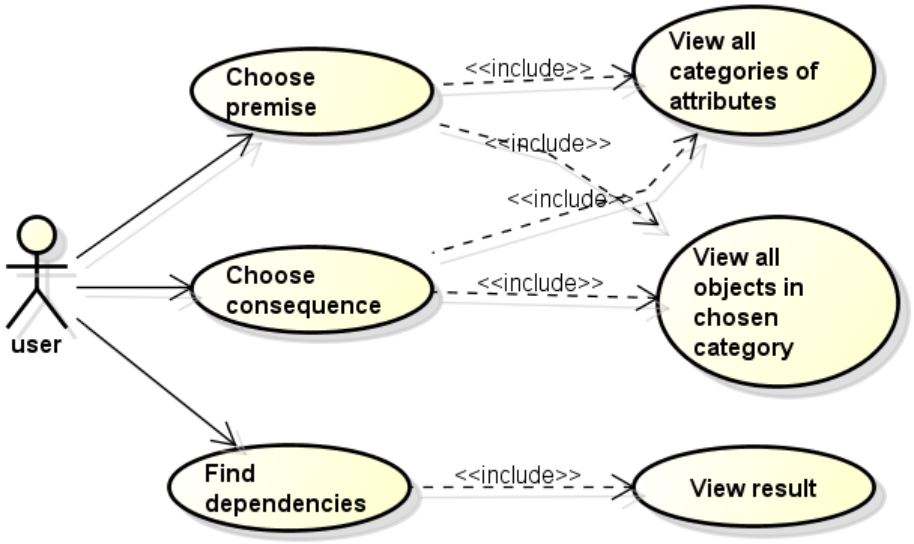

Рис.5 Use-case диаграмма модуля

Интерфейс программы представлен на рисунке 6. Пользователю необходимо выбрать категорию признаков, которую он хочет видеть в качестве посылки импликации, в первом раскрывающемся списке, объекты этой категории отобразятся в первой таблице. Затем пользователь должен выбрать другую категорию во втором списке, которая интересна для него как заключение импликации. Результат работы, содержащий зависимости между признаками выбранных пользователем категорий, выводится в третьей таблице.

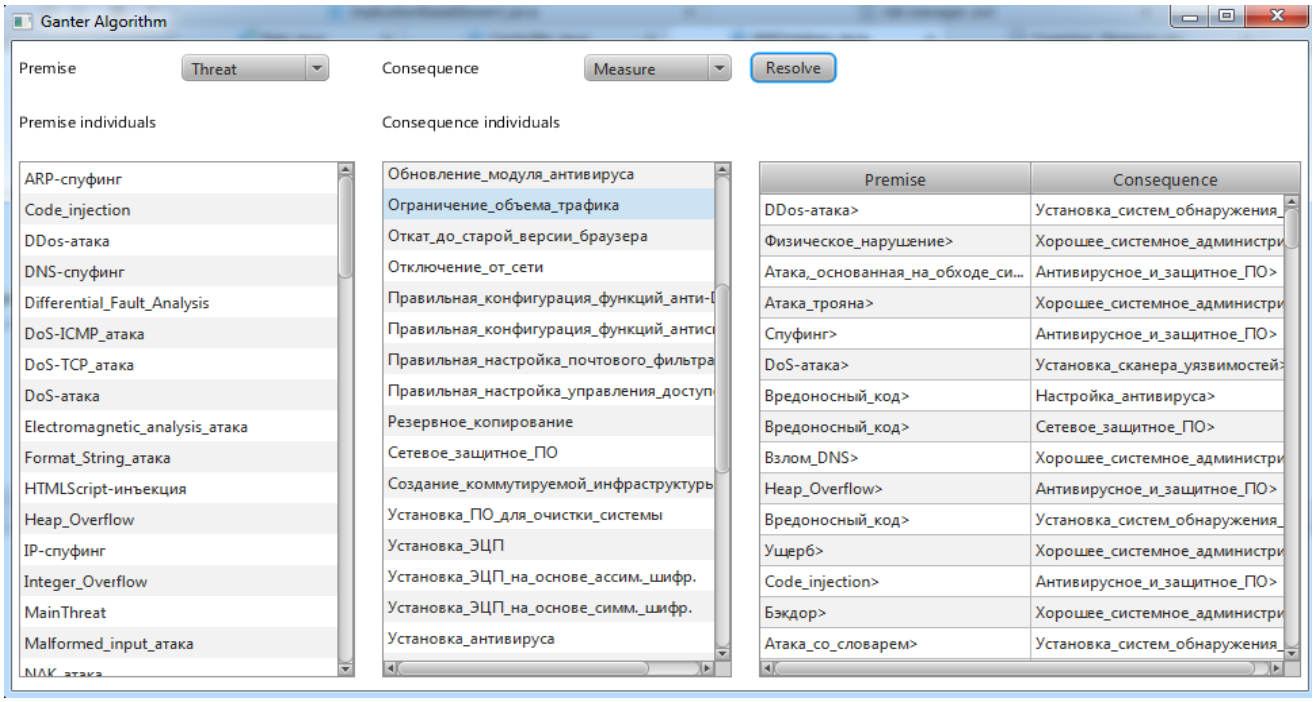

Рис.6 Интерфейс модуля

Заключение:

В результате проведенной работы были получены следующие результаты:

- построена онтология на языке OWL, описывающая предметную область компьютерных атак, и подготовлена база знаний;

- изучены и применены методы анализа формальных понятий для построения минимального базиса импликаций;

- разработан модуль, автоматически извлекающий связи между признаками прецедентов компьютерных атак;

- проведено тестирование модуля на подготовленном тестовом наборе прецедентов.

Данная работа была представлена на МНСК-2014 в секции «Информационные технологии» [10].

Литература

1. B. Ganter, R. Wille. Formal Concept Analysis: Mathematical Foundations // Berlin; Heidelberg: Springer Verlag, 1999.
2. Formal Concept Analysis [Электронный ресурс] Wikipedia, the free encyclopedia: URL: http://en.wikipedia.org/wiki/Formal_concept_analysis (дата обращения: 20.04.2014).
3. V. Duquenne, S.A. Obiedkov. Attribute-incremental construction of the canonical implication basis // Annals of Mathematics and Artificial Intelligence, 2007, vol. 10, no.1, pp. 77 - 99.
4. S.O. Kuznetsov. On the Intractability of Computing the Duquenne-Guigues Base. Journal of Universal Computer Science, 2004, vol. 10, no. 8, pp. 927 - 933.
5. S.O. Kuznetsov, S.A. Obiedkov. Some Decision and Counting Problems of the Duquenne-Guigues Basis of Implications. Discrete Applied Mathematics Vol. 156, no. 11, pp. 1994 - 2003, 2008.
6. С.А. Евтушенко. Система анализа данных "Concept Explorer" // Труды 7-ой национальной конференции по искусственному интеллекту КИИ-2000. – М.:Физмалит – 2000. – с. 127 - 134.
7. М.А. Бабин. О приближенном базисе импликаций // Научно-техническая информация. Сер. 2, Информационные процессы и системы. – 2012. – № 8. – с. 20 – 23.
8. Г.Э. Яхъяева, О.В. Ясинская. Применение методологии прецедентных моделей в системе риск-менеджмента, направленного на раннюю диагностику компьютерного нападения // Вестник Новосибирского государственного университета. Серия: Информационные технологии – 2012. – Т.10 – вып. 3 – с. 106 - 115.
9. Д.Е. Пальчунов, Г.Э. Яхъяева, А.А. Хамутская. Программная система управления информационными рисками RiskPanel // Программная инженерия – 2011 – № 7 – с. 29 -36.
10. Н.С. Буркова. Применение методологии анализа формальных понятий для автоматического извлечения причинно-следственных связей компьютерных атак // Материалы 52-ой международной научной студенческой конференции «Студент и

научно-технический прогресс», Информационные технологии. Новосибирск – 2014 – с. 235.

11. Методы инженерии знаний [Электронный ресурс] URL: https://sites.google.com/site/upravlenieznaniami/inzeneria-znanij/sredstva-inzenerii-znanij (дата обращения: 29.04.2014).
12. А.Н. Мошков: «Киберпреступность усиливает свои позиции» [Электронный ресурс] SecurityLab URL: http://www.securitylab.ru/blog/company/Personal_data/an-moshkov-kiberprestupnost-usilivaet-svoi-pozitsii.php (дата обращения: 14.04.2014).
13. Исследование Norton: ущерб от киберпреступности в России оценивается в 2 миллиарда долларов США в год [Электронный ресурс] URL: http://www.symantec.com/ru/ru/about/news/release/article.jsp?prid=20120913_01 (дата обращения 14.04.2014).
14. Итоги исследования: Киберпреступность в России и мире [Электронный ресурс] Сайт компании Symantec URL: http://rusbase.vc/news/itogi-issledovaniya-kiberprestupnost-v-rossii-i-mi/ (дата обращения: 14.04.2014).
15. Д.И. Игнатов, О.Н. Кононыхина. Решетки формальных понятий для анализа данных социологических опросов // Интегрированные модели и мягкие вычисления в искусственном интеллекте. Сборник научных трудов V-й Международной научно-технической конференции. Т1. – М.: Физматлит, 2009. – 546 с.
16. С.А. Паньгин. Проблема описания семантики предметной области в дистанционном обучении [Электронный ресурс] URL: http://window.edu.ru/resource/310/37310/files/2004_6_27-33.pdf (дата обращения 10.05.2014).