УДК 621.8

Ю.Н. Бардачев, В.И. Литвиненко, А.А. Дидык

ПРИМЕНЕНИЕ ИСКУССТВЕННЫХ ИММУННЫХ СИСТЕМ ДЛЯ ОБНАРУЖЕНИЯ КОМПЬЮТЕРНЫХ АТАК

Введение

Современные системы обнаружения атак способны контролировать в реальном масштабе времени сеть и деятельность операционной системы, обнаруживать несанкционированные действия, и, автоматически реагировать на них практически в реальном масштабе времени. Кроме того, системы обнаружения атак могут анализировать текущие события, принимая во внимание уже произошедшие события, что позволяет идентифицировать атаки, разнесенные во времени и тем самым прогнозировать будущие события.

В этой области можно выделить несколько подходов. Экспертные системы обнаружения атак, основанные на *сигнатурном методе*, (например, ASAX) базируются на конкретных правилах, добавляемых разработчиком системы по мере появления новых атак. Деятельность системы заключается в анализе сетевого трафика (network-based) и анализе журналов регистрации операционной системы или приложений на уровне хоста (host-based). В любом случае, система обнаружения атак ищет известные шаблоны, которые указывают на нападение. Эффективность таких систем напрямую связана со своевременным обновлением базы данных сигнатур, так как новые атаки и уязвимости обнаруживаются постоянно. Создание такой базы данных требует наличия хорошо осведомленных экспертов, имеющих доступ к большому числу источников информации об атаках. Способность создания и обновления такой базы данных является критическим параметром, по которому оцениваются сигнатурные системы обнаружения атак.

Другим методом обнаружения стал анализ состояний переходов, пример такого рода систем – NetSTAT, базирующийся на описании сценариев атак в виде автоматов с набором состояний атакующих объектов и последующем анализе переходов из состояния в состояние

по ходу сетевой сессии. Данную систему можно рассматривать как вариант экспертной системы обнаружения атак.

Бурный рост экспертных систем, как коммерческих (RealSecure), так и свободно распространяемых (Snort), показал достаточную эффективность такого подхода к задаче обнаружения атак. Однако такие системы обладают рядом недостатков: стоит проблема обнаружения ранее неизвестных атак, необходимость постоянного обновления базы данных сигнатур.

Другим развивающимся направлением в области Intrusion Detection являются *нейронные сети*, уже нашедшие практическое применение во многих других задачах. Первые опыты в применении нейросетей для обнаружения атак относятся к концу 90-х годов. Одна из таких систем была разработана Джеймсом Кеннеди (James Cannady) в университете NSU (США). Система была построена для исследования возможности применения нейросетей к задаче обнаружения атак методом выявления злоупотреблений. В результате проверок на тестовых примерах исследователи получили приемлемый уровень погрешности. Данная система использовалась только для изучения в лабораторных условиях. Первый прототип нейросети был предназначен для того, чтобы определить возможности нейросети идентифицировать конкретные события, указывающие на злоупотребление. Данные для обучения нейросети и тестирования прототипа генерировались с использованием сетевого монитора. В дополнение к «обычной» сетевой активности, информация о которой собиралась системой RealSecure, хост для монитора был «атакован» с использованием продукта Internet Scanner и сканером Satan. Сканеры были сконфигурированы на большое количество атак, начиная от атак типа «отказ в обслуживании» до сканирования портов. Приблизительно 10000 отдельных событий были собраны системой RealSecure и сохранены в базе данных, из которых приблизительно 3000 были смоделированными атаками. Далее, после предварительной обработки входных параметров, они подавались на вход нейросети для обучения, после чего были произведены испытания и получены оценки эффективности работы нейросетевой системы обнаружения атак.

Еще один пример - NNID (Neural Network Intrusion Detector, нейросетевой обнаружитель вторжений), разработанная в

университете Техаса в Остине, США. NNID – экспериментальная система, использующая знания о нормальном поведении. Система обнаружения аномалий NNID основана на идентификации авторизованного пользователя, на основе распределения команд, которые он выполняет. Различные пользователи имеют тенденцию использовать различные режимы работы в зависимости от своих потребностей в системе. Набор используемых команд и их частота образуют «отпечаток» пользователя, отражая выполняемую задачу и выбор прикладных программ, и на основе этой информации становится возможным идентифицировать пользователя. В результате испытаний система показала хорошие результаты.

В настоящий момент методы обнаружения атак, основанные на нейронных сетях, способны на основе данных, полученных во время предварительного обучения, распознавать сетевые атаки с некоторой точностью. Наряду с искусственными нейронными сетями существует еще один статистический метод – *искусственные иммунные системы* – результат математического моделирования принципов обработки информации естественной иммунной системой. Обзор математического аппарата и направлений ИИ показывает, что эта проблема мало рассматривалась, несмотря на ее актуальность. Большой интерес в этой области представляет исследование Стефании Форрест [1]. В нем разрабатывается система математических моделей обработки информации моделирующей отрицательный отбор Т-лимфоцитов как биологического прототипа. Актуальной является задача исследования возможности практического применения иммунных систем на основе отрицательного отбора в задаче обнаружения атак. Обладая уникальными свойствами иммунные системы обладают преимуществом в обучении. Также иммунные системы обладают большей по сравнению с нейросетями точностью распознавания.

Проблема выявления атак на компьютерные сети является наиболее актуальной при решении проблем защиты информации [2–4]. В настоящий момент существует два основных подхода обнаружения атак:

- поиск некоторого постоянного набора атрибутов, однозначно характеризующих наличие атаки;
- пропуск только того, что известно и разрешено.

Одной из перспективных технологий для решения подобных задач являются искусственные иммунные системы (ИИС). ИИС обладают рядом свойств (уникальность, распределенность в пространстве выявления чужеродных агентов, “грубое” или “несовершенное” распознавание, высокая адаптивность, устойчивость к ошибкам, отсутствие централизованного управления), которые в целом решают некоторые задачи лучше даже, чем нейросетевые технологии.

Существует три базовые модели ИИС: системы, основанные на принципах отрицательного отбора, на принципах клонального отбора и принципах идиопатических иммунных сетей. В работе предлагается использовать алгоритм отрицательного отбора для выявления компьютерных атак.

Алгоритм отрицательного отбора и его модификации, несмотря на кажущуюся свою простоту является мощным инструментом для решения таких задач, как защита компьютерных сетей [5].

Суть биологического механизма отрицательного отбора заключается в том, что иммунная система стремится обеспечить толерантность к “своим” клеткам и молекулам. Это развивает способность иммунной системы обнаруживать неизвестные антигены при отсутствии какой-либо реакции на свои клетки. В течение текущего поколения с помощью псевдослучайного процесса генетической перестановки создаются рецепторы Т-клеток. После этого, они подвергаются цензурированию в вилочковой железе (Тимусе), посредством процесса который называется “отрицательным отбором”. Далее в Тимусе, Т-клетки, реагирующие против “своих” белков, разрушаются и только те из них, которые не связываются со “своими” белками, позволяется покинуть Тимус. Эти созревшие Т-клетки после этого циркулируют по всему телу, чтобы осуществлять свои иммунологические функции и защитить в целом организм против чужеродных антигенов.

Вычислительная модель алгоритма *отрицательного отбора* (АОО) *была* предложена Стефанией Форрест в 1994 году [6]. Основная идея заключалась в генерировании набора бинарных датчиков создаваемых кандидатов и затем отказаться от тех из них, которые в дальнейшем распознают “свои” данные из набора обучающей выборки. В дальнейшем эти датчики или детекторы могут использоваться, для

обнаружения каких-либо аномалий. Так что АОО состоит из трех стадий: определение “своих”, генерирование датчиков и контроль за возникновением аномалий. Были предложены различные разновидности этого алгоритма (датчик, генерирующий шаблоны). Первичными применениями АОО было в области обнаружение изменений (или аномалий), где датчики генерировались в дополнительное пространство, которое может обнаружить изменения в образцах данных. Главный компонент АОО – выбор соответствующего правила, которое определяет сходство между двумя образцами, чтобы осуществить классификацию свои/чужие (нормальные/ неправильные) образы.

Временная сложность генерирования датчиков линейна относительно размера "своих" данных. Для гарантии заданного “уровня надежности” необходима также оценка обнаружении аномалий. Сгенерированные датчики, имеющие высокий порог соответствия, становятся чувствительными к любой аномалии в данных, так что необходимо большое количество датчиков позволяющие достигнуть желательного уровня полной надежности. С другой стороны, если порог является слишком маленьким, не может быть возможной генерации разумного размера набора датчиков от доступных “своих”. Это предлагает, что значение порога может использоваться, чтобы настроить надежность обнаружения против риска ложных положительных распознаваний. Это особенность применяется при обнаружении изменений в общем случае.

В работе [7] приводятся результаты работы, где автор рассмотрел и сравнил пять схем поколения датчиков (бинарных) алгоритмов отрицательного отбора: исчерпывающий, линейный, “жадный”, бинарный шаблон, и мутация АОО. В работе [8] авторы проанализировали и сравнили различный правила бинарного соответствия в отрицательном отборе: в том числе и r-смежное соответствие, соответствие r-куска, соответствие с использованием Хеммингового расстояния, и его разновидность – R&T-соответствие. Это таким образом обеспечивает директиву для того, чтобы выбрать различные правила соответствия для любых алгоритмов отрицательного отбора [13-14].

В других работах также исследовалось различное представление схемы отрицательного отбора и алгоритмов генерации датчиков для

таких представлений [9]. В частности исследуемые представления включали гиперпрямоугольники (который можно интерпретировать как правила), нечеткие правила, и гиперсферы.

В настоящий момент различными авторами предложено четыре различных алгоритма генерации датчиков, соответственно: 1) отрицательный отбор с правилами обнаружения (эволюционный алгоритм, для генерирования датчиков гиперкуба); 2) отрицательный отбор с нечеткими правилами обнаружения (эволюционный алгоритм, для генерирования датчиков нечетких правил), 3) отрицательный отбор с использованием действительных чисел (эвристический алгоритм, для генерирования гиперсферических детекторов), и 4) отрицательный отбор на основе рандомизированых действительных чисел (алгоритм позволяющий генерировать гиперсферические датчики используя для этого метод Монте Карло). Разработан также гибридный алгоритм обучения, обладающий иммунитетом, который комбинирует способы генерирования детекторов 3) или 4) с алгоритмами классификации.

Другая разновидность модели отрицательного отбора предлагает использовать динамический алгоритм клонального отбора (Динамика), чтобы иметь дело с "чужим", задача заключается в обнаружения в условиях непрерывно меняющейся окружающей среде [10]. В частности динамика основывается на идее Хофмеера [11] о динамике трех различных популяций: незрелая, зрелая, и популяция датчиков памяти. Первоначальные незрелые датчики, генерируются со случайными генотипами. Используя отрицательный отбор, новые незрелые датчики добавляются, для того чтобы общее количество датчиков сохранялось постоянным после предопределенного числа поколений (период поляризации Т). Если датчик - в пределах его предопределенной продолжительности жизни L, и соответствующее число является большим, чем предопределенный порог активации A, он становится датчиком памяти. Зрелые датчики используются, чтобы идентифицировать неизвестные нападения. Однако, необходимо подтверждение человека отвечающего за безопасность (ко-стимуляция, который делает этот подход, зависящим от человеческого взаимодействия. В отечественных исследованиях, к сожалению данный алгоритм все еще остается малоизвестным. Более детальных анализ алгоритмов отрицательного отбора и его экспериментальные исследования представлены нами в работах [13-18].

Постановка задачи

Нами разработана методика выявления атак с помощью системы построенной на основе механизмов отрицательного отбора. Задача идентификации атак рассматривалась как задача бинарной классификации [12], использующей принцип определения "свой/чужой" При этом предусмотрена реализация подхода, осуществляющего как поиск атрибутов, однозначно характеризующих наличие атаки, так и определение и пропуск только тех, чьи атрибуты хорошо известны, что позволяет значительно увеличить качество работы системы идентификации. Специфика задачи состоит в необходимости учета следующих обстоятельств – заданы два вектора, компоненты которого характеризуют трафик, классы ситуаций, которые определяют наличие атаки и ее отсутствие. Результаты тестирования показали, что даже в случае отсутствия некоторых данных система может правильно осуществлять классификацию.

Общая формулировка задачи обнаружения компьютерных атак

Используемая модель задачи обнаружения атак была предложена в [16, 17].

Под *атакой на распределенную информационную систему (РИС)* понимают воздействие (последовательность воздействий), производимое *нарушителем* и ведущее к нарушению информационной безопасности системы, т.е. переходу РИС из некоторого безопасного в некоторое опасное состояние. Атака всегда переводит РИС из безопасного состояния в опасное. Под нарушителем понимают объект, осуществляющий воздействие (последовательность воздействий). *Нормальное поведение* – воздействие (последовательность воздействий), не являющееся атакой.

Наблюдатель – программная или программно-аппаратная сущность, имеющая возможность собирать информацию о действиях объектов и состоянии ресурсов.

Задача обнаружения атак – это процесс выявления атаки, осуществляемый на основе информации о состоянии РИС, получаемой от наблюдателя.

Состояние ресурса РИС в текущий момент времени можно описать некоторым набором параметров, включающим в себя как

характеристики загруженности ресурса, так и информацию об объектах, пользующихся ресурсом.

Упорядоченная во времени последовательность состояний РИС называется *траекторией РИС*. *Траектория объекта* – упорядоченная во времени последовательность состояний ресурса РИС, изменяющихся вследствие воздействия, осуществляемого этим объектом на ресурс РИС.

Траекторию объекта, осуществляющего воздействие на РИС, повлекшее переход из безопасного состояния в опасное, называют *траекторией атаки*. Таким образом, конкретную атаку рассматривают как траекторию в n-мерном пространстве параметров. Множество всех атак можно условно разделить на классы по ряду признаков [16].

Для атак одного класса может существовать не одна траектория, а некоторое конечное множество возможных траекторий, образующих *пучки* близких друг к другу *траекторий*.

Вследствие меньшей вычислительной сложности, используются *дискретные траектории*. Пусть $\bar{t}$ - конечная длительность атаки заданного класса и τ - период времени замера наблюдаемой траектории. Таким образом, в траектории поведения атаки задано $k = \lceil \bar{t} / \tau \rceil$ состояний. Для каждого замера параметров (номер замера однозначно определяется значением k) в пространстве параметров существует набор несвязных областей таких, что попадание замера в одну из этих областей означает принадлежность траектории множеству L на данном замере. Это *«опасные области»*, которые, по сути, образуют срезы пучков траекторий атак.

Полагаем, что атака это активность, под воздействием которой система проходит через опасные области на всех замерах в течение атаки. То есть, если в ходе k замеров состояний сессии траектория на каждом замере попадала в одну из таких областей, то траектория принадлежит множеству L множеству траекторий, соответствующему определенному классу атак. Таким образом, задача обнаружения атак некоторого класса сводится к задачам:

1. построения набора опасных областей $G(t) = \{G_1(t), G_2(t), \ldots, G_l(t)\}$ для каждого замера;

 ISSN 1562-9945

2. определения принадлежности замеренного состояния сессии x к одной из этих областей $(x \in G)$.

Архитектура иммунной системы, реализующей алгоритм отрицательного отбора

Архитектурная схема иммунной системы, реализующей алгоритм отрицательного отбора, состоит из семи блоков, каждый из которых представлен одним или несколькими классами. Блоки на схеме соответствуют основным функциональным узлам системы, а стрелки, соединяющие блоки – основным потокам передачи данных и управляющих параметров. Краткое описание элементов схемы представлено ниже.

- интерфейс ввода/вывода данных. Блок предназначен для загрузки обучающих и распознаваемых данных в систему, получения результата распознавания, а, также, сохранения текущего состояния множества детекторов для возможности быстрой перенастройки системы на другую задачу без дополнительного обучения;

- база данных (БД). Данные, предназначенные для обучения или распознавания хранятся в базе данных системы и выбираются оттуда по мере надобности. Здесь же хранятся результаты распознавания и текущее состояние множества детекторов;

- генератор случайных чисел (ГСЧ). Использует несколько видов распределений и может генерировать целые или вещественные числа в заданных диапазонах;

- блок генерации кандидатов. Используя последовательности случайных чисел, создаваемые ГСЧ, производит множество кандидатов детекторов для последующего отбора их в качестве детекторов;

- блок проверки совпадения. Во время обучения системы данный блок используется для создания множества детекторов. Проверяет два вектора на предмет совпадения их между собой. Для проверки использует заданное правило совпадения и порог совпадения, определяющий границы зоны совпадения. В режиме распознавания системы данный блок распознает поступающие на его вход тестируемые вектора;

- интерфейс управления памятью. Реализует набор процедур, необходимых для управления памятью системы и работы с пространственными формами – внутренним представлением данных;

- **интерфейс настройки алгоритма.** Блок предоставляет возможность настройки системы с использованием подгружаемого файла конфигурации или интерактивную настройку при помощи стандартных графических элементов управления операционной системы.

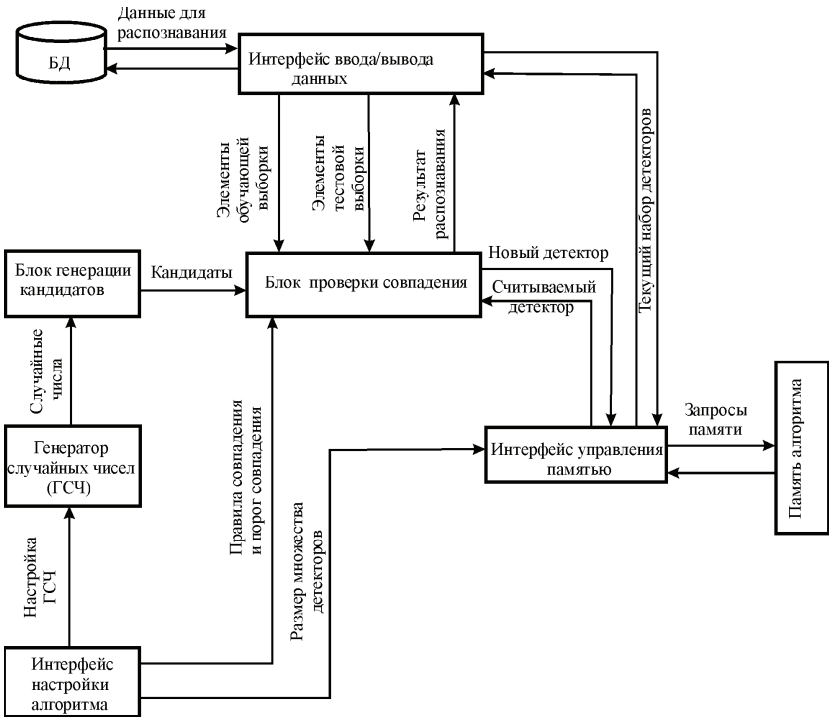


Рисунок 1 – Архитектура иммунной системы, реализующей алгоритм отрицательного отбора

Объектно-ориентированное представление алгоритма отрицательного отбора

Прежде чем приступить к описанию структуры библиотеки, остановимся на условных обозначениях, используемых в схемах (табл.1), остальные обозначения представлены в соответствии со стандартами языка UML..

Основываясь на данных модели, рассмотренной в предыдущем разделе можно выделить следующие объекты предметной области.

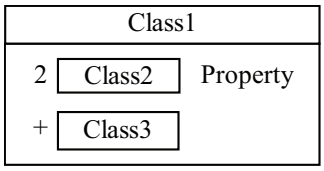
1. Генератор случайных чисел (Класс CRandomGenerator). Генерирует случайные числа в различных интервалах и с различными функциями распределения.
2. Абстрактный класс объектов алгоритма отрицательного отбора (CNSObject). Содержит базовые свойства для всех объектов системы.

 ISSN 1562-9945

3. Класс параметров алгоритма (CParametrs). Содержит все настройки алгоритма отрицательного отбора.
4. Класс атрибут формы (CAttribute). Представляет собой минимальный элемент пространственной формы, описывающей образ.
5. Класс пространственной формы. (CShape). Класс, описывающий образ объектов.
6. Класс множества объектов пространственной формы (CShapeSet).Описывает множество объектов пространственной формы (множество детекторов, множество кандидатов, обучающую выборку) и операции над ними.

Таблица 1

Используемые условные обозначения

Обозначение	Описание
	Показывает, что Class1 является контейнером для объектов Class2 и Class3. Цифра в левой части прямоугольника показывает, какое количество экземпляров объектов соответствующего класса может содержать контейнер. Знак «+» говорит о том, что должен содержаться как минимум один экземпляр

7. Класс алгоритма отрицательного отбора (CNSAlgorithm). Класс включающий в себя основные и вспомогательные объекты модели и обеспечивающий функционирование отрицательного отбора.
8. Класс с правилами совпадений (CMatchingRule). Класс содержащий в себе набор различных правил для сравнения образов (пространственных форм).
9. Класс внутреннего преобразования данных (CDataDriver). Класс предназначен для ввода/вывода данных (класс-интерфейс).

Пример диаграммы взаимного включения основных объектов для построения алгоритма отрицательного отбора показан на рисунке 2.

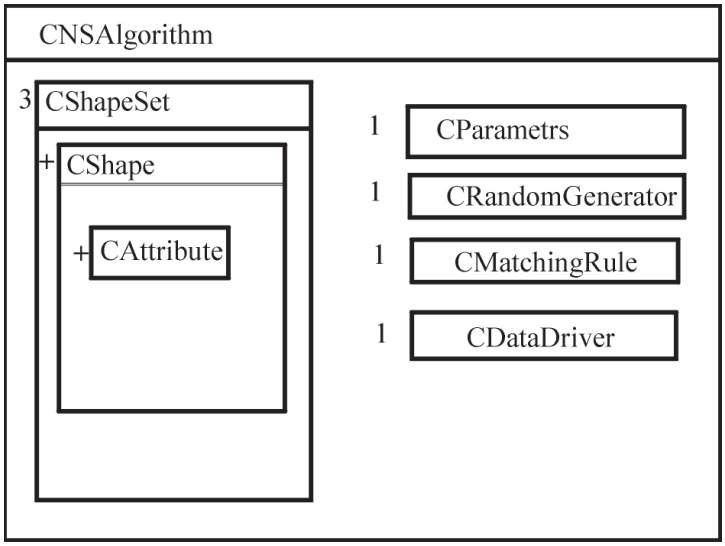


Рисунок 2 – Диаграмма взаимного включения объектов

Для дальнейшего анализа используемого алгоритма, а также для выражения поведения отдельных классов и их возможного взаимодействия (рис. 5), были разработаны диаграммы состояний в режиме обучения (рис. 3) и в режиме распознавания (рис. 4).

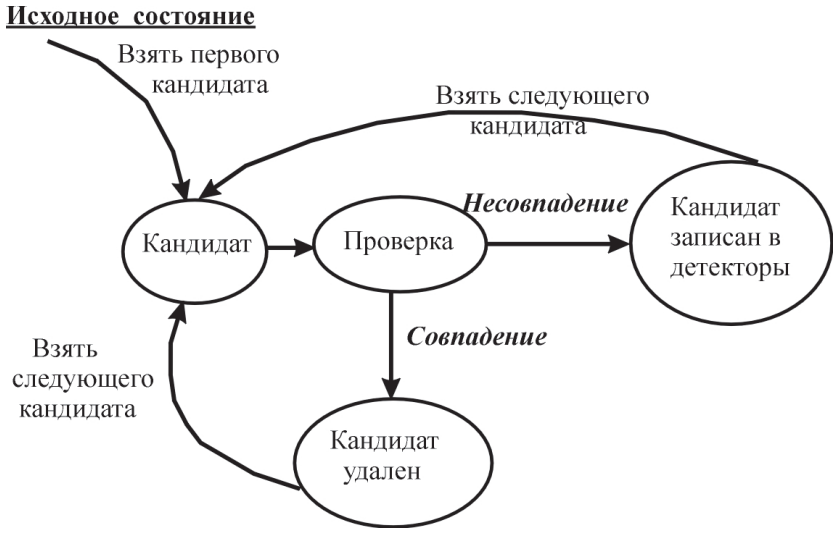


Рисунок 3 – Диаграмма состояний в режиме обучения

В соответствии с представленной диаграммой состояния системы в режиме обучения, предполагается, что в исходном состоянии предполагается, что обучающая выборка уже загружена и множество кандидатов-детекторов создано в соответствии с параметрами алгоритма. Вводится первый кандидат. Осуществляется его проверка в соответствии с правилом совпадения с обучающей выборкой. В случае хотя бы одного совпадения система переходит в состояние удаления кандидата и возвращается к состоянию оценки нового кандидата. В случае несовпадения кандидата с обучающей выборкой

 ISSN 1562-9945

система переходит в состояние записи кандидата в множество детекторов и также возвращается к проверке следующего кандидата. Процесс продолжается до тех пор, пока не будет исчерпано все множество кандидатов.

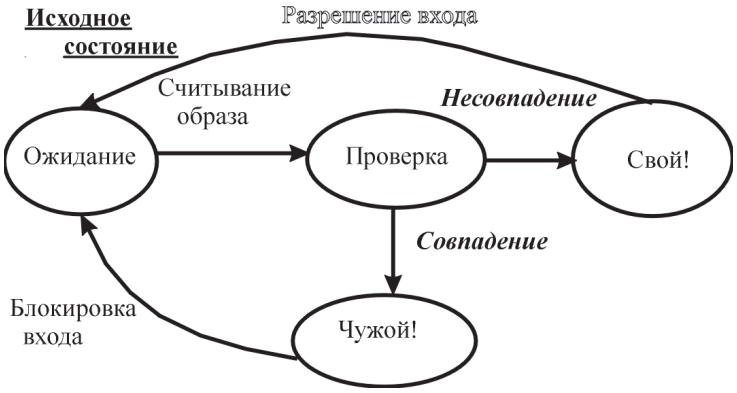


Рисунок 4 – Диаграмма состояний системы в режиме распознавания

В исходном состоянии предполагается, что множество детекторов уже сформировано. Находясь в состоянии ожидания система готова к считыванию входного образа. Если данное событие произошло, осуществляется проверка входного образа с множеством детекторов сформированным на этапе обучения. Если обнаружено совпадение входного образа хотя бы с одним элементом множества детекторов. Формируется состояние “чужой” и система блокирует вход. Иначе вход разрешается, и система переходит в состояние ожидания.

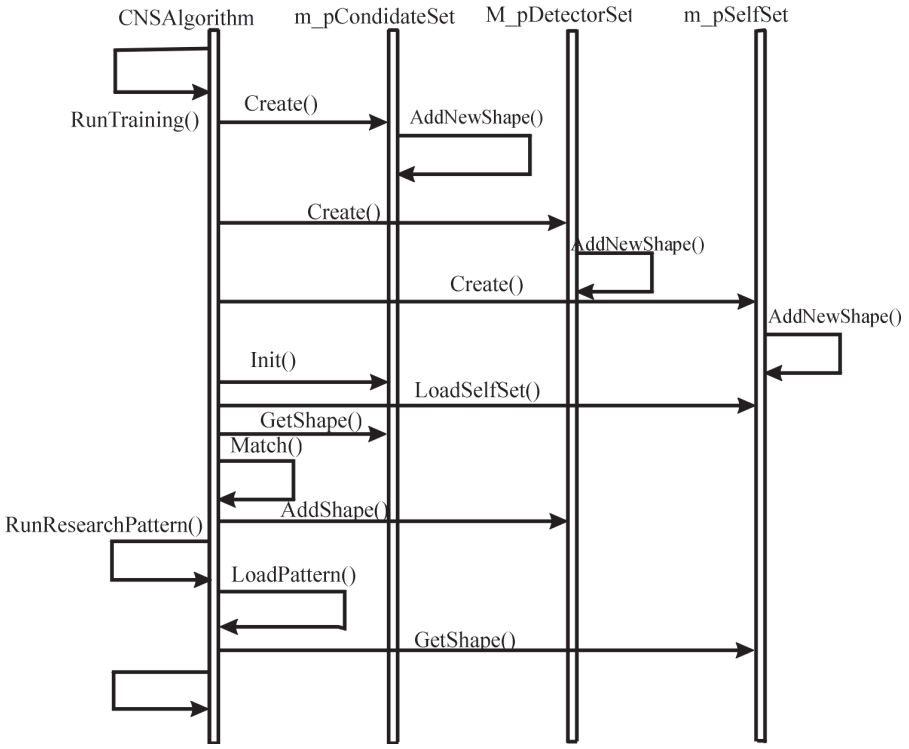


Рисунок 5 – Диаграмма взаимодействия основных объектов

В таблице 2 представлены результаты оценки работы трех различных систем обнаружения компьютерных атак. В первой колонке таблицы отображен тип оцениваемой системы: экспертная система (ЭС), нейросетевая система (НС), искусственная иммунная система (ИИС). Вторая колонка содержит количество атак, обнаруженных каждой системой и, через слеш, общее количество атак, которые эта система должна была обнаружить на основе входных данных. Информация, содержащаяся в остальных колонках таблицы, характеризует точность данных, идентифицирующих обнаруженные атаки.

В третьей колонке содержится процент обнаруженных атак, в которых был правильно определен тип атаки.

Выводы

Следующая колонка отображает процент обнаруженных атак, в которых были правильно определены атакуемые порты, и последняя колонка содержит процент обнаруженных атак, в которых были правильно идентифицированы IP-адреса атакующих.

Таблица 2

Результаты оценки работы систем обнаружения удаленных атак

	Обнаруженные атаки/ Общее число атак	Правильно определенный тип атаки	Правильно определенные атакуемые порты	Правильно определенные адреса источника атаки
ЭС	45/100	77%	54%	80%
НС	83/170	94%	71%	69%
ИИС	85/170	100%	63%	90%

Как видно из таблицы, результаты проведенных экспериментов, затрагивающие обнаружение атак разных классов, показывают о высокой эффективности и перспективности применения искусственных иммунных систем. Использование данных технологий позволит создать системы обнаружения атак более высокого уровня, чем существующие.

 ISSN 1562-9945

ЛИТЕРАТУРА

1. S. Forrest. A. S. Perelson, L. Allen, and R. Cherukuri. Self-Nonself Discrimination in a Computer. In Proceedings of IEEE Symposium on Research in Securitv and Privacy, pages202~212, Oakland, May 16-18 1994.
2. Литвиненко В.И. Классифицирующая система на основе механизмов отрицательного отбора.// Материалы III Міжнародної науково-практичної конференції "ДИНАМІКА НАУКОВИХ ДОСЛІДЖЕНЬ 2004" С.34-36
3. Литвиненко В.И. Разработка и применение искусственных иммунных систем// Третя міжнародна науково-практична конференція МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ (MIZIS-2005), ТЕЗИ ДОПОВІДЕЙ, Дніпропетровськ, 16-18 листопада 2005 р. с.105.
4. P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: algorithms, analysis, and implications. In Proceedings of IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1996.pp.110-119.
5. S. Forrest. A. S. Perelson, L. Allen, and R. Cherukuri. Self-Nonself Discrimination in a Computer. In Proceedings of IEEE Symposium on Research in Securitv and Privacy, pages202~212, Oakland, May 16-18 1994.
6. M. Ayara, J. Timmis, R. de Lemos. L. de Castro and R. Duncan, Negative Selection: How to Generate Detectors, 1st ICARIS, 2002
7. Fabio Gonzalez, Dipankar Dasgupta, An Immunogenetic Approach to Intrusion Detection, GECCO, 2002
8. Fabio Gonzalez, Dipankar Dasgupta. and Luis Fernando Nino, A Randomized Real-Value Negative Selection Algorithm, ICARIS-2003.
9. J. Kim amd P. J. Bentley, Toward an artificial immune system system for network intrusion detection: An investigation of dynamic Clonal selection, in Proceedings of the 2002 Congress on Evolutional Computation CEC2002. Honolulu. 2002
10. S. Hofmeyr and S. Forrest, Architecture for an artificial immune system, Evolutionary Compulation, vol. 8, no. 4, pp. 443-473. 2000

11.Кеннеди Дж. Нейросетевые технологии в диагностике аномальной сетевой активности. – / http://www.beda.stup.ae.ru/lib/neuro/id/neuronet.htm

12.Литвиненко В.И. Иммунный классификатор для решения задач бинарной классификации (Теоретические основы) Системные технологии 2006 г № 1 (42) с. 114-130

13. Литвиненко В.И. Иммунный классификатор для решения задач бинарной классификации (Практическая реализация) Системные технологии 2006 г № 5 (46) с. 113-126

14.Грицик В.В., Литвиненко В.І., Цмоць І.Г., Стех С.М. Теоретичні і прикладні проблеми застосування штучних імунних систем// Інформаційні технології і системи. –2003 –Т.6.-№1-2, с.7-45.

15.Литвиненко В.И. Бидюк П.И. Фефелов А.А., Баклан И.В. Программная реализация алгоритма отрицательного отбора для решения задач классификации //Тези доповідей Всеукраїнської конференції МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ, Дніпропетровськ,17-19 листопада 2003 р. с.10

16.Гамаюнов Д.Ю., Качалин А.И. Обнаружение атак на основе анализа переходов состояний распределенной системы // Искусственный интеллект. 2004. № 2, С.49-53.

17.Промежуточный научно-технический отчет по первому этапу НИР «Невод». – М.: ф-т ВМК МГУ, 2004.

 ISSN 1562-9945