УДК 622.276.001

М.Е. Бурлаков

Самарский национальный исследовательский университет им. С.П. Королева, Самара, Россия

БАЗОВЫЕ ПРИНЦИПЫ РАБОТЫ ЗАГРУЗЧИКА КОНФИГУРАЦИЙ В МНОГОУРОВНЕВОЙ СИСТЕМЕ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Рассматриваются базовые принципы работы загрузчика конфигураций в многоуровневой системе обнаружения вторжений. Определен генератор конфигураций как механизм по созданию унифицированных конфигураций с заданными параметрами конфигураций. Даны определения блоку по работе с базой данных, сформулировано понятие ядро обмена, а также описан провайдер конфигураций, как механизм унифицирующий процесс создания и загрузки конфигураций в многоуровневой системе обнаружения вторжений. Описаны основные режимы работы загрузчика конфигураций и даны подробные пояснения работы в режиме восстановления. Представлены программные визуализации общего функционала загрузчика конфигураций, процедуры формирования конфигурации в загрузчике конфигураций, процедуры загрузки конфигурации в загрузчике конфигураций. Описано строение базы данных конфигураций с механизмом анализа конфигураций и хранения истории конфигураций. Обозначен блок оценки эффективности конфигурации в рамках многоуровневой системы обнаружения вторжений.

Ключевые слова: загрузчик конфигураций, многоуровневая система обнаружения вторжений, генератор конфигурации, провайдер конфигураций, загрузчик параметров конфигураций.

M.E. Burlakov

Samara National Research University of S.P. Korolev, Samaram, Russian Federation

THE BASIC PRINCIPLES OF LOADER CONFIGURATION IN MULTI-LAYER INTRUSION DETECTION SYSTEM

In article the basic principles of the boot loader configuration in a multi-level intrusion detection system are viewed. The generator of configurations as a mechanism to create a standardized configuration to set configuration parameters is set. There are definitions of module for working with the database and exchange kernel. Exchange kernel is a mechanism which helps to create and upload the configurations to multi-level intrusion detection system. The basic modes of working the configuration loader are explained. The recovery mode is viewed in detail. Some software visualizations are presented. Such as: the visualization of basic functional of configuration loader, the visualization of procedures for creating configurations in configuration loader, the visualization of procedures for loading configurations in configuration loader. The scheme of configuration database with the mechanism for analyzing and storing the history of configurations is described. In multi-level intrusion detection system the block of evaluating the configuration effectiveness is determined.

Keywords: configuration loader, multi-layer intrusion detection system, configuration generator, configuration provider, loader of configuration parameters.

Введение. В настоящее время существует множество решений, обеспечивающих защиту данных в области целостности, доступности и конфиденциальности в рамках как локальных, так и глобальных сетей. К таковым механизмам можно отнести системы обнаружения и предотвращения вторжений (*Intrusion detection systems (IDS)* и *Intrusion prevention systems (IPS*)).

На сегодняшний день на рынке представлено множество решений класса обнаружения и предотвращения вторжений, каждое из которых обладает как преимуществами, так и недостатками [1, 2]. Основным способом выбора необходимого класса решения для защиты компьютерных сетей является его применимость на соответствующих уровнях модели *OSI*. Выделяют следующие типы систем обнаружения и предотвращения вторжений (далее – СОВ) [3]:

– узловая, предназначенная для работы на конечных хостах (компьютеры, серверы, и т.д.);

– сетевая, предназначенная для работы на сетевых устройствах, обеспечивающих непрерывную передачу информации между узлами (маршрутизаторы, файерволы и т.д.);

– гибридная (комбинация методов работы из узловой и сетевой).

В качестве механизмов классификации информации и ее валидации используется все множество как адаптивных [4–9], так и неадаптивных методов [10–14].

В случае если система использует один уровень модели *OSI* или предназначена для конкретного решения задачи в рамках одной логической модели, то обозначается, что она имеет одноуровневую типизацию или направленность, в противном случае считается, что система многоуровневая.

В работе [15] предлагается к рассмотрению базовая модель многослойной (многоуровневой) системы обнаружения вторжений, состоящая из следующих функциональных блоков:

– блок хранения единого реестра срабатываний;
– блок актуальных и инициализационных библиотек;
– блок логирования;
– панель управления;
– блок загрузчика конфигураций.

В данной статье рассматриваются базовые принципы работы одного из узлов многоуровневой системы обнаружения вторжений – загрузчика конфигураций.

1. Базовые аспекты загрузчика конфигураций. Загрузчик конфигураций (ЗК) – программно-аппаратное решение, обеспечивающее подготовку конфигураций для системы обнаружения и предотвращения вторжений с последующим их обновлением. Под конфигурацией понимается информационная структура, несущая в себе параметры настройки для какой-либо системы.

Загрузчик конфигураций обладает функцией считывания текущей конфигурации СОВ с возможностью последующего ее сохранения в базе данных конфигурации. Структурная схема загрузчика конфигураций представлена на рис. 1.

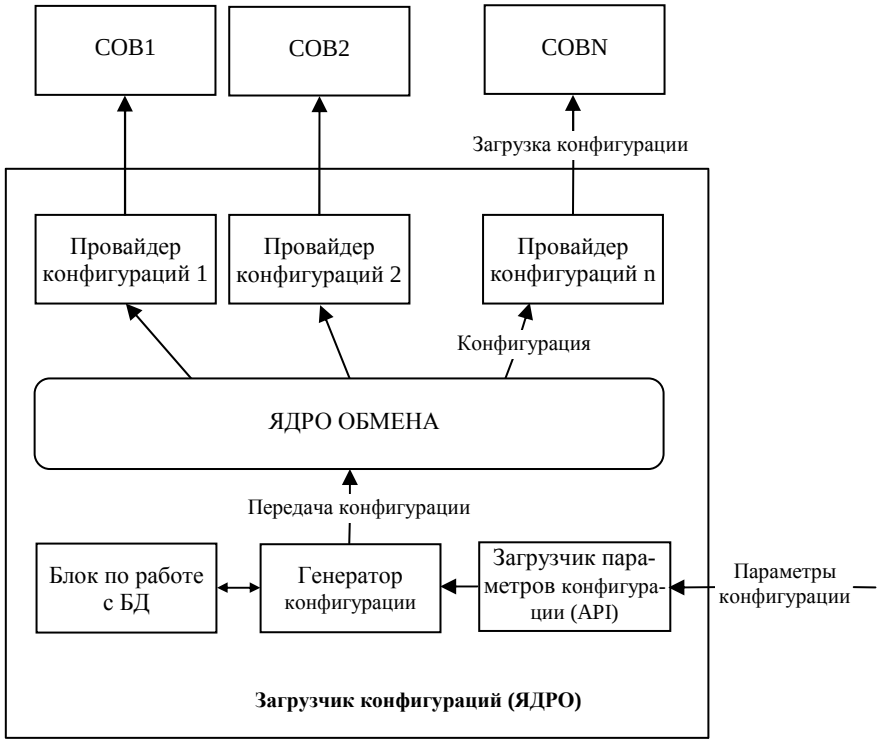

Рис. 1. Структурная схема загрузчика конфигураций

В ядро загрузчика конфигурации входят следующие механизмы.

Параметры конфигурации – содержимое какой-либо конфигурации.

Загрузчик параметров конфигурации (*API*[1]) – механизм, обеспечивающий возможность загрузки полученных параметров конфигурации в загрузчик конфигурации.

Генератор конфигурации – механизм, позволяющий создавать унифицированные конфигурации, полученные из загрузчика конфигурации.

Блок по работе с базой данных (БД) – механизм, обеспечивающий хранение и версионность конфигураций для систем обнаружения вторжений.

Ядро обмена – механизм, обеспечивающий формирование, валидацию и дальнейшую обработку конфигураций, полученных от генератора конфигураций. Именно ядро обмена отвечает за определение механизма преобразования конфигурации с использованием провайдера конфигураций.

Провайдер конфигураций – механизм, позволяющий из унифицированных конфигураций формировать конфигурации, пригодные для работы с соответствующими СОВ. В силу реализации загрузчика конфигураций в рамках многоуровневой СОВ требуется ориентация работы механизма для различных протоколов. Таким образом, для разных уровней модели *OSI* будут работать разные системы обнаружения вторжений с разными структурами конфигураций. Иными словами, например, для СОВ, работающей на прикладном уровне (*HTTP*-сервер), набор конфигураций будет один. Для СОВ, работающей на сетевом уровне (*CISCO*), набор конфигураций будет другим.

Основной функционал загрузчика конфигураций включает в себя:

1. Считывание конфигурации с системы обнаружения вторжений;
2. Запись конфигурации в систему обнаружения вторжений;
3. Считывание конфигурации из базы данных конфигураций;
4. Запись конфигурации в базу данных конфигураций.

Корректность конфигурации СОВ загрузчик конфигурации не контролирует. Однако это не исключает возможности наличия валидирующих механизмов.

[1] Интерфейс программирования приложений (иногда интерфейс прикладного программирования, *API*) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах [16].

В загрузчик конфигураций включен функционал по внешнему управлению, который позволяет:

1. Генерировать конфигурации для систем обнаружения вторжений согласно полученным от сторонних систем или оператора параметрам. Параметры могут быть присланы как оператором, так и сторонней системой. Загрузчик конфигураций оснащен API, позволяющим на основании присланных данных (например, путем *XML*-обмена) сформировать соответствующую конфигурацию для конкретной системы обнаружения вторжений.

2. Автоматически загружать поступившие конфигурации в системы обнаружения вторжений. Наличие данной опции позволяет после того, как конфигурация сгенерирована, автоматически обновить ее для соответствующей системы обнаружения вторжений. Ручное обновление конфигурации оператором через загрузчик конфигураций также присутствует.

3. Восстанавливать предыдущие конфигурации систем обнаружения вторжений (так называемый механизм версионности конфигураций). Поскольку база данных конфигураций хранит в себе все версии конфигурации для конкретной системы обнаружения вторжений, в загрузчике конфигурации присутствует механизм ручного восстановления конфигурации для соответствующей системы обнаружения вторжений.

Основные этапы работы загрузчика конфигураций включают в себя:

1. Прием параметров новой конфигурации в загрузчик параметров конфигураций (API);

2. Передача данных в генератор конфигураций и параллельное сохранение новых параметров конфигурации в базу данных конфигураций;

3. Создание исходной конфигурации для системы обнаружения вторжений и определение с помощью ядра обмена типа и параметров загрузки конфигурации в систему обнаружения вторжений. Выбор провайдера конфигураций.

4. Загрузка новой конфигурации в систему обнаружения вторжений.

2. Загрузчик конфигураций в режиме восстановления. Дополнительной характеристикой функционала загрузчика является возможность восстановления конфигурации для любой системы обнаружения вторжений. Общая структурная схема загрузчика конфигурации в режиме восстановления представлена на рис. 2.

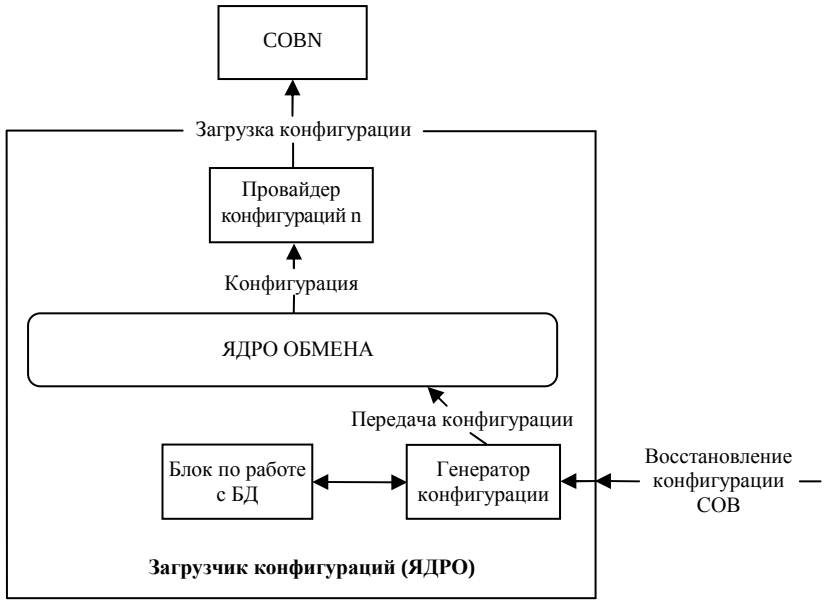

Рис. 2. Структурная схема загрузчика конфигурации
в режиме восстановления конфигурации

Режим восстановления несет в себе ряд отличительных от стандартной работы моментов. Основные шаги при восстановлении конфигурации СОВ загрузчиком конфигураций включают в себя:

1. Получение команды на восстановление конфигурации для конкретной СОВ. Получение команды предусмотрено как от оператора, так и от сторонней системы. Работа с режимом восстановления также возможна и через *API*.

2. Обращение генератора конфигурации к базе данных конфигураций с соответствующим запросом версии конфигурации для выбранной системы обнаружения вторжений.

3. Формирование конфигурации для системы обнаружения вторжений средствами генератора конфигурации и передача в ядро обмена загрузчика конфигурации.

4. Обновление/восстановление конфигурации системы обнаружения вторжений средствами провайдера конфигурации, определенного ядром конфигурации.

На основании вышеописанного рассмотрим программную визуализацию работы загрузчика конфигураций, согласно которой была выполнена программная реализация данного компонента системы.

3. Программная визуализация работы загрузчика конфигураций. Программная реализация загрузчика конфигураций для системы обнаружения вторжений выполняет следующие задачи:

1. Формирование конфигурации для системы обнаружения вторжений. Программная реализация дает возможность оператору или сторонней системе, используя заложенное в нее *API*, корректно формировать для соответствующей СОВ конфигурацию.

2. Загрузка конфигурации в СОВ. В заложенную функциональность включена возможность автоматической или ручной загрузки сформированной конфигурации в СОВ.

3. Восстановление конфигурации системы обнаружения вторжений. Оператору или сторонней системе дается возможность управлять всеми версиями конфигураций выбранной СОВ, в том числе и процессом восстановления.

Программная визуализация общего функционала загрузчика конфигураций должна пошагово выполнять следующие действия:

1) проверку на непустоту конфигурации и определенность принадлежности загружаемой конфигурации конкретной системе обнаружения вторжений (директивная валидация);

2) определение режима восстановления, т.е. в случае, если процедуре была отдана команда на восстановление, система выполняет ряд заложенных и специально определенных для этого действий;

3) процесс формирования конфигурации (провайдер конфигураций);

4) процесс сохранения конфигураций в базе данных конфигураций.

Программная визуализация общего функционала загрузчика конфигураций представлена далее.

Процедура формирования конфигурации состоит из следующих шагов:

1) извлечение правил формирования конфигураций для конкретно выбранной системы обнаружения вторжений;

2) создание (генерация) файла конфигурации с форматом, отвечающим конкретно выбранной системы обнаружения вторжений.

Программная визуализация процедуры формирования конфигурации в загрузчике конфигураций представлена ниже.

Процедура загрузки конфигурации в загрузчике конфигураций (провайдер конфигураций) состоит из следующих шагов:

1) извлечение правил формирования конфигураций для конкретно выбранной системы обнаружения вторжений;

2) формирование конфигурации и последующая ее загрузка в конкретную систему обнаружения вторжений.

Программная визуализация общего функционала загрузчика конфигураций

```
ПРОГРАММА ЗагрузчикКонфигура-
ции(Конф,ИДСОВ,ИДКонф="",Восстановить=0,БДКонф=1)
    ЕСЛИ  длинаСтроки(Конф)!=0 и число(ИДСОВ)!=0
          Конфигурация = СформироватьКонфигурацию(Конф, ИДСОВ)
          ЕСЛИ ЗагрузитьКонфигурацию(Конфигурация, ИДСОВ) = ЛОЖЬ
                ВЕРНУТЬ Ошибка(ИДСОВ)
          ЕСЛИ БДКонф=1
                СохранитьКонфигурацию(Конфигурация, ИДСОВ)
    ВОЗВРАТ
    ЕСЛИ длинаСтроки(ИДКонф)!=0 и Восстановить=1
          Конфигурация и ИДСОВ = ВосстановитьКонфигурацию(ИДКонф)
          ЗагрузитьКонфигурацию(Конфигурация, ИДСОВ)
    ВОЗВРАТ
```

Краткий список сокращений

Сокращение	**Обозначение**
ИДСОВ	Идентификатор системы обнаружения вторжений. Необходим для создания точного соответствия конфигурации конкретной системе обнаружения вторжений
Конф	Загружаемая для конкретной системе обнаружения вторжений конфигурация
ИДКонф	Идентификатор уже загруженной конфигурации, по умолчанию, пустая строка
Восстановить	Флаг, где 0-отключен режим восстановления,1 – восстановить конфигурацию для конкретной системы обнаружения вторжений
БДКонф	Флаг, отвечающий за сохранение конфигурации в базу данных конфигураций

Программная визуализация процедуры формирования конфигурации в загрузчике конфигураций

ПРОГРАММА СформироватьКонфигурацию(Конф,ИДСОВ)

1. Согласно ИДСОВ извлекаются правила формирования конфигураций
2. Из извлеченных правил и Конф, строится файл конфигурации, по формату соответствующий ИДСОВ

ВЕРНУТЬ Конфигурация

Краткий список сокращений

Сокращение	**Обозначение**
ИДСОВ	Идентификатор системы обнаружения вторжений. Необходим для создания точного соответствия конфигурации конкретной системе обнаружения вторжений
Конф	Загружаемая для конкретной системе обнаружения вторжений конфигурация

Ниже представлена программная визуализация процедуры загрузки конфигурации:

ПРОГРАММА ЗагрузитьКонфигурацию(Конф, ИДСОВ)

1. Согласно ИДСОВ извлекаются правила загрузки конфигураций
2. Из извлеченных правил и файла конфигураций (Конф), производится загрузка Конф в ИДСОВ

ЕСЛИ загрузка прошла успешно **ВЕРНУТЬ ИСТИНА**

ВЕРНУТЬ ЛОЖЬ

Краткий список сокращений

Сокращение	**Обозначение**
ИДСОВ	Идентификатор системы обнаружения вторжений. Необходим для создания точного соответствия конфигурации конкретной системе обнаружения вторжений
Конф	Загружаемая для конкретной системе обнаружения вторжений конфигурация

Таким образом, была проведена программная визуализация всех компонентов загрузчика конфигураций, которая использовалась в дальнейшей для практической реализации.

Отдельным компонентом следует рассмотреть базу данных конфигураций.

4. База данных конфигураций. База данных конфигураций предназначена для хранения всех конфигураций каждой системы обнаружения вторжений (механизм версионности). Общая структура базы данных конфигураций представлена в виде таблицы.

Структура базы данных конфигураций

СОВ(i)	КонфигурацияN	ДатаN
	…	…
	Конфигурация2	Дата2
	Конфигурация1	Дата1
СОВ(i+1)	КонфигурацияN	ДатаN
	…	…
	Конфигурация2	Дата2
	Конфигурация1	Дата1

Примечание: СОВ(i) – i-я система обнаружения вторжений в рамках многоуровневой системы обнаружения вторжений.

Изначально база данных конфигураций не является пустой. В ней хранятся первоначальные конфигурации, заложенные в блоке инициализационных библиотек. База данных конфигураций выполняет следующие задачи:

1. **Анализ конфигураций**, может проводиться как в автоматическом режиме методом сравнения соответствующих конфигурационных настроек, так и вручную оператором, либо сторонней системой. Цель анализа конфигураций – получение наиболее оптимальных и эффективных конфигураций при работе в уровне передачи информации системы обнаружения вторжений.

2. **Хранение истории конфигураций**, обеспечивает загрузчик конфигураций необходимыми данными. В случае, если в системе обнаружения вторжений произойдет технический сбой, не связанный с работой алгоритма, загрузчик конфигураций сможет восстановить с помощью механизма версионности самую последнюю (по дате добавления) конфигурацию. При контакте загрузчика конфигурации с базой данных конфигурации сначала происходит сохранение конфигурации и только потом ее дальнейшая запись в систему обнаружения вторжений.

Загрузчик конфигураций и база данных конфигураций тесно связаны с блоком оценки эффективности конфигурации конкретно взятой системы обнаружения вторжений.

5. Блок оценки эффективности конфигурации СОВ. Данный блок необходим для оценки эффективности полученной или имеющейся конфигурации, хранимой в базе данных конфигураций. Эффективность конфигурации может быть оценена либо по сравнению с уже имеющейся конфигурацией, либо определенными оператором минимальными необходимыми коэффициентами. Операция сравнения конфигураций для конкретной системы обнаружения вторжений заключается в выборке тестового набора данных из *Log* блока и дальнейшего применения этих данных в режиме тестирования для каждой из конфигураций.

Алгоритм сравнения эффективности одной конфигурации относительно другой ранее загруженной конфигурации, либо относительно заданных оператором характеристик определяется через единый реестр срабатываний и через базу данных конфигураций. Для сравнения конфигураций используется механизм эмуляции, который позволяет применять конкретную конфигурацию для конечного множества векторов угроз в выбранной системе обнаружения вторжений.

Алгоритм определения эффективности конфигураций для конкретной системы обнаружения вторжений задается следующим образом:

1. Из базы банных конфигурации выделяются две или более конфигурации (K_{11}, K_{12}, K_{13}, …, K_{1N}) для выбранной СОВ (СОВ1), а также время их записи в БД конфигурации.

2. Из единого реестра срабатываний выделяется конечное множество угроз (M_i), выявленных СОВ с конкретной конфигурацией в соответствующий промежуток времени.

3. Формируется общее множество векторов угроз ($M = \cap_i M_i$). Используя режим эмуляции, оператор или сторонняя система «прогоняет» это множество через каждую конфигурацию ($K_{1i}(M) = k_{1i}$).

4. Наиболее оптимальной будет признана та конфигурация, значение k_{1i} которой будет максимальным. Данное значение, иными словами, будет означать, что K_{1i}, является наиболее эффективной конфигурацией в данный момент времени, обеспечивающий наиболее высокие показатели обнаружения ранее выявленных угроз.

В случае, если, например, в многоуровневую систему обнаружения вторжений добавляется новая система обнаружения вторжений, работающая с теми же протоколами и на том же уровне *OSI*, загрузка наиболее оптимальной конфигурации происходит в автоматическом режиме.

Выводы

Таким образом, описаны базовые принципы функционирования элемента многоуровневой системы обнаружения – загрузчика конфигураций. Даны базовые понятия и определения структурных элементов, из которых он состоит. Определены его режимы работы: стандартный и восстановления. Дана программная визуализация основных компонент.

Наличие загрузчика конфигураций в многоуровневой системе обнаружения вторжений позволяет построить отказоустойчивую систему с режимами автоматической коррекции ее качества работы.

Библиографический список

1. IDS/IPS – Системы обнаружения и предотвращения вторжений [Электронный ресурс] // Net.Config Сетевые технологии. – 2014. – URL: http://netconfig.ru/server/ids-ips/ (дата обращения: 01.07.2016).

2. Дрозд А. Обзор корпоративных IPS-решений на российском рынке [Электронный ресурс] // Anti-Malware. – 2013. – URL: http://www.anti-malware.ru/IPS_russian_market_review_2013 (дата обращения: 30.06.2016).

3. Гузаиров М.Б., Машкина И.В. Управление защитой информации на основе интеллектуальных технологий // Машиностроение. – 2013. – С. 50–66.

4. Vacca J.R. Computer and Information Security Handbook // Newnes. – 2012. – С. 334–335.

5. Borger E. The Abstract State Machines Method for High-Level System Design and Analysis / Dipartimento di Informatica, Universita di Pisa, 2007. – P. 35–37.

6. Shim J.K. Information Systems and Technology for the Noninformation Systems Executive // CRC Press. – 2000. – P. 233–235.

7. Lunt T.F., Tamaru A., Gilham F. A real-time intrusion-detection expert system (IDES) // Final Technical Report. – 1992. – P. 13–14.

8. Zhou J. Alert Reduction for Network Intrusion Detection // ProQuest. – 2008. – P. 42–43.

9. Bellovin S.M., Gennaro R. Applied Cryptography and Network Security // Springer Science & Business Media. – 2008. – С. 230–233.

10. Васильев В.И. Интеллектуальные системы защиты информации. – М.: Машиностроение, 2012. – С. 165–171.

11. Nunes L., Timmis J. Artificial Immune Systems: A New Computational Intelligence Approach // Springer Science & Business Media. – 2002. – P. 2–10.

12. Хайкин С. Нейронные сети: полный курс. – 2-е изд. ред. – М.: Вильямс, 2008. – 1104 с.

13. Abe S. Support Vector Machines for Pattern Classification // Springer Science & Business Media. – 2005. – С. 39–41.

14. Kollias S. Artificial Neural Networks // Springer Science & Business Media. – 2006. – С. 159–161.

15. Бурлаков М.Е. Двухклассификационная искусственная иммунная система // Вестник Самар. гос. ун-та. – 2014. – № 7(118). – С. 207–221.

16. Hofmeyr S.A., Forrest S. Immunity by Design: An Artificial Immune System [Электронный ресурс] // University of New Mexico, 2011. – URL: http://www.cs.unm.edu/~immsec/publications/gecco-steve.pdf (дата обращения: 02.07.2016).

References

1. IDS/IPS – Sistemy obnaruzheniia i predotvrashcheniia vtorzhenii [About an intrusion and prevention detection systems] *Net.Config Setevye tekhnologii*, 2014, available at: http://netconfig.ru/server/ids-ips/ (accessed 01 July 2016).

2. Drozd A. Obzor korporativnykh IPS-reshenii na rossiiskom rynke [An overview of IPS solutions in Russia]. *Anti-Malware*, 2013, available at: http://www.anti-malware.ru/IPS_russian_market_review_2013 (accessed 30 June 2016).

3. Guzairov M.B., Mashkina I.V. Upravlenie zashchitoi informatsii na osnove intellektual'nykh tekhnologii [A manage of information security using intellectual technologies]. *Mashinostroenie*, 2013, pp. 50-66.

4. Vacca J.R. Computer and Information Security Handbook. *Newnes*, 2012, pp. 334-335.

5. Borger E. The Abstract State Machines Method for High-Level System Design and Analysis. *Dipartimento di Informatica, Universita di Pisa*, 2007, pp. 35-37.

6. Shim J. K. Information Systems and Technology for the Noninformation Systems Executive. *CRC Press*, 2000, pp. 233-235.

7. Lunt T.F., Tamaru A., Gilham F. A real-time intrusion-detection expert system (IDES). *Final Technical Report*, 1992, pp. 13-14.

8. Zhou J. Alert Reduction for Network Intrusion Detection. *ProQuest*, 2008, pp. 42-43.

9. Bellovin S.M., Gennaro R. Applied Cryptography and Network Security. *Springer Science & Business Media*, 2008, pp. 230-233.

10. Vasil'ev V.I. Intellektual'nye sistemy zashchity informatsii [Intellectual systems of information security]. Moscow: Mashinostroenie, 2012, pp. 165-171.

11. Nunes L., Timmis J. Artificial Immune Systems: A New Computational Intelligence Approach. *Springer Science & Business Media*, 2002, pp. 2-10.

12. Khaikin S. Neironnye seti [Neural networks]. Moscow: Vil'iams, 2008. 1104 p.

13. Abe S. Support Vector Machines for Pattern Classification. *Springer Science & Business Media*, 2005, pp. 39-41.

14. Kollias S. Artificial Neural Networks. *Springer Science & Business Media*, 2006, pp. 159-161.

15. Burlakov M.E. Dvukhklassifikatsionnaia iskusstvennaia immunnaia sistema [Two-classification artificial immune system]. *Vestnik Samarskogo gosudarstvennogo universiteta*, 2014, no. 7(118), pp. 207-221.

16. Hofmeyr S.A., Forrest S. Immunity by Design: An Artificial Immune System. *University of New Mexico*, 2011, available at: http://www.cs.unm.edu/~immsec/publications/gecco-steve.pdf (accessed 02 July 2016).

Сведения об авторе

Бурлаков Михаил Евгеньевич (Самара, Россия) – лаборант кафедры безопасности информационных систем Самарского национального исследовательского университета им. академика С.П. Королева (443086, Самара, Московское шоссе, 34, e-mail: knownwhat@gmail.com).

About the author

Mikhail E. Burlakov (Samara, Russian Federation) is a Labour in Department of information security systems Samara National Research University of S.P. Korolev (443086, Samara, 34, Moskovskoye Shosse, e-mail: knownwhat@gmail.com).

Получено 14.07.2016