Вестник УГАТУ
Уфа : УГАТУ, 2012 Т. 16, № 3 (48). С. 198–204

УДК 004.2

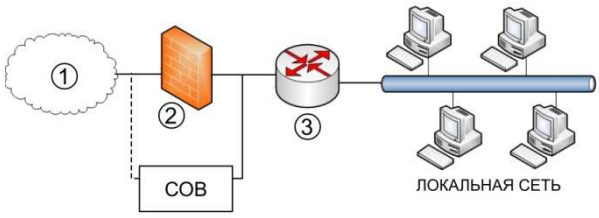
В. Д. Котов, В. И. Васильев

СОВРЕМЕННОЕ СОСТОЯНИЕ ПРОБЛЕМЫ ОБНАРУЖЕНИЯ СЕТЕВЫХ ВТОРЖЕНИЙ

Описаны современные подходы к обнаружению сетевых вторжений и перспективных направлений их развития. Дана классификация систем обнаружения вторжений, а также технологий, лежащих в их основе. В работе большое внимание уделяется передовым направлениям исследований в данной области. Поднимается проблема отсутствия вычислительных моделей обнаружения атак и формального обоснования. *Системы обнаружения вторжений*

ВВЕДЕНИЕ

Обнаружение вторжений – процесс мониторинга событий в компьютерной системе или сети и анализа их на предмет нарушений политики безопасности [1]. Система обнаружения вторжений (СОВ) – это устройство или программное решение, осуществляющее обнаружение вторжений (рис. 1).



1. Интернет
2. Межсетевой экран
3. Маршрутизатор

Рис. 1. Место СОВ в сетевой инфраструктуре (один из вариантов)

Типичными примерами атак, мониторинг которых осуществляет СОВ, являются:

- внедрение вредоносного кода;
- исчерпание полосы пропускания путем большого количества соединений;
- подбор пароля;
- сетевая активность троянских коней, червей и вирусов;
- и т. д.

Впервые концепция обнаружения вторжений была предложена в техническом отчете Дж. Андерсена [2] в 1980 г. Первое поколение СОВ осуществляло анализ системных журналов событий на предмет злоупотреблений и нарушений.

Контактная информация: 8(347) 273-06-72

Второе поколение СОВ началось со статьи Д. Деннинг «An Intrusion Detection Model» [3], описывающей экспертную систему обнаружения вторжений. Внимание уделяется таким аспектам, как профили нормальной активности системы, статистический анализ данных и т.д.

Третьему поколению систем обнаружения вторжений, по мнению многих, еще не наступило, мы пока только переходим к нему. Основные проблемы СОВ последнего поколения – это анализ данных в режиме реального времени, адаптивность, способность осуществлять активный ответ на атаку.

В данной статье приводится классификация систем обнаружения вторжений, подробно рассмотрены различные техники детектирования сетевых атак, а также представлены перспективные направления развития СОВ.

ТАКСОНОМИЯ СИСТЕМ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Классификацию систем обнаружения вторжений можно провести по многим критериям. Основные подходы к классификации представлены ниже в списке.

По типу объекта мониторинга:

- узловые СОВ – осуществляют мониторинг активности одного узла в сети;
- сетевые СОВ – объектом мониторинга является сетевой сегмент.

По архитектуре:

- централизованные – все вычисления совершаются на одной рабочей станции;
- распределенные – система состоит из нескольких элементов: сенсоров, разнесенных по сети, вычислительного центра, а также консоли администратора.

По технологии анализа:

- без сохранения состояния – каждое событие рассматривается независимо от других;

- с сохранением состояния – информация о предыдущих событиях сохраняется и учитывается при принятии решения.

По методу обнаружения атак:

- системы обнаружения злоупотреблений – осуществляют поиск шаблонов известных атак в сетевом трафике или высокоуровневых данных;
- системы обнаружения аномалий – обладают профилем нормальной активности системы и детектируют отклонения от него;
- системы обнаружения нарушений в протоколе – данный тип СОВ следит за корректностью соблюдения протоколов сетевого взаимодействия и фиксирует нарушения.

По способу реагирования:

- пассивные – во время инцидента подается сигнал тревоги и вносится запись в журнал событий;
- активные – осуществляют активный ответ (например, сбрасывают соединение, блокируют IP адрес и т. п.).

Современные системы обнаружения вторжений, как правило, активные, распределенные, обеспечивающие мониторинг сети в режиме, близком к режиму реального времени. Наиболее важным критерием классификация сегодня является метод обнаружения атак. Рассмотрим их более подробно в следующих разделах.

УЗЛОВЫЕ И СЕТЕВЫЕ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Узловые СОВ (УСОВ) детектируют атаки, направленные на определенный узел сети. Они зависимы от операционной системы. Согласно [4], узловые СОВ должны выполнять хотя бы одну из следующих функций:

- мониторинг файловой системы – проверка целостности файлов и директорий;
- анализ журналов событий – поиск подозрительных шаблонов в журналах событий;
- мониторинг сетевых соединений – анализ входящих и исходящих сетевых соединений на предмет нарушений политики безопасности;
- анализ на уровне ядра операционной системы – мониторинг системы с наивысшими привилегиями, что позволяет блокировать любые операции.

Мониторинг файловой системы подразумевает сбор информации о файлах и директориях (данные заголовков, сигнатуры различных форматов и т. п.), вычисление хеш-функций (например, MD5), данные о владельце и правах доступа и т. д. При несанкционированной модификации файлов или другом нарушении, подается сигнал тревоги.

Анализ журналов событий является также важной функцией узловых СОВ. Поиск шаблонов атак может осуществляться как в основном журнале операционной системы, так и в логах приложений. Так, например, web-сервер, как правило, вносит в журнал заголовки запросов, что позволяет впоследствии обнаруживать в них признаки SQL-инъекций или XSS.

Мониторинг сетевых соединений очень полезен для выявления подозрительной активности вредоносных программ. Например, исходящие соединения по протоколам IRC, SSH, Telnet могут быть признаками наличия сетевого червя или бэкдора. Входящие соединения могут свидетельствовать о несанкционированно запущенном серверном приложении, что тоже может оказаться симптомом заражения системы.

Эффективной реализацией УСОВ является соответствующий драйвер или модуль ядра операционной системы. Такие системы сложно обойти, поскольку они обладают всеми возможными привилегиями, а потому способны блокировать атаки.

Сетевые системы обнаружения вторжений (ССОВ) предназначены для детектирования атак в сегменте сети. Источником информации для них является сетевой трафик, заголовки и содержимое сетевых пакетов.

Современные ССОВ являются распределенными и состоят из следующих компонентов [1]:

- сенсор – компонент, ответственный за сбор данных и их первичный анализ;
- управляющий блок – компонент, собирающий данные с сенсоров и осуществляющий их вторичный анализ и корреляцию;
- база данных – как правило, выполнена в виде отдельного сервера, представляет собой хранилище информации о событиях;
- консоль – интерфейс управления системой.

Первичный анализ, осуществляемый сенсором, включает в себя поиск шаблонов известных атак и информации о событиях и соединениях в отведенном ему сегменте.

Управляющий блок получает информацию от сенсоров и на ее основе принимает решения о том, происходит ли в данный момент атака, а также осуществляет корреляцию полученных

данных и их анализ на предмет распределенных атак.

Сервер базы данных играет важную роль не только как хранилище информации, генерируемой сенсорами, но и как средство расследования инцидентов и поиска информации о похожих атаках.

СИСТЕМЫ ОБНАРУЖЕНИЯ ЗЛОУПОТРЕБЛЕНИЙ

Системы обнаружения злоупотреблений (СОЗ) осуществляют поиск шаблонов известных атак в сетевом трафике или высокоуровневых данных. Первые СОЗ были основаны на сигнатурах – последовательностях байт в сетевых пакетах, которые идентифицировали шелл коды или форматы данных сетевых червей и вирусов.

Рис. 2. Обнаружение NOP цепочки в сетевом пакете

На рис. 2 схематически изображена структура полезной нагрузки эксплойта – последовательности инструкций, эксплуатирующих уязвимость и выполняющую произвольный код на компьютере жертвы. Простые шелл коды, как правило, содержат NOP-последовательность. NOP – это инструкция на языке ассемблера, название которой расшифровывается как no operation. Иными словами эта инструкция означает простой процессора в текущем такте. Машинная инструкция команды NOP кодируется в шестнадцатиричной системе как 0x90. Последовательность из нескольких байт 0x90 может являться сигнатурой и быть индикатором попытки внедрения кода.

Современные СОЗ используют более сложные способы обнаружения атак. Например, известная СОВ с открытым исходным кодом Snort, оперирует правилами, обладающими следующей структурой [5]:

Заголовок правила:

- действие при активации сигнатуры;
- протокол (IP, ICMP и т. д.);
- информация об источнике;
- информация о приемнике.

Тело правила:

- последовательность байт (сигнатура);
- направление передачи данных;
- порты;
- флаги TCP;
- и т. д.

Важной особенностью современных СОЗ является то, что они используют комплексные правила активации. Срабатывание некоторых правил возможно только при наличии дополнительных условий, например, активации другого правила. Это позволяет более гибко описывать возможные атаки и снизить уровень ложных срабатываний.

Одним из существенных недостатков систем обнаружения злоупотреблений является тот факт, что при анализе они не принимают во внимание топологию сети. Таким образом, существует большое количество способов обойти СОЗ. Примером может служить манипуляция полем TTL (time to live) в сетевых пакетах. Как известно, маршрутизаторы, перенаправляя сетевой пакет, уменьшают значение TTL и, когда оно становится равным нулю, – пакет отбрасывается. Прием обхода СОЗ с помощью этого поля показан на рис. 3.

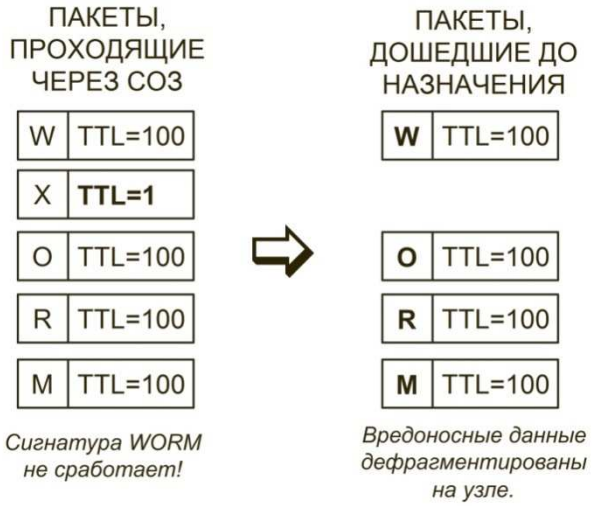


Рис. 3. Пример манипуляции значением поля TTL

Таким образом, одним из важнейших направлений исследования в области обнаружения злоупотреблений является интеграция СОЗ в топологию сети.

К преимуществам СОЗ можно отнести:

- высокую эффективность обнаружения известных атак;

- относительно простую настройку, нет надобности дополнительно обучать такие системы.

Недостатками сигнатурных систем обнаружения вторжений являются:

- необходимость в обновлении базы шаблонов атак;
- разрастание количества сигнатур и снижение производительности;
- неспособность детектировать неизвестные атаки.

Основной проблемой СОЗ является высокая частота появления новых вредоносных программ, в связи с чем приходится генерировать большое количество новых сигнатур. При таких темпах роста вредоносной активности число сигнатур будет расти быстрее, чем производительность вычислительной техники, что значительно снизит эффективность детектирования злоупотреблений.

Современный тренд развития СОЗ направлен на сигнатуры уязвимости (СУ) [18, 19]. Концепция СУ позволяет взглянуть на проблему обнаружения злоупотреблений в другом ракурсе. Вместо того, чтобы детектировать вредоносный код (или вредоносные действия), предлагается детектировать попытку эксплуатации уязвимости. Рассмотрим пример (рис. 4). Положим, существует приложение А, которое принимает по сети некие данные от приложения В. Положим также, что в приложении А существует уязвимость переполнения буфера и, если пользователь приложения В отправит данные в объеме, превышающем 40 байт и содержащие вредоносные инструкции, то он сможет произвести запись в стек и выполнить этот код.

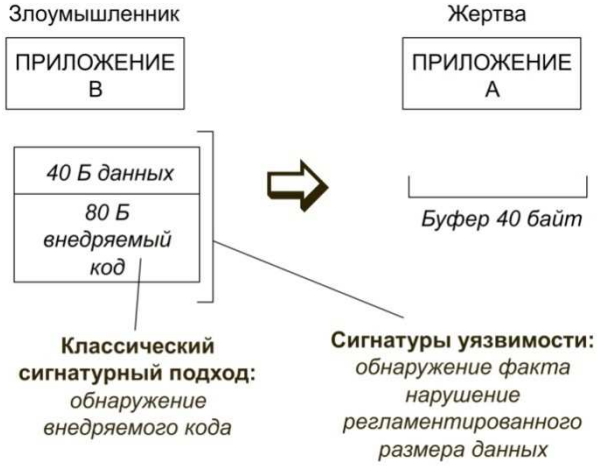


Рис. 4. Иллюстрация принципа работы сигнатур уязвимости

В случае обычных сигнатур детектируется именно вредоносный код – последовательность байт, идентифицирующая его максимально точно. Однако стоит злоумышленнику поменять код или изменить его путем обфускации – сигнатура перестает работать. Теперь, чтобы обнаружить новый код, необходимо писать новую сигнатуру. В отличие от классического подхода, сигнатуры уязвимости детектируют сам факт попытки эксплуатировать уязвимость. В нашем примере сигнатура уязвимости определяет, что размер передаваемых данных больше, чем в действительности обрабатывает приложение А. При этом злоумышленник может как угодно менять код, шифровать его или маскировать другими способами – ему не удастся обойти эту проверку.

На сегодняшний день сигнатуры уязвимостей являются самым надежным способом описания атак и детектирования их. Однако это только в теории. На практике реализовать этот подход пока очень трудно по следующим причинам:

- сложность обнаружения уязвимостей в приложении, поскольку они зачастую являются неочевидными;
- не существует успешной формальной модели описания уязвимости.

СИСТЕМЫ ОБНАРУЖЕНИЯ АНОМАЛИЙ

Системы обнаружения аномалий (СОА) «знают», каким должно быть нормальное поведение контролируемой сети, и реагируют на отклонения от профиля нормальной активности. Для осуществления такого типа мониторинга СОА сначала необходимо «обучить». В основе СОА, как правило, лежит алгоритм классификации (или кластеризации), позволяющий выделить аномалии в общем потоке информации. Для этого необходимо сгенерировать обучающую выборку, содержащую как нормальные данные, циркулирующие в сети или системе в обычном рабочем режиме, так и данные, содержащие информацию об атаках. Причем последние должны быть четко определены и произведены в объеме, сравнимом с нормальными данными. Очевидно, что создание обучающей выборки для СОА является весьма сложной и дорогостоящей задачей. Поэтому на практике такой подход реализуется лишь частично – строится только профиль нормальной активности в виде набора параметров, а также интервалы, в пределах которых эти параметры считают-

ся нормальными. Выход за интервалы считается аномалией. Это, однако, порождает большое количество ложных срабатываний, поэтому в реальных приложениях никогда не делается ставка на один только алгоритм обнаружения аномалий.

Согласно [6], системы обнаружения аномалий можно классифицировать по трем критериям:

- используемый алгоритм;
- анализ отдельных пакетов или соединений;
- анализ заголовков или содержимого пакетов.

Первый критерий играет важнейшую роль в обнаружении аномалий, поскольку определяет механизм их детектирования. Второй и третий критерии являются вторичными и определяют лишь способы представления данных.

Для реализации СОА необходимо в первую очередь формализовать данные (будь то сетевой трафик или высокоуровневые данные). Для этого вводится понятие «событие», являющееся минимальной единицей данных, идущих на вход алгоритма обнаружения аномалий.

Одной из основных работ на эту тему является публикация [7]. Авторы в качестве события предлагают использовать вектор из 41 элемента, описывающий сетевое соединение. Следует оговорить, что касаемо протоколов, не ориентированных на соединение (UDP, ICMP, ARP), событие рассматривается как сетевое соединение длительностью 0 секунд, в случае же протокола TCP, термин «соединение» следует понимать в обычном смысле.

Все параметры разделены на три группы:

- внутренние параметры – данные, полученные из заголовков пакетов, такие, как число указателей срочности или флаги TCP;
- параметры содержимого – сюда входят такие показатели, как количество полученных сеансов суперпользователя, попыток авторизации, создания файлов и т. п.
- параметры трафика – к этой категории относятся параметры, полученные с помощью скользящего окна в две секунды, это, например, число соединений к одному узлу или порту.

Авторы постарались учесть как можно больше признаков, которые могли бы адекватно отличить нормальный трафик от атак. Например, большое число соединений к одному и тому же серверу может сигнализировать об атаке типа отказ в обслуживании.

Для обнаружения аномалий на основе статистических методов используются в основном следующие подходы:

Алгоритмы кластеризации – кластеризация подразумевает разбиение множества данных на классы по определенному признаку, для этого используются различные метрики, такие, как расстояние Евклида, Хэмминга и т. п. Наиболее популярным методом кластеризации является метод K-средних [8];

Марковские модели (ММ) – использование ММ в обнаружении вторжений впервые было предложено в [3]. В узловых системах обнаружения вторжений ММ, как правило, применяются для моделирования последовательностей действий пользователей [3], в сетевых СОВ – для моделирования последовательности полей в сетевых пакетах (например, TCP флагов [9]);

Вейвлет анализ – сущность подхода заключается в использовании вейвлет преобразования применительно к анализируемым данным [10].

Кроме статистических методов обнаружения атак, в исследовательских работах часто применяются технологии искусственного интеллекта:

Нейронные сети – наиболее часто используемым типом сети является многослойный персептрон [11]. В качестве формата представления данных в большинстве работ используется способ, описанный в [7].

Искусственные иммунные системы – в последнее время часто применяются попытки реализовать механизмы иммунной системы человека в вычислительной среде, такие, как отрицательный отбор и клональная селекция. Существуют как простые подходы, использующие один аспект иммунного ответа [12, 13], так и более сложные, имитирующие несколько механизмов человеческого иммунитета [14, 15].

Иммунокомпьютинг – концепция машинного обучения, основанная на моделировании взаимодействия протеинов. Ключевой моделью является формальная иммунная сеть. Данный подход применяется как для обнаружения сетевых атак в широком смысле [16], так и для обнаружения более специфических атак, например, направленных на web-сервер [17].

К достоинствам систем обнаружения аномалий можно отнести следующие:

- способность детектировать неизвестные атаки;
- отсутствие сигнатур позволяет сделать конечного пользователя СОА относительно независимым от вендора.

Недостатками СОА являются:

- высокий уровень ложных срабатываний, поскольку любое (даже не опасное) отклонение от нормальной активности спровоцирует реакцию системы;
- сложность создания обучающей выборки;
- сложность внедрения и настройки, поскольку стратегия обучения и характер данных для обучающей выборки зависит от специфики вычислительной среды.

ПРОТОКОЛЬНЫЕ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ

Существует еще один класс СОВ, который многие называют компромиссным между системами обнаружения аномалий и системами обнаружения злоупотреблений. Идея заключается в том, что СОВ следит за тем, чтобы приложения, обмениваясь информацией, следовали спецификации протокола – такие системы обнаружения вторжений называются протокольными [20].

Рассмотрим этот механизм на примере HTTP запроса. Протокол регламентирует следующую структуру запроса:

```
GET / HTTP/1.0\r\n
User-Agent: Mozilla 4.0 (MSIE 6.0, Windows NT 5.2)\r\n
<другие заголовки>
…
\r\n
```

Если злоумышленник попытается составить вредоносный запрос, например, содержащий внедряемый код, то он может выглядеть, например, так:

```
GET / HTTP/1.0\x90\x90…\xcd\x80\r\n
User-Agent: Mozilla 4.0 (MSIE 6.0, Windows NT 5.2)\r\n
<другие заголовки>
…
\r\n
```

Выделенные жирным символы не соответствуют спецификации по двум причинам:

- за версией протокола должна следовать конструкция «возврат каретки, перевод строки» (\r\n);
- байты \x90, \xcd, \x80 не являются отображаемыми ASCII символами.

Следует отметить, что протокольные СОВ всегда вынуждены хранить состояние соединения, что несколько усложняет их архитектуру.

К достоинствам таких СОВ можно отнести следующие:

- способность детектировать неизвестные атаки;
- хотя обновления таких СОВ необходимы, они происходят значительно реже.

Есть у протокольных систем и недостатки:

- хотя они способны детектировать новые типы атак, однако далеко не все, а лишь те, которые нарушают протокол;
- далеко не все приложения строго следуют спецификациям протоколов, что может повлечь за собой ложные срабатывания;
- отслеживание состояний соединения является весьма затратным с точки зрения вычислительных ресурсов.

ЗАКЛЮЧЕНИЕ

Несмотря на большое количество работ на тему обнаружения вторжений удовлетворительных результатов пока не достигнуто. Коммерческие СОВ используют классический сигнатурный поиск, а подходы к обнаружению аномалий еще слишком далеки от того, чтобы в полной мере быть реализованными на практике.

В настоящей статье было показано текущее состояние области обнаружения вторжений и основные направления исследований.

Хочется отметить, что большинство исследований носят сугубо эмпирический характер и в принципе не ставят теоретических проблем. По мнению одного из авторов настоящей статьи, вычислительные аспекты обнаружения вторжений, в частности, внедряемого кода, не только не изучены, но и не сформулированы. Данная ситуация должна быть исправлена, поскольку только при наличии границ вычислимости проблемы обнаружения вторжений возможны целенаправленные исследования и эксперименты.

Предлагаемый тезис можно пояснить на простом примере. Сегодня всё внимание уделяется конкретным угрозам, атакам и путям препятствования их реализации, однако мало кто руководствуется тем фактом, что существует счетно-бесконечное множество возможных типов атак и классические методы в принципе не способны обеспечить приемлемую защиту. А попытка детектировать вредоносную активность a priori обречена на провал, поскольку проблема верификации программы является невычислимой.

В связи с этим наиболее важным направлением изучения проблемы детектирования атак, как, впрочем, и проблемы детектирования вирусов и червей, является исследование вопросов верификации алгоритмов и поиска таких реше-

ний, который позволяют с определенной долей вероятности сделать вывод о том, копирует ли код сам себя, меняет ли он ход выполнения программы и т.п.

СПИСОК ЛИТЕРАТУРЫ

1. **Scarfone K., Mell P.** Guide to intrusion detection and prevention systems. Интернет ресурс, режим доступа: csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf, дата доступа: 20.10.2011 г.

2. **Anderson J.** Computer security threat monitoring and surveillance. Интернет ресурс, режим доступа: csrc.nist.gov/publications/history/ande80.pdf, дата доступа: 20.10.2011 г.

3. **Denning D.** An intrusion detection model // Proc. of IEEE Symposium on Security and Privacy, 1987. P. 118–131.

4. **de Boer P., Pels M.** Host-based intrusion detection systems. интернет ресурс, режим доступа: http://staff.science.uva.nl/~delaat/snb-2004-2005/p19/report.pdf.

5. Snort 2.1. Обнаружение вторжений / Р. Алдер [и др.]. М: Бином-Пресс, 2006. 656 с.

6. **Di Pietro R., Mancini L. V.** Intrusion Detection Systems. Springer-Sceince+Buisness Media, 2008. 264 c.

7. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM project / [S. J. Stolfo, et al.]. Интернет ресурс, режим доступа: http://www.cs.columbia.edu/~wfan/ PAPERS/JAM99.pdf, дата доступа: 20.10.2011 г.

8. **Portnoy L., Eskin E., Stolfo S. J.** Intrusion detection with unlabeled data using clustering // Proc. of ACM Workshop on Data Mining Applied to Security, 2001. P. 5–8.

9. **Callegari C., Vaton S., Pagano M.** A new statistical approach to network anomaly detection // Proc. of Performance Evaluation of Computer and Telecommunication Systems (SPECTS). 2008. P. 441–447.

10. **Callegari C., Giordano S., Pagano M.** Application of wavelet packet transform to network anomaly detection // Proc. of International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN). 2008. P. 246–257.

11. **Planquart J. P.** Application of neural networks to intrusion detection // SANS Institute, InfoSec Reading Room, Интернет ресурс, режим доступа: http://www.sans.org/reading_room/whitepapers/detection/application-neural-networks-intrusion-detection_336, дата доступа: 20.10.2011 г.

12. Self-nonself fiscrimination in a computer / S. Forrest [et al.] // Proc. of 1994 IEEE Symposium on Research in Security and Privacy. 1994. P. 202–212.

13. **Котов В. Д.** Система обнаружения вторжений на основе технологий искусственных иммунных систем // Интеллектуальные системы управления. М: Машиностроение, 2010. 544 с. С. 525–535.

14. **Kim J., Bentley P.** An artificial immune model for network intrusion detection, Интернет ресурс, режим доступа: http://neuro.bstu.by/our/immune3.pdf, дата доступа: 20.10.2011 г.

15. **Kotov V., Vasilyev V.** Immune approach to network intrusion detection // Proc. of Security of Information and Networks. 2010. P. 233–237.

16. **Tarakanov A. O.** Immunocomputing for intelligent intrusion detection // IEEE Computational Intelligence Magazine, Май 2008 г., с. 23–30.

17. **Kotov V., Vasilyev V.** Detection of web server attacks using principles of immunocomputing // Proc. of 2nd World Congress on Nature and Biologically Inspired Computing, 2010. P. 25–30.

18. Shieldgen: Automated data patch generation for unknown vulnerabilities with informed probing / H. J. Wang [et al.] // Proc. of IEEE Security and Privacy, 2007. P. 252–266.

19. Shield: Vulnerability-driven network filters for preventing known vulnerability exploits / H. J. Wang [et al.] // Proc. of ACM SIGCOMM, 2004. P. 193–204.

20. **Das K.** Protocol anomaly detection for network-based intrusion detection, SANS Institute InfoSec Reading Room, Интернет ресурс, режим доступа: http://www.sans.org/reading_room/whitepapers/detection/protocol-anomaly-detection-network-based-intrusion-detection_349, дата доступа: 20.10.2011 г.

ОБ АВТОРАХ

Котов Вадим Дмитриевич, асп. каф. вычислительн. техники и защиты информации.

Васильев Владимир Иванович, проф., зав. той же каф. Дипл. инженер по промэлектронике (УГАТУ, 1970). Д-р техн. наук по сист. анализу и автом. управлению (ЦИАМ, 1990). Иссл. в обл. многосвязн., многофункц. и интел. систем